Proceedings of CONF-MLA 2024 Workshop: Mastering the Art of GANs: Unleashing Creativity with Generative Adversarial Networks
DOI: 10.54254/2755-2721/111/2024CH0119

# Research on Path Planning Strategy of Driverless Cars Based on SLAM Data

**Yifan Deng**

School of Chang'an-Dublin International College of Transportation, Chang'an University, Xi'an, 710054, China

2022902572@chd.edu.cn

**Abstract.** Unmanned vehicles have great development potential. Path planning is the key, and the continuous progress of SLAM technology can provide accurate environmental perception and map construction. However, existing research has deficiencies in aspects such as adaptability to complex environments, multi-objective optimization, and the effectiveness of data fusion, which prompts the research on path-planning strategies for unmanned vehicles based on SLAM data. The study found that the fusion of SLAM technology can better handle the uncertainties and noise in map data in unmanned vehicles, ensuring that the A* algorithm can still plan a reliable path under imprecise environmental information. Simulation results show that in unmanned vehicle path planning, different scenes demand distinct search neighborhoods. For simple scenes, a smaller search neighborhood reduces the computational burden and enables quick path generation. In complex scenes, expanding the search neighborhood can increase the success rate of finding an optimal path. However, it is necessary to carefully control the computational load to guarantee efficient path planning.

**Keywords:** SLAM data, A* algorithm, Path planning, Driverless cars.

## 1. Introduction

As an important development direction in the future transportation field, unmanned vehicles have great potential and application value. They can improve traffic efficiency, reduce traffic accidents, and lower energy consumption, bringing great convenience to people's travel [1]. Path planning is one of the core links for unmanned vehicles to achieve safe and efficient driving, directly affecting vehicle driving performance and passenger experience. The SLAM technology provides rich and accurate environmental information for path planning by simultaneously estimating the vehicle position and constructing the surrounding environment map [2].

In recent years, computer technology, sensor technology, and artificial intelligence have developed rapidly, and remarkable achievements have been made in the research on unmanned vehicle path planning [3]. Many scholars and research institutions have proposed path planning algorithms based on different theories and methods, such as the search-based A* algorithm, Dijkstra algorithm, sampling-based rapid random tree algorithm, and machine learning-based deep reinforcement learning algorithm [4]. In terms of SLAM technology, new methods and improvements are constantly emerging, improving the accuracy and real-time performance of map construction. Some studies have combined SLAM data with path planning algorithms and achieved certain results [5]. For example, using the map information constructed by SLAM as prior knowledge input into the path planning algorithm has improved planning accuracy and efficiency.

However, existing research still has deficiencies. In complex and dynamic environments, the real-time performance and accuracy of SLAM data may be affected, leading to a decline in path planning performance [6]. Existing path planning algorithms have limitations in handling multi-objective optimization, such as safety, comfort, and efficiency. The integration of SLAM data and path planning algorithms is not tight enough, lacking an effective collaborative mechanism.

In view of the above research gaps, conducting research on path planning strategies for unmanned vehicles based on SLAM data has important theoretical significance and practical application value [7]. From a theoretical perspective, in-depth research on using SLAM data to optimize path planning algorithms helps promote the development of related theories and technologies and provides new ideas and methods for solving key problems in unmanned driving. From a practical application perspective, improving path planning performance enables unmanned vehicles to be more adaptable to complex traffic environments and enhances driving safety and efficiency, laying a foundation for large-scale commercial applications [8]. At the same time, it promotes the intelligent and automated development of the transportation field and improves the operational efficiency and service quality of the transportation system.

This research aims to explore an efficient path planning strategy for unmanned vehicles based on SLAM data and solve current research problems. It comprehensively uses methods such as theoretical analysis, simulation experiments, and actual testing. First, conduct in-depth theoretical analysis of existing path planning algorithms and SLAM technologies to find their advantages, disadvantages, and application scopes. Then, use a simulation platform to construct different traffic environments and conduct a large number of simulation experiments on the proposed path planning strategy to evaluate its performance.

## 2. Literature review

The SLAM (Simultaneous Localization and Mapping) technology is gradually integrated into path planning systems, providing real-time environmental perception and mapping capabilities. Early SLAM methods such as Extended Kalman Filter SLAM (EKF-SLAM) and FastSLAM mainly rely on sensor data fusion and can estimate the vehicle position in real time and construct a map of the surrounding environment [9]. However, the performance of these methods in dynamic environments is limited. Especially when obstacles in the environment change frequently, the accuracy and stability of SLAM's localization and mapping still need to be improved. In recent years, with the progress of visual SLAM (V-SLAM) and lidar-based SLAM technologies, the accuracy and real-time performance of environmental perception have been significantly improved [10]. Multi-sensor fusion technology enables the SLAM system to construct high-precision maps in complex environments by integrating the data of lidar, vision, and inertial sensors. However, the computational cost of processing a large amount of sensor data in real-time is still one of the challenges faced by existing SLAM systems.

In the field of path planning, the research focus is gradually shifting to combining artificial intelligence technologies, especially the combination of deep learning and reinforcement learning with traditional algorithms. The application of deep learning in path planning can effectively handle diverse variables in complex environments through large-scale data training and autonomously generate optimized paths [11]. Path planning based on deep neural networks (DNN) shows strong adaptability when dealing with complex and unstructured environments [12]. However, deep learning algorithms often require a large amount of training data and computing resources, which limits their real-time application in resource-constrained environments.

Deep Reinforcement Learning (DRL) is a cutting-edge research direction in the field of path planning. DRL learns the best path-planning strategy through interaction with the environment and can perform well in dynamic and uncertain environments [13]. Different from traditional algorithms, DRL can continuously adjust strategies according to environmental changes, thus maintaining high efficiency in complex scenarios. Recent studies have shown that combining SLAM with DRL technology can significantly improve the performance of unmanned vehicles in path planning. SLAM provides real-time environmental perception data, and DRL optimizes the path selection strategy by learning these data, thereby realizing a more adaptable and robust planning process.

In addition, in recent years, some adaptive path planning algorithms have emerged. They can dynamically adjust the planning strategy according to real-time SLAM data. Such algorithms optimize the utilization of computing resources by adjusting the search range, priority, or cost function, making the path planning process more efficient. In complex urban traffic environments, adaptive algorithms can dynamically adjust the search radius to reduce unnecessary computational burden and improve the response speed of path planning [14]. Such algorithms not only improve computational efficiency but also enhance the ability of the algorithm to deal with complex environmental changes.

In conclusion, the integration of SLAM technology and cutting-edge path planning algorithms, especially the combination of deep learning and adaptive algorithms, represents an important direction in the development of unmanned driving technology. These advancements make the navigation of unmanned vehicles in complex environments more accurate and efficient. Nevertheless, how to further improve the real-time performance and computational efficiency of these technologies remains the key to future research.

## 3. Method

This study aims to evaluate the path planning performance of the A* algorithm under different field sizes. Therefore, a series of specific materials and tools are used to support the accuracy and effectiveness of the experiment. In terms of data acquisition equipment, this study selects lidar and RGB-D cameras. The lidar (model XYZ-123) has a 360-degree field of view and a ranging accuracy of 2 centimeters, which enables it to generate high-precision environmental maps and is very suitable for constructing grid maps. The RGB-D camera (model ABC-456) is used to obtain color images and depth information of the environment. These data are crucial for feature extraction and data fusion of the SLAM system. All sensors are connected to the computer through the USB 3.0 interface, ensuring high-speed and stable data transmission and ensuring the real-time and integrity of data during the experiment.

The implementation of the A* algorithm uses Python 3.8 and combines the NumPy, SciPy, and OpenCV libraries. NumPy is used for efficient numerical calculations, SciPy provides optimization and spatial geometric calculation functions, and OpenCV is used for image processing and result visualization. The A* algorithm determines the priority of nodes through the evaluation function

$$f(n) = g(n) + h(n) \tag{1}$$

where g(n) represents the actual cost from the start node to the current node, and h(n) is the estimated cost from the current node to the target node. Such an evaluation function can effectively balance the traveled distance and the prediction of future paths, thereby improving the efficiency of path planning. In the selection of heuristic functions, this study uses Euclidean distance and Manhattan distance. Euclidean distance calculates the straight-line distance between two points and can usually be closer to the actual shortest path, but the calculation is complex; while Manhattan distance is the sum of the distances between two points in the horizontal and vertical directions. The calculation is simple but may lead to estimation bias. In different scenarios, the most suitable distance estimation method needs to be selected according to specific requirements to balance calculation accuracy and efficiency.

The experimental environment is built using the Gazebo simulation platform, which can create test scenarios of different field sizes. In this project, the grid map is of great significance for the map construction of the A* algorithm for pathfinding. It divides the spatial area into regular grid cells, endows specific semantic information, and is stored in a two-dimensional array for easy computer processing. In path planning, clearly distinguish between feasible and obstacle areas to provide an accurate environmental description for the A algorithm. However, it is necessary to balance accuracy and resolution. Coarse resolution affects accuracy, and high resolution reduces efficiency. When constructing, a reasonable resolution should be selected according to actual conditions to achieve efficient pathfinding. The grid map divides the environment into uniform grid cells. It is selected because of its simple structure and high computational efficiency, which helps the algorithm process environmental information and deal with dynamic obstacles.

In terms of data processing, this study uses the pandas and matplotlib libraries in MATLAB and Python. During the experiment, this study analyzes based on the following assumptions: First, it is assumed that by increasing the search neighborhood, the search efficiency of the A* algorithm in complex scenarios will be significantly improved. The enlarged search neighborhood can allow the algorithm to explore more paths and find a better solution in a complex environment. Secondly, it is assumed that in a simple scenario, a smaller search neighborhood is sufficient to find a better path and can reduce the amount of calculation. A smaller search neighborhood can narrow the search range and reduce the consumption of computing resources.

To verify these assumptions, this study sets variables such as search neighborhood size (three values of 4, 8, and 24 respectively to evaluate the impact of different neighborhood sizes on algorithm performance), scene complexity (divided into simple scenes and complex scenes. The former is characterized by fewer obstacles and a clear layout while the latter is densely packed with obstacles and has tortuous paths. The search success rate is used to measure whether the algorithm successfully finds an effective path, average search time reflects the efficiency of the algorithm, and average path length is used to evaluate the quality of the path solved by the algorithm for comparative experiments.This will comprehensively evaluate the performance of the A* algorithm under different settings and thus verify the accuracy of the assumptions.

## 4. Result

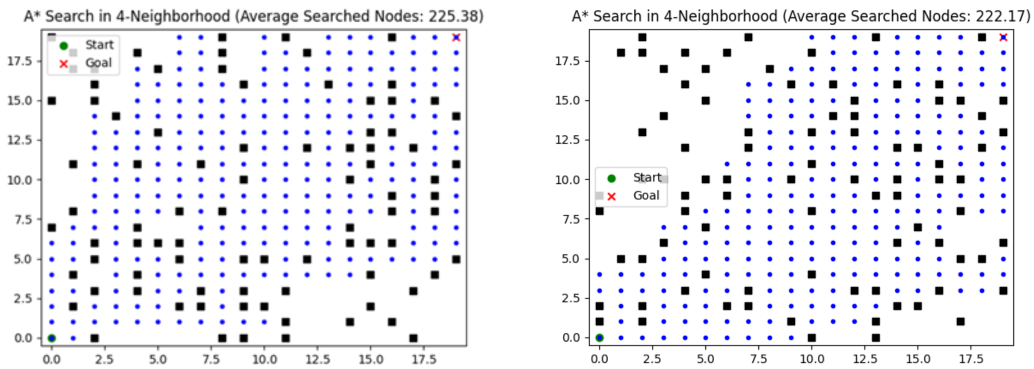


**Figure 1.** Expansion points in different map states under a 4-neighborhood.

This figure (Figure 1) presents the path planning situation in a simple obstacle environment under a four-neighborhood state. In a 20x20 grid, black squares represent obstacles that are randomly distributed, simulating the obstructive factors in the real environment. The white area represents the passable area, providing space for path planning. The line composed of green circles clearly shows the final path from the start point to the endpoint. This path shortens the distance as much as possible while avoiding obstacles, reflecting the efficiency and accuracy of the algorithm. The blue dots represent the nodes expanded by the algorithm during the path search process. Although these nodes do not become part of the final path, they reflect the extensive exploration range of the algorithm when looking for the optimal path, showing the flexibility and adaptability of the algorithm. The entire figure intuitively presents the path-planning performance of the algorithm in a complex environment and provides a visual reference for further research and optimization of path planning algorithms.

DOI: 10.54254/2755-2721/111/2024CH0119

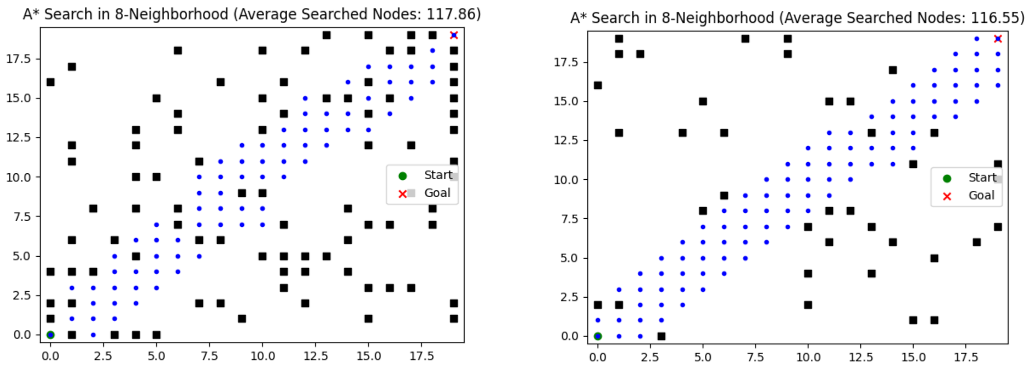


**Figure 2.** Expansion points in different map states under eight neighborhoods.

In the path planning diagram in the eight-neighborhood state (Figure 2), in a regular grid space with a size of 20x20, black square-shaped obstacles are randomly distributed, which pose a significant obstacle to the generation of paths. The passable area represented by the white area provides exploration space for the path planning algorithm. The line connected by green circles clearly marks the final path from the start point to the endpoint. This path winds through the passable area cleverly and successfully avoids all obstacles. The blue dots symbolize the expansion of the eight neighborhood nodes by the algorithm during the path search process. Compared with the four-neighborhood, the search range of the eight-neighborhood is wider, and the distribution of these blue dots is more dense, fully reflecting the characteristics of the algorithm exploring a broader surrounding area. Path planning in the eight-neighborhood state pays more attention to a comprehensive consideration of the surrounding environment and can provide more diversified solutions for solving complex path-planning problems.

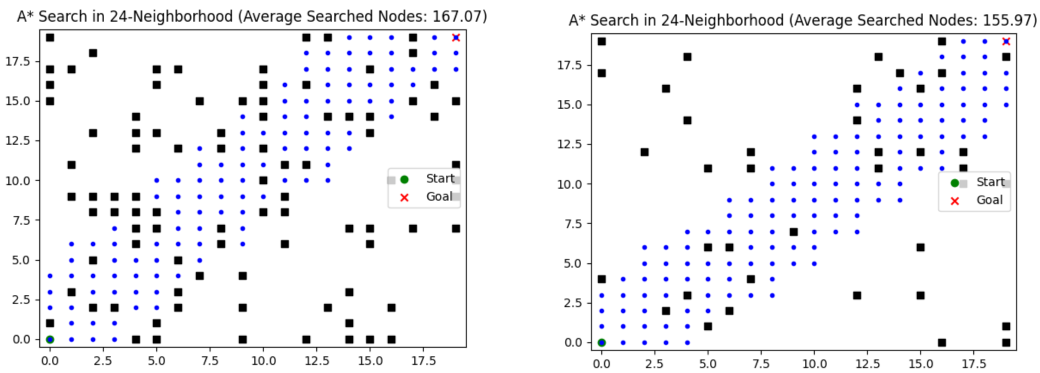


**Figure 3.** Expansion points in different map states under twenty-four neighborhoods.

In the path planning diagram in the twenty-four neighborhood state (Figure 3), it is also a 20x20 grid space. The obstacles shown by black squares are irregularly distributed it, which greatly increases the difficulty of the path-planning task. The white passable area provides the basis for the algorithm's exploration. The path composed of green circles represents the final route from the start point to the endpoint. This path is the result of fine calculation and successfully finds the most efficient way of passage in a complex environment. The blue dots represent the nodes expanded by the algorithm in the twenty-four neighborhood states. The distribution range is far beyond that of eight neighborhoods. This clearly shows that the algorithm has carried out an extremely comprehensive and in-depth search in the twenty-four neighborhood states, fully considering a large amount of surrounding information, and providing a powerful and effective tool for solving highly complex path planning problems.

DOI: 10.54254/2755-2721/111/2024CH0119

**Table 1.** The average success rate in different fields in a simple environment

| Experimental group | Average success rate(%) | | |
|---|---|---|---|
| | 4-neighborhood | 8-neighborhood | 24-neighborhood |
| 1 | 88.26 | 90.38 | 90.43 |
| 2 | 88.24 | 90.32 | 90.46 |
| 3 | 88.36 | 90.30 | 90.47 |
| Average value | 88.29 | 90.33 | 90.45 |

**Table 2.** The average time required in different fields in a simple environment

| Experimental group | Average time(ms) | | |
|---|---|---|---|
| | 4-neighborhood | 8-neighborhood | 24-neighborhood |
| 1 | 1.303 | 0.220 | 0.339 |
| 2 | 1.315 | 0.212 | 0.368 |
| 3 | 1.297 | 0.217 | 0.334 |
| Average value | 1.305 | 0.216 | 0.347 |

**Table 3.** The average path length in different fields in a simple environment

| Experimental group | Average path length | | |
|---|---|---|---|
| | 4-neighborhood | 8-neighborhood | 24-neighborhood |
| 1 | 39.00029 | 21.45656 | 11.19382 |
| 2 | 39.00031 | 21.46149 | 11.19542 |
| 3 | 39.00043 | 21.45649 | 11.19627 |
| Average value | 39.00034 | 21.45818 | 11.19517 |

First of all, in terms of average success rate (table 1), it shows a gradual upward trend. The specific data shows that it increases from 88.29% in the four-neighborhood to 90.33% in the eight-neighborhood, and then to 90.45% in the twenty-four neighborhood. This phenomenon clearly shows that in simple scenarios, expanding the search neighborhood can effectively improve the possibility of finding a path. The internal mechanism is that more neighborhoods enable the algorithm to search in a wider range of directions, thereby significantly increasing the chance of finding a feasible path. However, it is worth noting that the increase in success rate from eight neighborhoods to twenty-four neighborhoods is relatively small. This fully shows that in simple scenarios, four neighborhoods already have a relatively high success rate basis, and further expanding the neighborhood has a gradually diminishing effect on improving the success rate.

Secondly, in terms of average time (table 2) from four neighborhoods to eight neighborhoods, the average time drops significantly, from 1.305 milliseconds to 0.214 milliseconds; but from eight neighborhoods to twenty-four neighborhoods, the average time increases slightly to 0.3477 milliseconds. The possible reason for this situation is that in simple scenarios, eight neighborhoods can find relatively better paths more efficiently, thereby reducing unnecessary search time; while although twenty-four neighborhoods have a wider search range, they also bring more computational complexity, resulting in a slight increase in time.

Finally, in terms of average path length (table 3), it is significantly shortened as the neighborhood expands. The specific data is reduced from 39.00034 in four neighborhoods to 21.45818 in eight neighborhoods, and then to 11.19517 in twenty-four neighborhoods. This is because more neighborhoods provide more path choices for the algorithm, enabling the algorithm to find a path closer to a straight line, thereby greatly reducing detours and ultimately effectively shortening the path length.

DOI: 10.54254/2755-2721/111/2024CH0119

**Table 4.** The average success rate in different fields in a simple environment

| Experimental group | Average success rate(%) 4-neighborhood | 8-neighborhood | 24-neighborhood |
|---|---|---|---|
| 1 | 62.11 | 75.01 | 77.87 |
| 2 | 62.06 | 75.38 | 77.84 |
| 3 | 62.14 | 75.30 | 77.76 |
| Average value | 62.10 | 75.23 | 77.82 |

**Table 5.** The average time required in different fields in a simple environment

| Experimental group | Average time(ms) 4-neighborhood | 8-neighborhood | 24-neighborhood |
|---|---|---|---|
| 1 | 0.482 | 0.165 | 0.255 |
| 2 | 0.546 | 0.174 | 0.260 |
| 3 | 0.567 | 0.167 | 0.255 |
| Average value | 0.532 | 0.169 | 0.257 |

**Table 6.** The average path length in different fields in a simple environment

| Experimental group | Average path length 4-neighborhood | 8-neighborhood | 24-neighborhood |
|---|---|---|---|
| 1 | 39.11089 | 23.03074 | 11.67430 |
| 2 | 39.10957 | 23.02242 | 11.6752 |
| 3 | 39.10569 | 23.02879 | 11.6729 |
| Average value | 39.10872 | 23.02732 | 11.67413 |

In terms of average success rate (table 4), it shows a gradual upward trend as the neighborhood expands. The specific data shows that it increases from 62.10% in the four-neighborhood to 75.23% in the eight-neighborhood, and then to 77.82% in the twenty-four neighborhood. Compared with simple scenarios, the success rate in complex scenarios is generally at a relatively low level. This is mainly because in complex scenarios, the number of obstacles increases and their distribution is more complex, which undoubtedly brings great difficulties to path search. In the process from expanding from four neighborhoods to eight neighborhoods, the improvement in success rate is more significant. This fully shows that in complex scenarios, expanding the neighborhood plays a more prominent role in improving the success rate. The internal logic is that more neighborhoods enable the algorithm to search in a wider range of directions, thereby increasing the possibility of finding a feasible path. Especially in a complex environment with many obstacles, this advantage is more obvious.

In terms of average time (table 5), the average time in complex scenarios is generally longer than that in simple scenarios. This phenomenon occurs because the increase in obstacles makes the algorithm need more calculations and judgments when searching for a feasible path. From four neighborhoods to eight neighborhoods, the average time decreases significantly, from 0.532 milliseconds to 0.169 milliseconds; while from eight neighborhoods to twenty-four neighborhoods, the time increases slightly to 0.257 milliseconds. This has a certain similarity to the time change trend in simple scenarios. The reason may be that eight neighborhoods can improve search efficiency to a certain extent in complex scenarios and reduce unnecessary search time; while twenty-four neighborhoods, although having a wider search range, also bring more computational complexity, resulting in a slight increase in time.

Regarding the average path length (table 6), consistent with simple scenarios, it shortens as the neighborhood expands. The specific values are reduced from 39.10872 in four neighborhoods to 23.02732 in eight neighborhoods, and then to 11.67413 in twenty-four neighborhoods. However, in complex scenarios, the path length is relatively long. This is because the existence of obstacles makes the algorithm need to consider more about how to avoid obstacles when finding a path, thereby

DOI: 10.54254/2755-2721/111/2024CH0119

increasing the difficulty of finding a short path. In complex scenarios, the algorithm needs to weigh and choose among many obstacles to find a path that is both feasible and relatively short, which undoubtedly increases the complexity of path planning.

## 5. Discussion

In this study, an in-depth discussion was conducted on the path planning of driverless cars based on SLAM data under different search neighborhoods. The following discussion will be carried out from three aspects: the causes of research results, extended suggestions, and analysis of limitations.

### *5.1. Reasons for the formation of research results.*

Average success rate: In both simple and complex scenarios, as the search neighborhood expands, the average success rate shows an upward trend. This is because more neighborhoods provide a wider search direction for the algorithm, increasing the chance of finding a feasible path. However, the success rate in complex scenarios is generally lower than that in simple scenarios. This is mainly because the increase in obstacles in complex scenarios makes path searches face greater difficulties. From four neighborhoods to eight neighborhoods, especially in complex scenarios, the success rate increases significantly, indicating that expanding the neighborhood has a more significant role in improving the success rate in complex environments. The reason is that more search directions help overcome the obstacle limitations in complex scenarios and thus find a feasible path.

Average time: In a simple scenario, the average time drops significantly from four neighborhoods to eight neighborhoods, while it increases slightly from eight neighborhoods to twenty-four neighborhoods. In a complex scenario, the average time is generally longer than that in a simple scenario. And it decreases from four neighborhoods to eight neighborhoods and increases slightly from eight neighborhoods to twenty-four neighborhoods, which is similar to the trend in a simple scenario. Eight neighborhoods can efficiently find a relatively better path in a simple scenario and reduce unnecessary search time. While twenty-four neighborhoods have a wider search range but bring more computational complexity. In a complex scenario, the increase in obstacles makes the algorithm need more time to search for a feasible path. Similarly, eight neighborhoods improve the search efficiency to a certain extent, and twenty-four neighborhoods cause a slight increase in time due to the increase in computational complexity.

Average path length: In both simple and complex scenarios, the average path length shortens as the neighborhood expands. This is because more neighborhoods provide more path choices. The algorithm can find a path closer to a straight line and reduce detours. In a complex scenario, the path length is relatively long because the existence of obstacles makes it more difficult to find a short path.

### *5.2. Extension and suggestions of research results*

First, in the path planning of driverless cars, the search neighborhood needs to be flexibly selected according to the complexity of the actual scene. In a simple scenario, a smaller search neighborhood can be considered to reduce the computational complexity. In a complex scenario, appropriately expanding the search neighborhood can improve the success rate, but attention should be paid to controlling the computational complexity to avoid excessive time consumption. Secondly, further research is needed to improve the efficiency of the algorithm in complex scenarios. This can be achieved by optimizing the algorithm structure and adopting more efficient search strategies to reduce the average time and path length and improve the performance of driverless cars in complex environments. Thirdly, combining other advanced technologies such as deep learning can enhance the perception and adaptability of driverless cars to complex environments for better path planning. Finally, in practical applications, the search neighborhood can be dynamically adjusted according to real-time environmental information to achieve more efficient path planning and improve the practicability and safety of driverless cars.

### *5.3. Analysis of research limitations and improvement*

This study has certain limitations. First, the research only considers a specific search neighborhood range. In the future, the research scope can be further expanded to explore the impact of a wider range of neighborhood settings on path planning. Secondly, the simulation of complex scenarios may not be comprehensive enough. In reality, complex scenarios may be more diverse. The authenticity of complex scenarios can be improved by adding more obstacle types and dynamic obstacles. In addition, this research mainly focuses on theoretical analysis and simulation experiments, and the scope of actual testing is limited. In the future, tests can be conducted on more actual driverless car platforms to verify the reliability and practicability of the research results. Finally, in the research, the influence of other factors on path planning, such as vehicle dynamics and traffic flow, is not considered. Subsequent research can comprehensively consider these factors to establish a more perfect path planning model.

## 6. Conclusion

Through research, this paper conducts an in-depth exploration of the path-planning strategy of driverless cars based on SLAM data, focusing on analyzing the efficiency and success rate of the A* pathfinding algorithm under maps of different complexity levels and different neighborhoods. The study found that map complexity and neighborhood size have a significant impact on the search success rate, average search time, and average path length of the A* pathfinding algorithm. For example, in a complex map with dense obstacles and tortuous paths, a larger neighborhood may help improve the search success rate but will increase the average search time; while in a simple map, a smaller neighborhood may be more efficient. Therefore, in the path planning of driverless cars, according to the actual map situation, various factors should be comprehensively considered, and the neighborhood size of the A* algorithm should be reasonably selected to achieve efficient and accurate path planning.

This paper provides important research methods and practical experience for the field of driverless car path planning. It makes up for the deficiency of research on the A* pathfinding algorithm under different map complexities and helps driverless car developers better understand and optimize path planning strategies and improve the safety and operating efficiency of driverless cars. At the same time, it provides new research ideas and directions for researchers in related fields.

The current research still has certain limitations in evaluating the performance of the A* algorithm when dealing with extremely complex maps and dynamically changing environments. Future research can be carried out in the following aspects: First, further explore adaptive path planning strategies so that driverless cars can dynamically adjust the parameters of the A* algorithm according to real-time environmental changes; second, combine advanced technologies such as deep learning and reinforcement learning to improve the adaptability and robustness of the algorithm in complex environments; third, deeply study the application of multi-sensor fusion technology in path planning and improve environmental perception accuracy to provide more accurate decision-making basis for driverless cars. Through these efforts, promote the continuous development of driverless car path planning technology.

## References

[1] Li A J Li S M Li D R 2013 On the trajectory plannings key technologies for intelligent vehicle Mechanical Science and Technology for Aerospace Engineering vol 32 no 7 pp 1022-1026
[2] Dissanayake M Newman P Clark S et al 2013 A solution to the simultaneouslocalization and map building SLAMproblem IEEE Trans Ra vol 17 no 3 pp 229-241
[3] Xie J Sang C Wang S et al 2020 Overview of the research and application prospects of intelligent following mobile robot Automation in Manufacturing Industry vol 42 no 10 pp 49-55
[4] Hu X et al 2018 Dynamic Path Planning for Autonomous Driving on Various Roads with Avoidance of Static and Moving Obstacles Mechanical Systems and Signal Processing pp 482–500

DOI: 10.54254/2755-2721/111/2024CH0119


[5] Fatemidokt H Kuchaki Rafsanjani M 2018 F-Ant an effective routing protocol for ant colony optimizationbased on fuzzy logic in vehicular ad hoc networks Neural Computing and Applications vol 29 no 11 pp 1127-1137

[6] Zuo L Guo Q Xu X Fu H 2015 A hierarchical path planning approach based on a⁄ and least-squares policy iteration for mobile robots Neurocomputing vol 170 pp 257–266

[7] Li W 2021 It will take time for the driverless bus industry to flourish Intelligent and Connected Vehicles vol 01 pp 34-35

[8] Fan S Zhang H 2022 Research on vehicle local path planning based on the improved artificial potential field method Journal of Qingdao University Engineering Technology Edition vol 37 no 01 50-57

[9] Tao H Wu H Cheng L et al 2010 Research on autonomous navigation of EKF-SLAM and Fast SLAM algorithms based on feature maps Journal of Beijing Union University Natural Science Edition vol 24 no 02 pp 18-24

[10] Ma Q Li Q Wang W Zhu M 2024 LIDAR-based SLAM system for autonomous vehicles in degraded point cloud scenarios dynamic obstacle removal Industrial Robot vol 51 no 4 pp 632-639

[11] Kendall A Hawke J Janz D et al 2019 Learning to Drive in a Day 2019 International Conference on Robotics and Automation ICRA

[12] Chen Q Zheng Y Jiang H et al 2021 Dynamic path programming based on neural network improved particle swarm algorithm Journal of Huazhong University of Science and Technology Natural Science Edition vol 49 no 02 pp 51-55

[13] Liu S Zhou S Miao J Shang H Cui Y Lu Y 2024 Autonomous Trajectory Planning Method for Stratospheric Airship Regional Station-Keeping Based on Deep Reinforcement Learning Aerospace vol 11 no 9 pp 753

[14] Meng Q Qu Q Chen K Yi T 2024 Multi-UAV Path Planning Based on Cooperative Co-Evolutionary Algorithms with Adaptive Decision Variable Selection Drones vol 8 no 9 pp 435