An Improved Path Planning Algorithm for A* Guiding Robot Based on Indoor Scenes

Chuanjiang Li, Chenming Wang, Zhijian Zhang and Ziming Tom Qi, Senior Member, IEEE

***Abstract*—This paper presents a smooth trajectory A* path planning algorithm to improve smooth trajectory and stable movement of a guiding robot in the indoor scenes, the improved path planning algorithm is made in result the trajectory is smoother by giving priority to generating nodes in the same direction as the original path and avoiding generating adjacent child nodes in the diagonal direction with obstacles. On the other hand, the computation time is reduced by removing the CLOSE table and judging by the g value of the BestNode in the algorithm. Ultimately, the improved algorithm is integrated into Robot Operation System (ROS). The simulation and realistic navigation experiment illuminated that the improved A* algorithm can reduce the movement time of the robot and optimize path planning more secure and smoother.**

Index Terms*—Guiding robot, Improved A* algorithm, Laser radar, Navigation system, ROS.

I. Introduction

Recently robots and artificial intelligence became research hotspot again [1]. Autonomous mobile robots are a popular research topic in the field of robotics where navigation is one of the key research questions [2]. With the development of advanced sensors, mobile robots have also been widely studied. Mobile robots are able to understand their operational environment and their operational states based on sensing technologies and move autonomously toward the target in an obstructed environment to complete the assigned task [3]. Path planning is one of the important research questions for the intelligent mobile robot in autonomous navigation [4][5]. Various rules are set to ensure that mobile robots can move avoiding any collisions [6]. Trajectory planning is to find a feasible non-collision trajectory for the mobile robot in the workspace with obstacles. Then according to one or more criteria such as time, distance or energy to optimize path planning algorithms in intelligent robot and automobile navigation [7], such as the rapidly exploring random trees algorithm (RRT) [8], the ant colony optimization (ACO) algorithm [9], Gelasius bio-inspired neural network algorithm (GBNN)[10]. Contrasted to these path planning algorithms, standard A* algorithm had the advantages in simple principle and easy implementation, but the actual application scenarios were complex and changeable. In order to improve the practicability of A* in certain scenarios, many improved A* algorithms had been proposed [11] [12] [13] [14]. For instances, a cellular decomposition algorithm of coverage path planning based on the grid map was presented. According to the counter-clockwise direction, A* algorithm of global planning based on the grid and V-graph environmental model are used in the sub-region's connection. Therefore, it is very valuable for the standard A* algorithm to make some improvements when a robot moves in the indoor scenes.

This paper presents a novel real-time, low cost and smooth method of path planner based on ROS combining with the characteristics of the modern indoor scenes. Comparing the performance of traditional algorithm and A* algorithm, the advantages of A* algorithm is illustrated and explained in detail. This was mainly reflected in removing the CLOSE table and setting up the reward and punishment mechanism.

II. Overall Design

The design of a guiding robot navigation system based on ROS was completed based on hardware and software development. The software part adopted ROS which had the advantages of cross-platform, reusable code and distributed deployment. In the hardware part, real-time positioning and path planning for guiding robot was conducted by the laser radar, encoder and inertial measurement unit (IMU).

A. Hardware development

The hardware design of guiding robot navigation system was mainly composed of three parts: power supply module, sensor module and controller module. Sensor module contained laser radar, IMU, camera, microphone, motor with encoder, infrared and ultrasonic sensors. Controller module included a self-designed circuit board based on imbedded CPU STM32F103, Windows industrial personal computer (IPC) and Linux IPC. The hardware development of guiding robot navigation system based on ROS and laser radar is demonstrated as shown in Fig.1.

This paper was submitted for review on June 3 2022. This work was supported by the Shanghai Science and Technology Foundation of China under Grant No. 16070502900.

Chuanjiang Li is with College of Information, Mechanical and Electronic Engineering, Shanghai Normal University, Shanghai, China. (e-mail: licj@shnu.edu.cn).

Chenming Wang is with College of Information, Mechanical and Electronic Engineering, Shanghai Normal University, Shanghai, China. (e-mail: 1956472138@qq.com).

Zhijian Zhang is with College of Information, Mechanical and Electronic Engineering, Shanghai Normal University, Shanghai, China. (e-mail: 1552902795@qq.com).

Ziming Tom Qi with New Zealand Maritime School, Manukau Institute of Technology, Auckland, New Zealand. (e-mail: tom.qi@manukau.ac.nz).

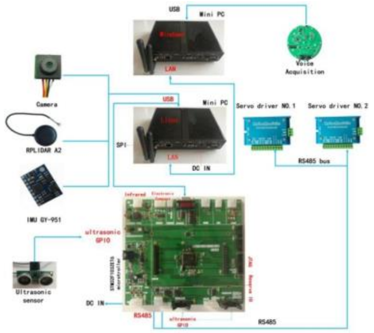
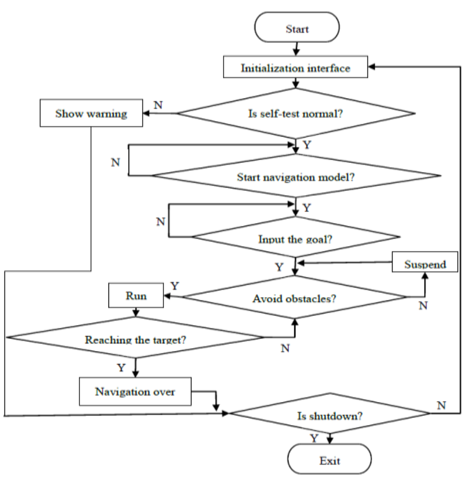


Figure 1 Guiding robot hardware diagram

As for the functions of each hardware, laser radar could detect obstacles of about 10 meters and locate itself while ultrasonic sensor could detect about 2 meters. The camera sensor was used to control speed for tracking. Encoder and IMU provided mileage and attitude to build the motion model of the robot through the track deduction algorithm [15]. Microphone collected sound data to build a man-machine dialogue. The STM32F103 board was used to collect sensor data, control motor, communicate with Linux IPC, and process the feedback information of ultrasonic sensors. Finally, human-computer interface was carried out by Windows IPC.

B. Software Development

The sensor data was visually displayed to allow the users to see the state of the guiding robot. To ensure the users safety and improve stability of guiding robot, initial self-test is applied to check all the sensors function before the robot performs navigation tasks. In case the initial self-test failed the robot stops moving with an alert. The design diagram of the guiding robot human-computer interaction system is shown in Fig.2.



(a)

(b)

Figure 2 The image (a) is the flowchart of guiding robot human-computer interaction, and (b) is the navigation user interface

III. PATH PLANNING ALGORITHM

A. Standard Path Planning Algorithm

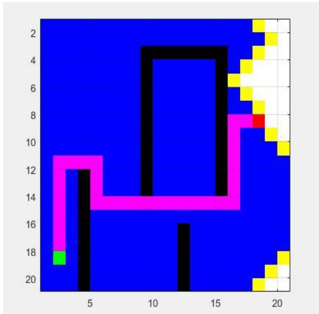
Traditional path planning algorithms mainly includes Dijkstra, RRT (Rapidly exploring Random Tree) and A*. Dijkstra is a typical shortest path algorithm, which is used to calculate the shortest path from one node to another node [16] [17]. Its main feature is to extend from the starting point (breadth first search ideas) to the end point [18]. RRT searches the blank area through random sampling points in the state space to plan the path between the starting point and the target point, which is suitable for the path planning of multi-dof robots in complex and dynamic environments [19] [20]. The standard A* algorithm is a heuristic path planning algorithm [21] [22]. Through the evaluation function to guide the search direction and calculate the value of the evaluation function of each node. Then select the lowest cost node as the extension node. Repeat the process until the final node is found [23].

The general form of the evaluation function in A* algorithm is:

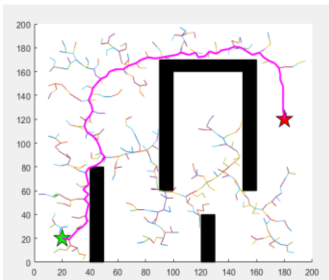
$$f(n) = h(n) + g(n) \tag{1}$$

where $g(n)$ represents the actual cost from the start node S_0 to the current node S_n, $h(n)$ represents the estimated cost from the current node S_n to the target node S_t.

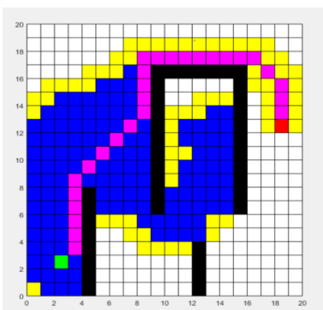
Fig.3(a), (b) and (c) shows the results of Dijkstra, RRT algorithm and A* algorithm respectively under the same environment simulation. The green and red dots in the figure represent starting and end point respectively. The black area is the obstacle, the blue area and branches of the random tree are all the searched areas when planning the path, and the pink line is the planned path in the end. According to the comparison in Fig.3, A* algorithm is much better than the other two algorithms in terms of both search range and path planning speed.



(a)



(b)



(c)

Figure 3 effect diagram of path planning under different algorithms

B. Improved A Algorithm*

The standard A* algorithm generally only considers the surrounding 8 nodes or 24, 48 adjacent nodes expansion sub-nodes in the path planning, and the generated nodes are equal in cost [24]. The minimum turning radius of guiding robot, width information and trajectory characteristics of the guiding robot motion is not considered in an indoor environment such as a shopping mall. In this paper, an improved A * algorithm is proposed to make it more suitable for indoor scene requirements of guiding robot path planning. The basic ideas were as follows:

(1) Considering the width information and trajectory characteristics of the guiding robot, a sub-node generation strategy of the reward and punishment mechanism was adopted. That is, nodes in the same direction as the original path were preferentially generated to avoid the adjacent child nodes of diagonal direction and obstacles. The methods as below:

a. In the case of obstacles in the horizontal or vertical direction, the guiding robot would not generate nodes near the obstacles. If there were fewer obstacles, the corresponding node would be generated.

b. If the vector direction of the temporary node and the current node was consistent with the direction of the parent node and the current node vector, the g value would be reduced. Otherwise no processing was performed.

c. Apply different strengths to the angles of different sizes of corners. The greater the turning force, the greater the penalty.

(2) For the Standard A* algorithm, it is impossible to apply in indoor scenarios such as factories efficiently and permanently. In other words, the standard A* algorithm creates the OPEN table and the CLOSE table. And the OPEN table stores the nodes that are not traversed, CLOSE table stores the nodes that have been traversed. When performing path planning, it is necessary to traverse all the nodes in the OPEN table. Normally, the same node often needs to be repeatedly calculated many times, resulting in an algorithm that is not efficient enough. The proposed improving A* algorithm used the grid map to establish an approximate optimal path, each node only needs to be accessed once without establishing a CLOSE table. Based on the g value of the BestNode to judge the processing and closing state of the path planning. The basic methods as below:

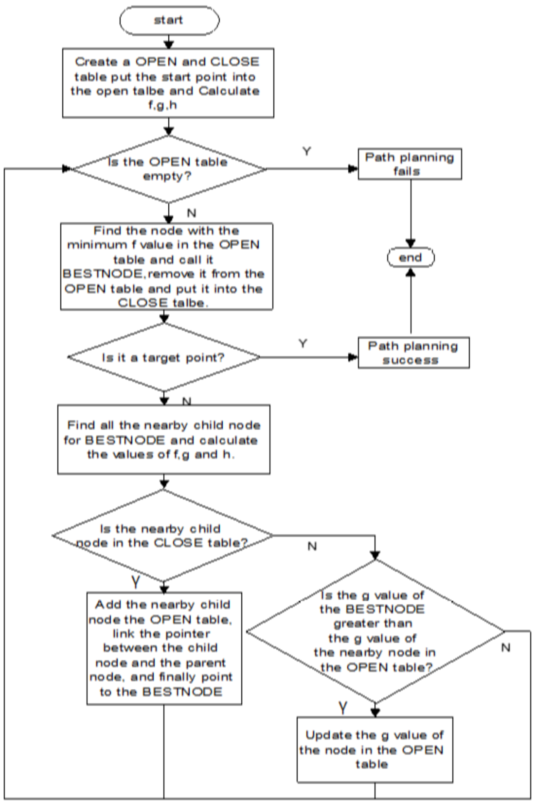
a. Establish an OPEN table, put the starting point into the OPEN table, record the g value of the starting point as zero, and the g value of other points as infinity, then calculate the h value and the "f" value.

b. Find the node with the lowest "f" value in the OPEN table as the BestNode and remove it from the OPEN table.

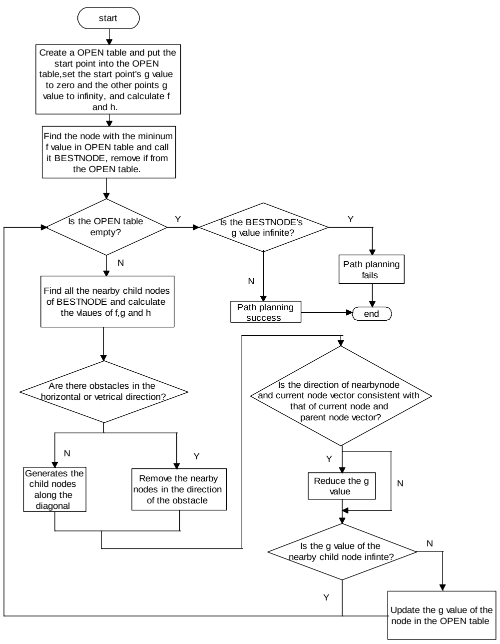
c. If the OPEN table was empty and the g value of the BestNode existed, that means the path planning was successful. If the OPEN table was not empty then find all neighboring child nodes of BestNode and calculate their "f" value, g value and h value.

d. If the g value of the neighboring child nodes were not existed, return to step c to determine whether the OPEN table was empty. If the g value of the neighboring child nodes existed, update the g value of the BestNode in the OPEN table with the g value of the neighboring child nodes. And join the neighboring child nodes into the OPEN table.

The flow chart of the standard A* algorithm and the improved A* algorithm flow chart are shown in Fig.4. This paper proposed a reward and punishment mechanism that preferentially generated nodes in the same direction as the original path, avoided the occurrence of adjacent child nodes in the diagonal direction with obstacles. This made the resulting trajectory smoother. On the other hand, removed the CLOSE table and judged the processing and closing state of path planning by the g value of BestNode, thus saving computation time.



(a)



(b)

Figure 4 Comparison between the standard A* algorithm and the improved A* algorithm. The image (a) was the flowchart of standard A* algorithm, and (b) the flowchart of improved A* algorithm

IV. EXPERIMENTAL RESULTS AND ANALYSIS

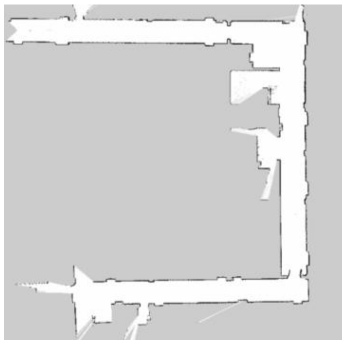
A. Experimental Platform and Environment

The experimental platform was a guiding robot which was independently developed by ourselves. The appearance and driving mechanism of the guiding robot were shown in Fig.5(a) and Fig.5(b). The actual navigation site was selected in the corridor of the second floor of Shanghai normal university Fengxian campus and its two-dimensional grid map, as shown in Fig.5(c) and Fig.5(d). Fig.5(e) represented the simulated scene which divided into 50 × 50 grids with a grid of length 1 m. Each grid represented a state where the robot could stay. The black regions were known obstacle locations while the white regions were known free space locations.

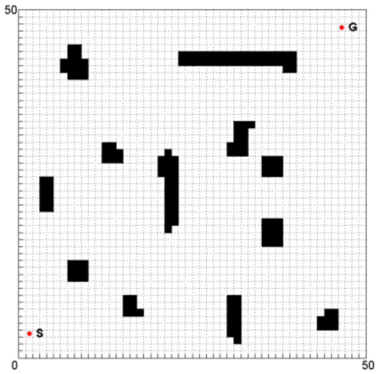
(a)

(b)

(c)



(d)

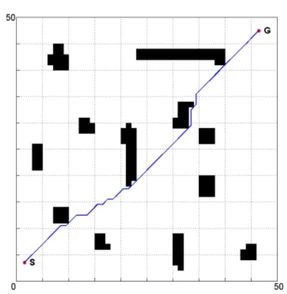


(e)

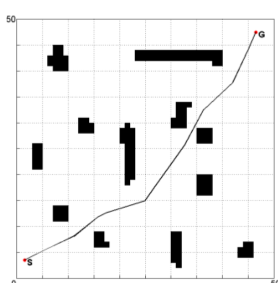
Figure 5 Experimental platform and site. The image (a) showed the appearance of the experimental platform, (b) was the guiding robot driving mechanism, (c) and (d) displayed the real-world experimental site, and its grid map, and (e) was the simulated and experimental environment

B. Simulation and Analysis

To demonstrate the practicability of A* path planning algorithm with smoother trajectory, this paper simulated on Matlab R2015b platform. The Matlab code was used for programming within the Windows 7 professional operating system and on a personal computer equipped with a 2.50 GHz Intel Pentium(R) Dual-Core E5200 CPU and 8 GB RAM. The simulation experiment was summarized in Fig6. S (3, 5) was the localization coordinates of the initial position and G (46, 47) was the localization coordinates of the goal position. The task target was to find out an optimal path of the guiding robot from S to G. To analyze the smooth performance, this paper adopted the method proposed in [25] to search for the optimal path. Finally, the experimental evaluation index was summarized in Table I.



(a)



(b)

Figure 6 Comparison between the standard A* algorithm and the improved A* algorithm. The image (a) was the flowchart of standard A* algorithm, and (b) the flowchart of improved A* algorithm

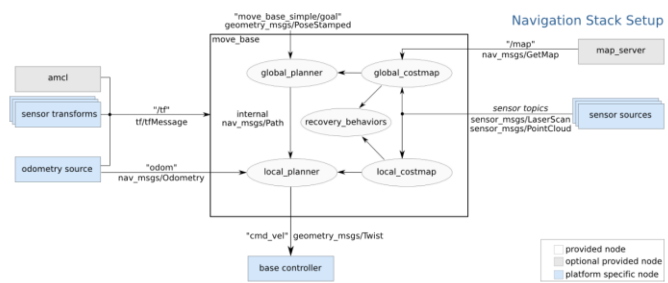
TABLE I
EVALUATION INDEX COMPARISONS OF THE STANDARD A* AND IMPROVED A*

Parameter	Standard A*	Improved A*
Path length (m)	66.15	66.45
Computation time (s)	0.36	0.31
Length of dangerous path (m)	30.83	10.16
Sum of turning angles (degree)	630.00	168.97

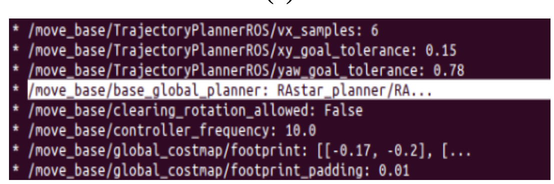
It could be concluded by analyzing Fig.6(a) that the trajectory planned by the standard A* algorithm had many turning points and it was close to the obstacles; accordingly, it was dangerous to the real-world navigation of the guiding robot. The Fig.6(b) demonstrated that the smooth trajectory planned by the improved A* algorithm had smaller turning angles and fewer turning points than the path planned by the standard A* algorithm. Table 1 summarized that the comparisons between standard A* algorithm and improved A* algorithm in evaluating performance indicators. Contrasted to the standard A* algorithm, although the improved A* algorithm only grew up the path length by 0.5%, it decreased the sum of turning angles, the length of the dangerous path, the computation time by 73.17%, 67%, and 13.8% respectively.

C. Integration in ROS and Analysis

To demonstrate the Practicability and effectiveness of the proposed path planners in real-world scenarios. The improved A * algorithm was integrated into ROS as a global path planner, replacing the default global path planner. The plugin method in ROS was used and integrated the improved A* algorithm into the original navigation framework. As shown in Fig.7.



(a)

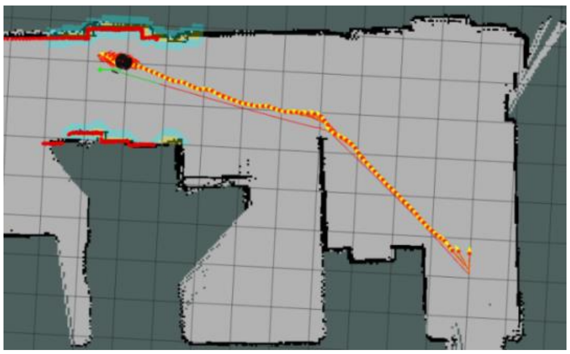


```
* /move_base/TrajectoryPlannerROS/vx_samples: 6
* /move_base/TrajectoryPlannerROS/xy_goal_tolerance: 0.15
* /move_base/TrajectoryPlannerROS/yaw_goal_tolerance: 0.78
* /move_base/base_global_planner: RAstar_planner/RA...
* /move_base/clearing_rotation_allowed: False
* /move_base/controller_frequency: 10.0
* /move_base/global_costmap/footprint: [[-0.17, -0.2], [...
* /move_base/global_costmap/footprint_padding: 0.01
```

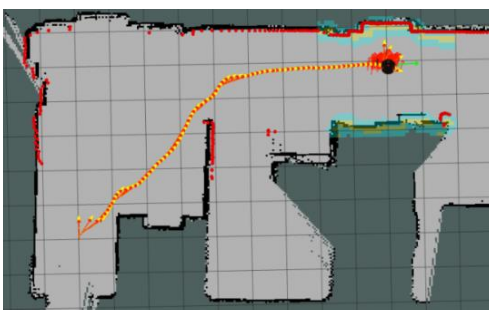
(b)

Figure 7 The image (a) is the move-base navigation stack integrated into ROS and (b) is the name of the global planner plugin

Considering the place of the real-world navigation experiment, we chose the corridor as the realistic navigation experiment site and used different corridor types to verify the real-world path planning effect. The short l-shaped corridor with a length of about 7 meters and the medium u-shaped corridor with a length of about 95 meters were selected. In the l-shaped corridor, the standard A* algorithm and the improved A* algorithm are respectively presented in Fig.8(a) and Fig.8(b). In the u-shaped corridor, the standard A* algorithm and the improved A* algorithm are respectively presented in Fig.8(c) and Fig.8(d). Finally, we respectively calculated the parameters of the l-shaped corridor and the u-shaped corridor, and summarized them into Table II.



(a)



(b)

(c)

(d)

Figure 8 Comparison between the standard A* algorithm and the improved A* algorithm. In the l-shaped corridor, the image (a) is navigation results of standard A*algorithm and (b) is navigation results of improved A*algorithm. In the u-shaped corridor, the image (c) is navigation results of standard A*algorithm and (d) is navigation results of improved A*algorithm.

TABLE II
EVALUATION INDEX COMPARISONS OF THE STANDARD A* AND IMPROVED A*

Type	Parameter	Standard A*	Improved A*
L-shaped corridor	Path length (m)	6.15	6.33
	Computation time (s)	0.033	0.027
	Length of dangerous path (m)	3.61	1.58
	Sum of turning angles (degree)	88.21	67.97
U-shaped corridor	Path length (m)	92.55	96.16
	Computation time (s)	0.51	0.43
	Length of dangerous path (m)	33.56	10.12
	Sum of turning angles (degree)	405.0	245.0

The planned path generated by the standard A* algorithm has created more turning points toward the destination. it is not a smoother path, and too close to obstacles. In contrast, the planned path designed by the improved A* algorithm keeps a certain distance from obstacles to ensure the safety of the robot. Table II shows the detailed parameters of the evaluation indicators for the four paths shown in Figure 8. In a U-shaped corridor with a length of 95 meters, although the improved A* algorithm increases the path length by 3.9%, it reduces the length of the dangerous path by 69.8%, the sum of the turning angles is reduced by 39.5%, and the calculation time is reduced by 15.7% %. In an L-shaped corridor with a length of 7 meters, although the improved A* algorithm increases the path length by 2.9%, it reduces the sum of the turning angle, the length of the dangerous path and the calculation time by 22.9%, 18.1% and 56.2%, respectively.

Through the above analysis and summary of experimental evaluation index, contrasted to the standard A* algorithm and the improved A* algorithm, it could be concluded that the improved A* algorithm find out a smoother trajectory and less computer time in indoor scenes for guiding robot and also could approve the practicality and feasibility of the improved A* path planning algorithm.

V.CONCLUSION

This paper presents a smooth trajectory A* path planning algorithm to improve smooth trajectory and stable movement of a guiding robot in the indoor scenes, the improved path planning algorithm is made in result the trajectory is smoother by giving priority to generating nodes in the same direction as the original path and avoiding generating adjacent child nodes in the diagonal direction with obstacles. On the other hand, the computation time is reduced by removing the CLOSE table and judging by the g value of the BestNode in the algorithm. Ultimately, the improved algorithm is integrated into Robot Operation System (ROS). Based on the results in simulation and realistic navigation experiment, the improved A* algorithm can reduce the movement time of the robot and optimize path planning more secure and smoother. The future work will be focused on robust design based on improved A* algorithm in the indoor scenes.

REFERENCES

[1] Wenxing Luo, Xu Deng, Feiqing Zhang, Youmei Wen and Dheyaa Jasim Kadhim, Positioning and guiding educational robots by using fingerprints of WiFi and RFID array. EURASIP Journal on Wireless Communications and Networking, (2018) 2018:170.

[2] Jianjun Ni, Liuying Wu, Xinnan Fan and Simon X. Yang, Bioinspired Intelligent Algorithm and Its Applications for Mobile Robot Control: A Survey. Computational Intelligence and Neuroscience, 2016, 1-16.

[3] Liping Wang. Automatic control of mobile robot based on autonomous navigation algorithm, Artificial Life and Robotics, 2019, Vol.24 (4), pp.494-498

[4] K. H. Sedighi, K. Ashenayi, T. W. Manikas and Heng-Ming Tai, Autonomous local path planning for a mobile robot using a genetic algorithm, Proceedings of the 2004 IEEE Congress on Evolutionary Computation, 2004,1338-1345.

[5] Hao Zhang, Xiaoyuan Zheng, Huaicheng Yan, Chen Peng, Zhuping Wang, Qijiun Chen. Codesign of event-triggered and distributed H∞ filtering for active semi-vehicle suspension systems, IEEE/ASME Transactions on Mechatronics, 22(2),2017, 1047-1058.

[6] Qing Wu, Hao Lin, Yuanzhe Jin, Zeyu Chen, Shuai Li, Dechao Chen, A new fallback beetle antennae search algorithm for path planning of mobile robots with collision-free capability, Soft Computing, 2020, Vol.24 (3), pp.2369-2380.

[7] Daqi Zhu, Man Mei and Bing Sun, The tracking control of Unmanned Underwater Vehicles based on Model Predictive Control, International Journal of Robotics & Automation, 32(4),2017, 351-359.

[8] Steven M. Lavalle, Rapidly-exploring random trees: A new tool for path planning, Algorithmic and Computational Robotics New Directions, 1998.

[9] Ming-Ru, Lu Xi, Ai-Min Xiao, The free step length ant colony algorithm in mobile robot path planning, Advanced Robotics, 30(23),2016, 1509–1514.

[10] Bing Sun, Daqi Zhu, Complete Coverage AUV Path Planning Based on GBNN Algorithm for Discrete and Centralized Programming, IEEE Transactions on Cognitive and Developmental Systems, 2018, DOI:10.1109/TCDS.2018.2810235.

[11] Cen Zeng, Qiang Zhang and Xiaopeng Wei, Robust algorithm for real-time route planning, IEEE Transactions on Aerospace and Electronic Systems, 36(3),2000, 869–878.

[12] Yan Bai, Zhichen Li, Congzhi Huang, Huaicheng Yan. New stability criteria for interval time-delay systems by improved integral inequalities, International Journal of Robotics & Automation, 32(5), 2017, 524-529.

[13] Hao Zhang, Xue Zhou, Zhuping Wang, Huaicheng Yan, Jian Sun. Adaptive consensus-based distributed target tracking with dynamic cluster in sensor networks, IEEE Transactions on Cybernetics, 2018, DOI: 10.1109/TCYB.2018. 2805717.

[14] Baoyan Zhu, Caihong Li, Li Song, Yong Song and Yibin Li, A* algorithm of global path planning based on the grid map and V-graph environmental model for the mobile robot. IEEE Chinese Automation Congress, 2017, 4973-4977.

[15] Daisuke Sekimori and Fumio Miyazaki, Precise Dead-Reckoning for Mobile Robots using Multiple Optical Mouse Sensors, Springer Netherlands, 2007, 145-151.

[16] Yan Xu,Guofei Guan,Qingwu Song,Chao Jiang,Lihui Wang. Heuristic and random search algorithm in optimization of route planning for Robot's geomagnetic navigation[J]. Computer Communications,2020.

[17] Wolfgang Fink,Victor R. Baker,Alexander J.-W. Brooks,Michael Flammia,James M. Dohm,Mark A. Tarbell. Globally optimal rover traverse planning in 3D using Dijkstra's algorithm for multi-objective deployment scenarios[J]. Planetary and Space Science,2019,179.

[18] Minhang Zhou, Nina Gao. Research on Optimal Path based on Dijkstra Algorithms. Proceedings of the 3rd International Conference on Mechatronics Engineering and Information Technology (ICMEIT 2019), 2019.

[19] Pérez-Higueras Noé,Jardón Alberto,Rodríguez Ángel,Balaguer Carlos. 3D Exploration and Navigation with Optimal-RRT Planners for Ground Robots in Indoor Incidents. [J]. Sensors (Basel, Switzerland),2019,20(1).

[20] Wei Kun,Ren Bingyin. A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm. [J]. Sensors (Basel, Switzerland),2018,18(2).

[21] Mobile Robot Path Planning Based on Ant Colony Algorithm With A* Heuristic Method[J]. Frontiers in Neurorobotics,2019,13

[22] Akshay Kumar Guruji,Himansh Agarwal,D.K. Parsediya. Time-efficient A* Algorithm for Robot Path Planning[J]. Procedia Technology,2016,23.

[23] Xie Lei,Xue Shuangfei,Zhang Jinfen,Zhang Mingyang,Tian Wuliu,Haugen Stein. A path planning approach based on multi-direction A* algorithm for ships navigating within wind farm waters[J]. ,2019,184(C).

[24] Zeng Cen, Zhang Qiang, Wei Xiaopeng. Robotic Global Path-Planning Based Modified Genetic Algorithm and A Algorithm, IEEE 2011 Third International Conference on Measuring Technology and Mechatronics Automation, 2011, 167-170.

[25] Hernández, Juan David, Mark Moll, Eduard Vidal, Marc Carreras and Lydia E. Kavraki. Planning feasible and safe paths online for autonomous underwater vehicles in unknown environments. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) ,2016, 1313-1320.

Chuanjiang Li received the MS. degree in department of electrical engineering from Henan University of Science and Technology, China in 2003, and Ph.D. degree in department of automation from Shanghai University, China in 2014. He is currently a professor in Shanghai Normal University. His current research interests include intelligent mobile robots, human robot interaction. (e-mail: licj@shnu.edu.cn).

Chenming Wang received his bachelor's degree from Shanghai Normal University TIANHUA College. He is currently a master student at Shanghai

Normal University and his current research focuses on control engineering, College of Information, Mechanical and Electronic Engineering, Shanghai Normal University, Shanghai, China. (e-mail: 1956472138@qq.com).

Zhijian Zhang received the MS. degree in engineering from Shanghai Normal University, Shanghai of China, in 2021.He is currently an Assistant Technical Manager with the BOC Financial Technology Co. Ltd. His current research interests and publications are in the areas of autonomous driving, autonomous navigation, big data, and artificial intelligence. (e-mail: 1552902795@qq.com).

Ziming Tom Qi Received the MPhil in Electronic Engineering from Waikato University, New Zealand, and PhD in Computer Engineering from Massey University, New Zealand, He is currently with New Zealand Maritime School, Manukau Institute of Technology, Auckland, New Zealand. Tom is a senior member of IEEE (e-mail: tom.qi@manukau.ac.nz).