Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.ijamce.com

Patrol Robot Path Planning Method Based on Boundary Fusion Algorithm

Jiajun Pan[a], Xingguo Song[a,*], Xiaojie Fang[a], Qiulin Yu[a]

[a] *School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, 610031, China;*

ARTICLE INFO

Article history:
Received 12 January 2024
Accepted 22 March 2024
Available online 24 March 2024

Keywords:
Patrol robot
Unstructured environment
Boundary fusion
Path planning
Safety performance

ABSTRACT

This article proposes a safer path planning method for 4WD wheeled patrol robots traveling in unstructured road environment, which solves the problem that the robots cannot be restricted to the road area. In this article, we propose an improved Open-planner path planning algorithm integrating RoadEdge boundaries, so that the robot gives priority to driving to the right in unstructured road environment, and always drives in the road area during the obstacle avoidance process, so as to ensure the safety of the patrol robot during the cruise. The proposed algorithm is tested in simulation and real scenarios, and the experimental results are better than the original algorithm...

1. Introduction

With the advancement of SLAM (Simultaneous Localization and Mapping) technology and the continuous improvement of computer performance, autonomous navigation technology has been widely applied in patrol robots. Autonomous navigation for patrol robots has already achieved mature applications in two-dimensional ground environments such as orchards (e.g., Cheng C et al., 2023), shopping malls (e.g., Dehuai Z et al., 2005), and campuses (e.g., Cheng C et al., 2021). However, in unstructured three-dimensional complex environments, path planning for patrol robots still presents certain challenges. These difficulties are mainly reflected in two aspects: 1) The planned path between two points for the patrol robot is often optimized for time and distance without considering the impact on other dynamic operations, as shown in Fig. 1(a); 2) Unstructured three-dimensional terrain and dynamic obstacles affect the robot's ability to drive within safe areas, as shown in Fig. 1(b).

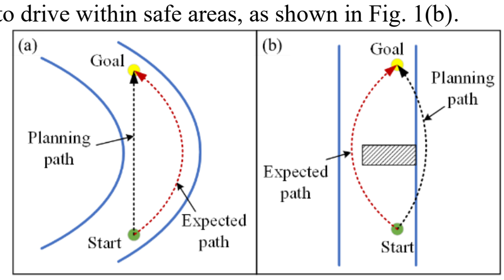

Fig. 1. Difficulty of unstructured terrain path planning.

Based on Open-planner, this article proposes an optimized path planning method to generate safer travel paths on unstructured terrain. It focuses on local path planning, visualizing road boundaries in the form of vector maps in the constructed three-dimensional point cloud map, and introducing RoadEdge boundary constraints in path evaluation to determine safe driving areas by intersection and safe distance. Compared with the original algorithm, the proposed method can generate a safer trajectory in unstructured environment to ensure the safety of the patrol robot.

In this article, the security and robustness of the proposed algorithm are tested under simulation and real scenarios, and the results are satisfactory

2. Related Work

Path planning methods based on ground robots have made remarkable progress in recent years. Commonly used path planning algorithms include Dijikstra, Floyd, A*, RRT, as well as ant colony algorithm, genetic algorithm, firefly algorithm, artificial bee colony algorithm, artificial potential field method and other intelligent algorithms (Patle B K et al., 2019). These technologies are not only widely used in the path planning of various types of patrol robots, but also continuously improved to adapt to new challenges.

In terms of indoor environment patrol, Zheng J et al. (2022) proposed an improved deep reinforcement learning path planning method to shorten the path convergence time. X Zhou and his team (2023) proposed an improved Dijkstra algorithm, which realizes global path planning and conflict coordination of multi-robot systems by introducing real-time node occupancy, but requires tracks to be

* Corresponding author.
E-mail addresses: xg.song@hotmail.com (X. Song)
Digital Object Identifiers: https://doi.org/10.62953/IJAMCE.645830

laid in advance. Aiming at the agricultural greenhouse environment, K Tsiakas et al. (2023) used semantic segmentation to identify heating guide rails and radar ranging to achieve autonomous tracking, but this algorithm could not achieve obstacle avoidance and path conversion.

Zhao D et al. (2020) proposed a new multi-objective cauchy mutation cat swarm and Artificial Potential Field method joint optimization algorithm， which can effectively balance the relationship between the shortest path and good path smoothness. Teh CK et al. (2021) proposed an extended Dijkstra algorithm based on vision, which performs dynamic obstacle avoidance through multi-sensor fusion. Although these indoor patrol algorithms have made many achievements in two-dimensional raster maps and planar path planning, they still face limitations in the perceptual dimension in unstructured environments.

Outdoor patrol robot is widely used in power plant field，The multi-interval objective path planning scheme (Chen Z et al., 2022) adopts the interval multi-objective particle swarm optimization method, which can find the shortest collision-free path with the least risk in the obstacle environment, but it is only applicable to the static environment. Lu D et al. (2023) designed an adaptive 3D obstacle avoidance path automatic planning model, which controlled the planning time and planning distance through multi-objective optimization and depth grid correction technology, but the algorithm lacked the dynamic adjustment ability for obstacle recognition. Wang et al. (2022) also proposed a bidirectional search method for substation inspection environment, which uses artificial potential field and ant colony algorithm to conduct bidirectional search, which improves the patrol efficiency and safety of robots, but there are local optimal problems. Chen L et al. (2022) proposed a new algorithm integrating chaotic neural network and genetic algorithm for wind farm environment. Through chaotic neural network planning and genetic algorithm optimization, the inspection path of the robot is shortened, but the high algorithm complexity makes it difficult to meet the real-time requirements. Due to the scarcity of personnel in power plant inspection environment, most path planning algorithms are based on the shortest path, optimal efficiency and static environment, but do not consider the dynamic environment and complex road requirements.

In terms of other outdoor environments, for the environment of chemical industrial park, Li Y et al. (2022) incorporated the fire risk level into the path planning assessment, and improved the accuracy and stability of TSP by introducing an improved IDESA algorithm. Gao Y et al. (2022) combined AFSA and DWA algorithms and introduced improved genetic algorithm to reduce the inspection time of patrol robots in a coal mine environment with low visibility and improve the path smoothness. Liu J et al. (2022) proposed A method combining improved A* and genetic algorithm to search the shortest patrol path. These methods perform well in avoiding static obstacles, but their adaptability in dynamic complex environments remains to be explored.

These studies show that in recent years, the research on the path planning of patrol robots mainly focuses on the optimization of distance and time efficiency, which is usually suitable for static and flat road environments. However, in unstructured environments, such as variable mountain roads, where there are multiple dynamic obstacles, these current algorithms may struggle to adapt to new challenges.

In recent years, more and more scholars have studied the path planning in unstructured complex environment. Aiming at the indoor environment with stairs and slopes, Wang C et al (2019) extracted multi-layer 2D maps from 3D OctoMap as navigation input, and used variable step length RRT algorithm to detect gradients for path planning, so as to avoid stairs. However, hierarchical OctoMap maps are time-consuming and inaccurate in a large-scale environment. Zhang B et al. (2022) proposed A hierarchical path planning method based on A* and Q-learning algorithms, which can divide stairs, obstacles and ramps, etc. However, when facing slopes with large angles, this method may mistakenly classify them as impassable obstacles. Huang Y and others (2023) use multi-layer Costmap as input for path planning and add radiation influence factor to the heuristic function of A* algorithm to achieve reasonable planning in a radiation environment, but the path has local oscillation, which affects the driving stability.

Jian Z et al. (2022) proposed an uneven terrain navigation framework (PUTN) based on plane fitting. By introducing improved PF-RRT* and Gaussian process regression, the framework can effectively deal with path planning in uneven terrain, but it is prone to stall in the process of turning. Josef S et al. (2020) proposed a DRL deep reinforcement learning method, which can avoid obstacles and potholes in unknown terrain, but the path in the drivable area is random and lacks practical verification. H. Desarweesh and his team (2017) developed an Open-planner open-source planning algorithm based on the autoware platform. The architecture is shown in Fig. 2 The algorithm independently plans on the existing three-dimensional point cloud map, and generates the main and auxiliary trajectories locally, so that the vehicle can preferentially drive on the main path, but may drive out of the road area when avoiding obstacles.

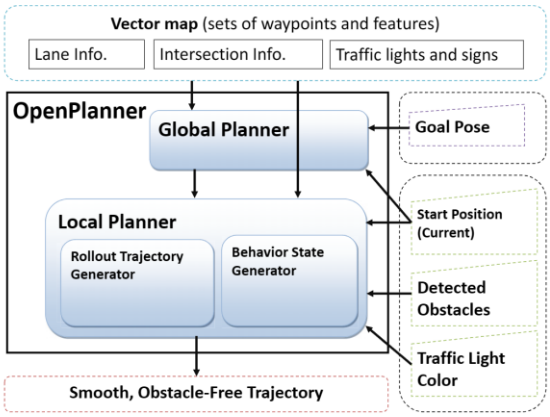

Fig. 2. Open-planner architecture

The above path planning algorithm may have its unique advantages in terms of planning efficiency and obstacle avoidance, but it may not be able to deal with two problems in practical application: 1) It will affect other people or things in the process of path planning and obstacle avoidance; 2) When there are exercisable areas in the same level, such as roadside grass, these areas cannot be identified. In order to solve these problems, this paper will study the Open-planner path planning algorithm.

3. Path planning method optimization

3.1 Path planning process

Open-planner makes global path planning easier and faster because it removes kinematic optimization from the equation and uses vector diagrams to solve the problem. The red line shown in Fig. 3(a) is a Lane line vector diagram visualized in the form of lane, consisting of a series of points, each of which contains three-dimensional position information and vectors pointing to the next Point. In the case that the robot position and target point are known, the system looks for the optimal sequence from the point closest to the robot's current position to the point closest to the target point, as shown in Fig. 3(b).

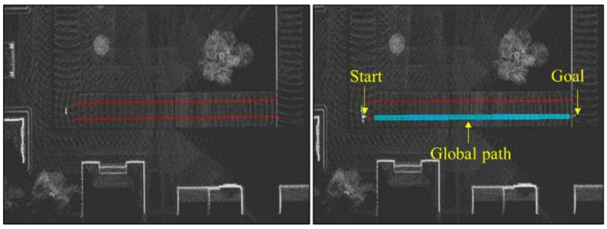

Fig. 3. Global path planning

The local path is planned based on the global path. The main process is as follows: 1) Receiving the global path information; 2) Obtain filtered and clustered information about perceived obstacle point clouds; 3) Candidate path generation: Generate initial RollOut paths according to initialization parameters such as path number and transverse density, and conduct path smoothing and post-processing, as shown in Fig. 4(a); 4) Path assessment: calculate the cost function from the three dimensions of central cost, transition cost and collision cost, select the local path with the lowest generation value as the optimal path, and block the path with possible collision. As shown in Fig. 4(b), red segments are locked paths, pink segment is the cost optimal path, and other colors segments are passable paths. 5) Behavior generation: The optimal local path is converted into a specific trajectory, and information such as robot speed and turning radius is generated to output robot control commands.

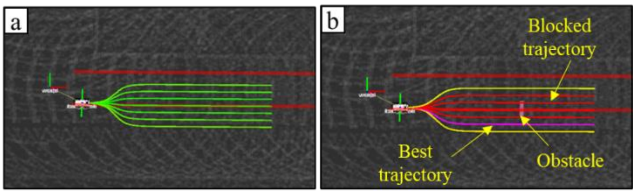

Fig. 4. Local trajectory generation

3.2 RoadEdge border fusion

It can be seen from Fig. 4(b) that when avoiding obstacles, the robot will drive along the cost optimal trajectory. Judging from the habit of preferentially driving to the right on domestic roads, the robot may drive away from the road area at this time. In this article, RoadEdge boundary fusion is used to solve this problem.

This article builds Gazebo simulation environment independently, as shown in Fig. 5(a), and uses NDT-Mapping to construct 3D map, as shown in Fig. 5(b). Similar to the Lane vector diagram, we drew the RoadEdge vector diagram based on the point cloud diagram, and generated a point set with bidirectional index, and added a visual display in Rviz, as shown in Fig. 5(b) green line segment.

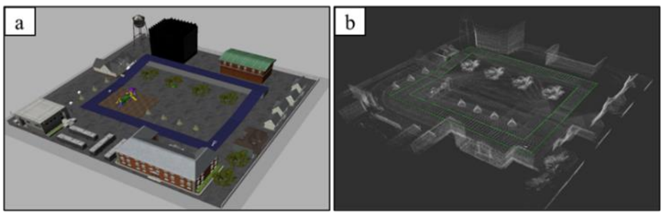

Fig. 5. Simulation environment and map construction

Given m generated local trajectories $T=\{T_1,T_2,...T_m\}$, each trajectory is equidistantly distributed with n Waypoints $P=\{p_1,p_2,...,p_n\}$ from the current position to the furthest distance planned by the local path. Similarly, given i generated RoadEdges $R=\{R_1,R_2,...R_i\}$, each RoadEdge equidistantly distributed with j EdgePoints $P'=\{p'_1,p'_2,...p'_j\}$ from the starting point to the endpoint. Using all distributed points as input, calculate the intersection and safety distance between each local path and road boundaries, and output feasible and safe path index for subsequent evaluation and selection.The basic flow of the algorithm is shown in Algorithm 1.

Algorithm 1 Path-Edge Intersection and Proximity Detection

Input: *T, P, R, P', SafetyDistance*
Output: *BlockedID*- ID of the blocked trajectory
1: Initialize *id* as -1
2: **For each** T_a in *T* **do**
3: *TrajectoryBlocked* ← False
4: **For each** P_b in T_a **do**
5: **For each** R_c in *R* **do**
6: **For each** P'_d in R_c **do**
7: **if** *IsIntersect*($(P_b,P_{b+1}),(p'_d,p'_{d+1})$) OR *Distance*($(P_b,P_{b+1}),(p'_d,p'_{d+1})$)< *SafetyDistance*
8: *TrajectoryBlocked* ← True
9: Exit all loops and go to Step 10
10: **If** *TrajectoryBlocked*
11: *id* ← ID of the current trajectory T_a
12: Append *id* to *BlockedID*
13: **Return** *BlockedID*

In the process, IsIntersect() is a function to calculate the intersection of two line segments, which takes as input two line segments connected by two path points adjacent (P_b,P_{b+1}) to the local path and two points adjacent (p'_d,p'_{d+1}) to RoadEdge. First, a fast exclusion test is performed to check whether the boundary boxes of two line segments intersect. If the intersection problem on the three-dimensional terrain is simplified to the XY two-dimensional plane problem of the road surface where the robot is currently located as

$$\begin{cases} \min x = \min(p_b.x, p_{b+1}.x) \\ \min y = \min(p_b.y, p_{b+1}.y) \\ \min x' = \min(p_d'.x, p_{d+1}'.x) \\ \min y' = \min(p_d.y', p_{d+1}.y') \end{cases} \tag{1}$$

If

$$\min x < \min x' \,\|\, \min x' < \min x \;\| \quad \min y < \min y' \,\|\, \min y' < \min y \tag{2}$$

the line segments do not intersect, otherwise the boundary boxes of the two line segments intersect. It is necessary to conduct further straddle experiments to determine whether line segments intersect. The judging condition is that if two line segments intersect, the two endpoints of one line segment must be on both sides of the line where the other line segment is located, and vice versa.

For line segment $\overline{p_b p_{b+1}}$ relative to $\overline{p'_d\, p_{d+1}}$, calculate as

$$\begin{cases} d_1 = (p'_d - p_b)\times(p_{b+1}-p_b) \\ d_2 = (p'_{d+1} - p_b)\times(p_{b+1}-p_b) \end{cases} \tag{2}$$

For line segment $\overline{p'_d\, p_{d+1}}$ relative to $\overline{p_b p_{b+1}}$, calculate as

$$\begin{cases} d_3 = (p_b - p'_d) \times (p'_{d+1} - p'_d) \\ d_4 = (p_{b+1} - p'_d) \times (p'_{d+1} - p'_d) \end{cases} \tag{3}$$

Where ‘ × ’ represents the cross product in two-dimensional space, if d_1 and d_2 symbols are different, and d_3 and d_4 symbols are different, the two line segments intersect, otherwise they do not intersect.

If only by the intersection judgment, the result is shown in Fig. 6(a). It can be seen that the outermost path is blocked at this point, but there is still a candidate path close to the RoadEdge. Due to the width and wheel spacing of the robot, if the path trajectory is selected, the robot safety frame may exceed the driving boundary. Therefore, the *SafetyDistance* parameter should be set according to the robot safety frame size, and the minimum Distance L between the candidate path point and the RoadEdge point should be calculated in the *Distance()* function as

$$L = \min(l(p_b, p'_d), l(p_b, p'_{d+1}), l(p_{b+1}, p'_d), l(p_{b+1}, p'_{d+1})) \tag{4}$$

Where l represents the distance between two endpoints, when $L < SafetyDistance$, the ID path is blocked, as shown in Fig. 6(b).

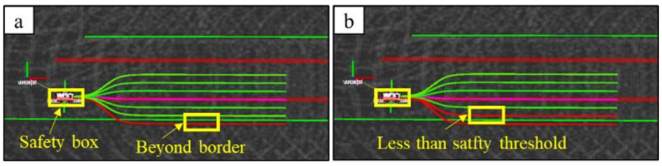

Fig. 6. Effect of RoadEdge road boundary

If only distance judgment is carried out, since the path and boundary are a series of spacing points, there may be special cases, as shown in Fig. 7. If $L > SaftyDistance$ at that time, the paths would not blocked, but the paths actually intersect. Therefore, through the dual judgment of intersection and boundary safety distance, the robot can fully ensure that it always chooses a safe trajectory.

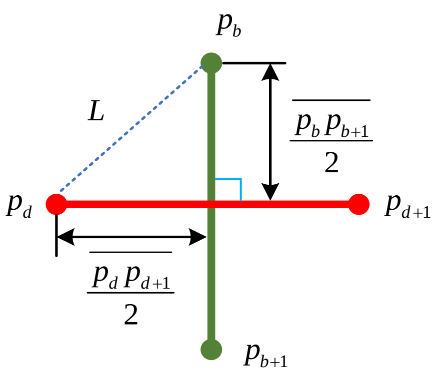

Fig. 7. Calculation of distance between path and boundary

4. Path planning method optimization

4.1 Simulation Experiment

In order to verify the superiority of the improved algorithm in this article, we built several simulation environments in Gazebo to compare the improved algorithm with the original Open-planner algorithm. The robot model is driven by 4WD chassis, equipped with C16 LiDAR for environment perception, and integrated with GPS for positioning. The main parameters are set as shown in Tab. 1. The trajectories of the robot are tracked and drawn by LK optical flow method.

Tab. 1. Gazebo simulation parameters

Name of parameter	Default
Maximum speed	1m/s
Maximum acceleration	2m/s
Maximum angular velocity	1rad/s
Path planning length	15m
Maximum obstacle avoidance distance	10m
Minimum obstacle avoidance distance	2m
Filtering Angle	20°

Scenario 1:

As shown in Fig. 8(a), the road is composed of the first half of the horizontal straight road and the second half of the 15° ramp, the road width is 5m, and the starting position of the robot is shown in the yellow circle.

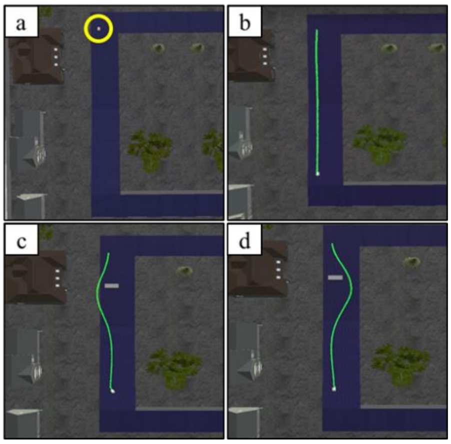

Fig. 8. Straight road scene simulation experiment

In barrier-free environment, the paths before and after algorithm improvement are consistent as shown in Fig. 8(b). The robot drives on the right side of the road. After placing obstacles in the road, the paths before and after algorithm improvement are shown in Fig. 8(c) and Fig. 8(d) respectively. The original algorithm will cause the robot to drive away from the road area when avoiding obstacles, but the improved algorithm keeps the robot driving in the road area all the time.

Scenario 2:

Compared with scenario 1, multiple obstacles are added in this scenario. As shown in Fig. 9, the longitudinal spacing of these obstacles is greater than 5 times the maximum driving speed to ensure that the robot has sufficient obstacle-avoiding reaction time.

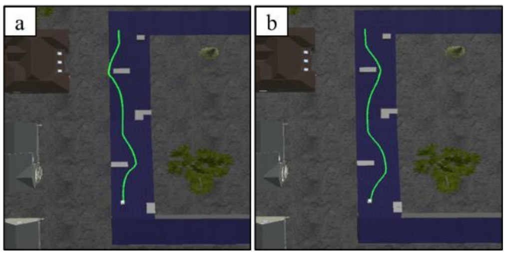

Fig. 9. Straight multi-obstacle scene simulation experiment

Before integrating the RoadEdge, the robot can avoid obstacles in a multi-obstacle environment, but there is a risk of driving away from the safe road area, as shown in the position of the red circle in Fig 9(a). The driving trajectory after the fusion of RoadEdge is shown in Fig 9(b). At this time, when the robot passes the first obstacle, it will choose the path in the drivable area to ensure that the robot can always drive in the road area even in the process of frequent avoidance of obstacles, and ensure the safety of the robot.

Scenario 3:

In this scenario, we build a horizontal S-shaped road, as shown in Fig. 10. The road consists of two semicircles with a central axis radius of 12m and a road width of 4m. The robot starts at one end of the road and travels to the end of the road.

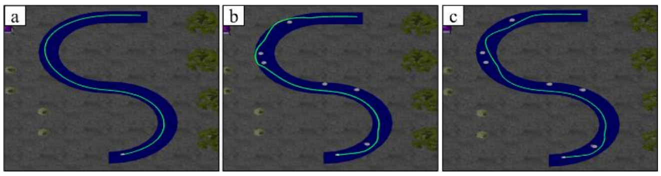

Fig. 10. Plane S-shaped road scene simulation experiment

When there are no obstacles, the trajectory of the robot before and after the optimization algorithm is the same, as shown in Fig. 10(a). Open-planner can make the robot give priority to driving on the right in normal driving state.

Randomly place cylindrical obstacles with a diameter of 1m on the road. The driving trajectory of the original Open-planner algorithm is shown in Fig. 10(b). The robot drives off the track when it passes the first cylinder, then returns to the road and drive off the road again when it passes the second and third cylinders. When using this optimization algorithm for path planning, the trajectory is shown in Fig. 10(c). When the robot passes the first cylinder, RoadEdge changes the candidate path selection strategy so that the robot drives on the left side of the cylinder. In the subsequent navigation process, the robot always maintains the obstacle avoidance strategy in the road area.

Scenario 4:

In this Scenario, there is a curved ramp with varying width, as shown in Fig. 16. The maximum slope of the road is 15°, the maximum width is 4m, the minimum width is 2m, the minimum turning radius of the road boundary is 1m, and the maximum turning Angle is 120°. The robot starts from the bottom of the slope and climbs along the road to the top target point.

When using the original Open-planner algorithm for path planning, the robot's driving trajectory in barrier-free condition was shown in Fig. 11(a). The robot can safely drive to the target point and stay on the right side of the road. When placing cylindrical obstacles with a diameter of 1 meter in the area with the larger width of the road. as shown in Fig. 11(b). The robot drives off the road area and fall off the ramp while avoiding the last obstacle. When using the optimization algorithm to plan the path, due to the influence of RoadEdge fusion, the robot will only select the candidate path in the road area for tracking. At this time, the robot can avoid all obstacles and reach the planned target point, as shown in Fig. 11(c).

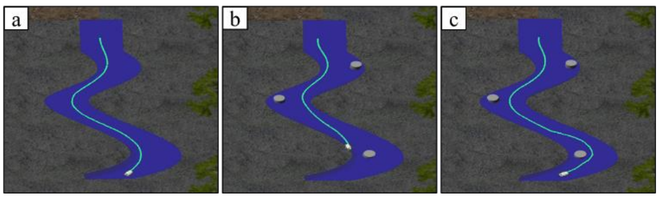

Fig. 11. Curved lane change wide ramp scene simulation experiment

4.2 Actual experiment

To further verify the robustness of this algorithm in practical applications, we independently built a 4WD wheeled robot prototype, as shown in Fig. 12.

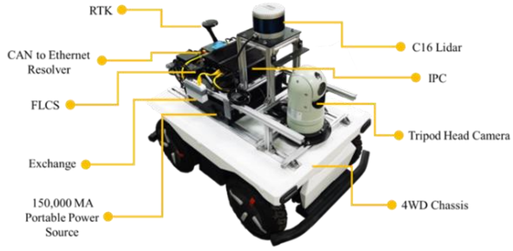

Fig. 12. Patrol robot experiment platform

With reference to the simulation experiment, this article built two static experiment scenarios. In the experiment, the yellow lines were used as the road boundary, Between the two yellow lines is the safe area, and outside the yellow lines is the unsafe area, the road width is 4m, and the robot moved forward at 1m/s.

Scenario 1:

As shown in Fig. 13, this scene is an outdoor flat straight road environment in which multiple obstacles are randomly placed.

Before algorithm optimization, in order to avoid the first obstacle at t=6.5s, the robot chooses the right path to follow and drives out of the boundary, then returns to the safe driving area and drives off the road again at t=18s, and finally reaches the end at t=22s.

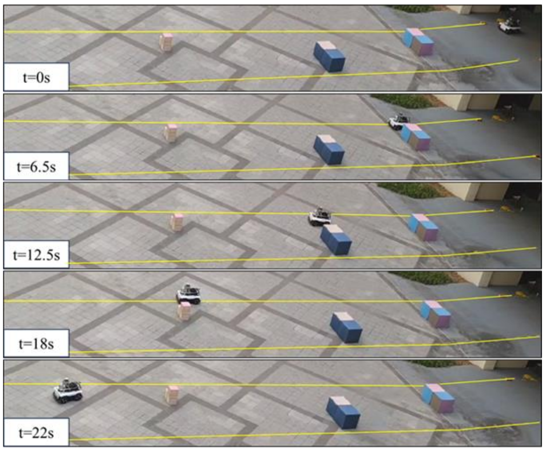

Fig. 13. Straight original path planning algorithm

After algorithm optimization, the driving state in the same scene is shown in Fig. 14. At t=7s, affected by RoadEdge, the robot changes to drive on the left side of the road area. At t=13.5s, the robot drives on the right to avoid obstacles. At t=20s, the robot also chooses a safe driving strategy different from the original algorithm, and reaches the end at t=23s. The robot drives in the safe area throughout.

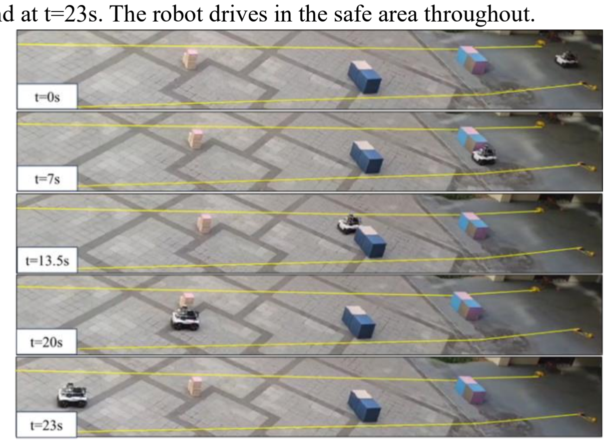

Fig. 14. Straight optimization path planning algorithm

Scenario 2:

As shown in Fig. 15, the scene is an outdoor flat curve environment. Multiple obstacles are placed in the road area surrounded by yellow lines. The robot starts on one side of the curve and reaches the target point on the other side. When using the original Open-planner for path planning, the robot travels within the road range from departure to t=12s. However, in the period of 18s to 23s, due to the lowest value of the outer pat, the robot tracks the path away from the road and finally reaches the end point at t=32s.

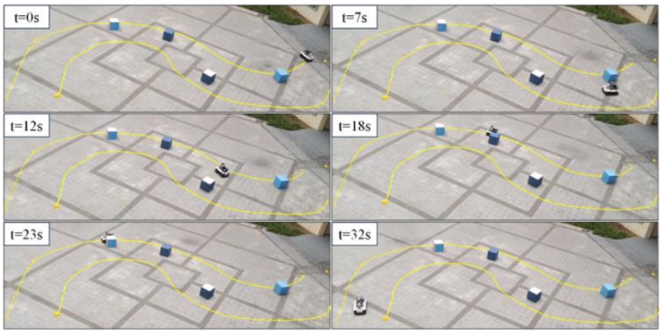

Fig. 15. Curve original path planning algorithm

The obstacle-avoiding driving state after optimization of the algorithm is shown in Fig. 16. By integrating safety boundary constraints in the path planning stage, the robot is ensured to always stay in the road area during the entire autonomous navigation process, and the safe obstacle-avoiding driving of the robot is realized.

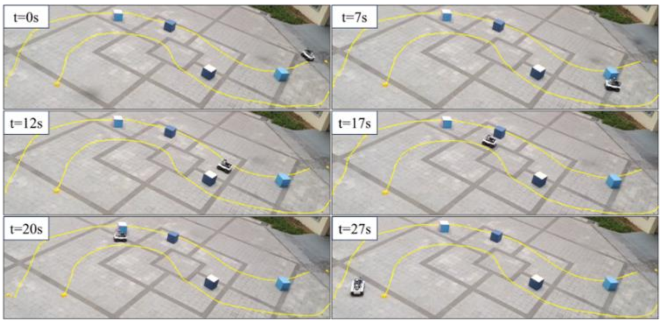

Fig. 16. Curve optimization path planning algorithm

5. Conclusion

This article presents an improved Open-planner path planning algorithm based on RoadEdge fusion. Through boundary fusion, rapid exclusion experiment and straddle experiment were used to verify the intersection between the candidate path and the boundary, and sets a safe distance threshold, so that the patrol robot would not drive away from the planned road area while preferentially driving to the right, and improves the driving stability of the patrol robot in the multi-obstacle unstructured environment. We verify the robustness of the improved algorithm in simulation and real environment.

References

Cheng C, Fu J, Su H, et al. Recent advancements in agriculture robots: Benefits and challenges[J]. Machines, 2023, 11(1): 48.

Dehuai Z, Cunxi X, Xuemei L. Design and implementation of a security and patrol robot system[C]//IEEE International Conference Mechatronics and Automation, 2005. IEEE, 2005, 4: 1745-1749.

Kitchagiri P, Vaddi S, Rajanala S R, et al. Night Vision Patrolling Robot for Security Pat rolling Using Raspberry Pi[J]. International journal of research and application, 2021, 11.

Patle B K, Pandey A, Parhi D R K, et al. A review: On path planning strategies for navigation of mobile robot[J]. Defence Technology, 2019, 15(4): 582-606.

Zheng J, Mao S, Wu Z, et al. Improved path planning for indoor patrol robot based on deep reinforcement learning[J]. Symmetry, 2022, 14(1): 132.

Zhou X, Yan J, Yan M, et al. Path Planning of Rail-Mounted Logistics Robots Based on the Improved Dijkstra Algorithm[J]. Applied Sciences, 2023, 13(17): 9955.

Tsiakas K, Papadimitriou A, Pechlivani E M, et al. An Autonomous Navigation Framework for Holonomic Mobile Robots in Confined Agricultural Environments[J]. Robotics, 2023, 12(6): 146.

Zhao D, Yu H, Fang X, et al. A path planning method based on multi-objective cauchy mutation cat swarm optimization algorithm for navigation system of intelligent patrol car[J]. IEEE Access, 2020, 8: 151788-151803.

Teh C K, Wong W K, Min T S. Extended Dijkstra algorithm in path planning for vision based patrol robot[C]//2021 8th International Conference on Computer and Communication Engineering (ICCCE). IEEE, 2021: 184-189.

Chen Z, Wu H, Chen Y, et al. Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm[J]. Applied soft computing, 2022, 116: 108192.

Lu D, Ning X, Li L. Research on automatic path planning for obstacle avoidance of intelligent substation patrol robot[C]//International Conference on Automation Control, Algorithm, and Intelligent Bionics (ACAIB 2023). SPIE, 2023, 12759: 142-150.

Wang L, Ci W, Liu X, et al. Autonomous global path planning method for substation remote patrol robot[C]//18th International Conference on AC and DC Power Transmission (ACDC 2022). IET, 2022, 2022: 1261-1266.

Chen L, Hu Z, Zhang F, et al. Remote wind farm path planning for patrol robot based on the hybrid optimization algorithm[J]. Processes, 2022, 10(10): 2101.

Li Y, Chen S, Bai K, et al. Path planning of patrol robot based on improved discrete electrostatic discharge algorithm[J]. Journal of Intelligent & Fuzzy Systems, 2022, 42(6): 5919-5930.

Gao Y, Dai Z, Yuan J. A multiobjective hybrid optimization algorithm for path planning of coal mine patrol robot[J]. Computational Intelligence and Neuroscience, 2022, 2022.

Liu J, Xi B, Chen S, et al. The Path Planning Study of Autonomous Patrol Robot based on Modified Astar Algorithm and Genetic Algorithm[C]//2022 34th Chinese Control and Decision Conference (CCDC). IEEE, 2022: 4713-4718.

Wang C, Wang J, Li C, et al. Safe and robust mobile robot navigation in uneven indoor environments[J]. Sensors, 2019, 19(13): 2993.

Zhang B, Li G, Zheng Q, et al. Path planning for wheeled mobile robot in partially known uneven terrain[J]. Sensors, 2022, 22(14): 5217.

Huang Y, Shi X, Zhou Y, et al. Autonomous navigation of mobile robot in radiation environment with uneven terrain[J]. International Journal of Intelligent Robotics and Applications, 2023, 7(3): 497-509.

Jian Z, Lu Z, Zhou X, et al. Putn: A plane-fitting based uneven terrain navigation framework[C]//2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022: 7160-7166.

Josef S, Degani A. Deep reinforcement learning for safe local planning of a ground vehicle in unknown rough terrain[J]. IEEE Robotics and Automation Letters, 2020, 5(4): 6748-6755.

Darweesh H, Takeuchi E, Takeda K, et al. Open source integrated planner for autonomous navigation in highly dynamic environments[J]. Journal of Robotics and Mechatronics, 2017, 29(4): 668-684.

Jiajun pan is currently pursuing his MS study at the School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China. He obtained her BS degree from Southwest Jiaotong University of Commerce, China in 2018. His main research interests are in the areas of SLAM, autonomous navigation and object detection.

Xingguo Song, Ph.D., graduated from Harbin Institute of Technology, School of Mechanical and Electrical Engineering, majoring in Mechanical Design and Theory, is a visiting scholar at Rice University and a postdoctoral fellow at Johns Hopkins University, USA. His main research interests are intelligent robotics, UAV path planning, bionic robotics, and computer vision.

Xiaojie Fang received his bachelor's degree from Qingdao University of Science and Technology, China, in 2021. At present, he is studying for a master's degree with the School of Mechanical Engineering, southwest Jiaotong University, China. His current research interests include semantic segmentation, object detection, and intelligent robot path planning.

Qiulin Yu is currently pursuing his MS study at the School of Mechanical Engineering, Southwest Jiaotong University, Chengdu, China. He obtained her BS degree from Southwest Jiaotong University of Commerce, China in 2022. His main research interests are in the areas of Intelligent identification and autonomous navigation technology.