RESEARCH AND DESIGN OF A PATH PLANNING USING AN IMPROVED RRT* ALGORITHM FOR AN AUTONOMOUS MOBILE ROBOT

THAI-VIET DANG

Mechatronics Department, School of Mechnical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

DOI: 10.17973/MMSJ.2023_10_2023051

viet.dangthai@hust.edu.vn

Path planning is innately viewed as a multi-objective optimization problem (MOOP) of the shortest path and the smallest distance to collision-free obstacles. The position of the mobile robot is definitively identified based on data derived from constructing maps with SLAM or extracting data from camera frames. Due to the complexity of the surroundings, obstacle avoidance still requires a complex sensor system with a high processing efficiency demand. The study provides an improved RRT* algorithm for mobile robot path planning in a given environment. RRT* will optimize the number of grid nodes based on the optimal cost functions of nodes in the searching environment. The improved RRT* algorithm with a safety cost increases the efficacy of obstacle avoidance. Hence, an improved RRT* algorithm based on the traditional RRT algorithm will remove redundant path nodes. Furthermore, the third-degree B-spline curve will smooth the path while ensuring MOOPs. Last but not least, simulations and experiments are shown to demonstrate the effectiveness of the suggested strategy.

KEYWORDS

RRT* algorithm, mobile robot, navigation, obstacle avoidance, semantic segmentation.

1 INTRODUCTION

The remarkable convergence of emerging technologies makes the Fourth Industrial Revolution an exciting time for the intelligent systems industries [La 2023]. Nowaday, autonomous mobile robots (AMRs) are indispensable to human existence. The applications of path-planning algorithms have been in the security, exploration, transportation, unmanned route optimization, etc… [Dang 2023, Tran 2023]. Path planning entails the generation of suitable obstacle-avoidance paths for mobile robot navigation between the origin and destination [Han 2017, Dang 2023]. Mobile robots have used planning techniques such as potential fields, visibility graphs, heuristic approaches, sampling-based, and grid-based methods [Noreen 2016].

Another planning tool suitable for low-dimensional space is grid-based methods. The A* algorithm was a heuristic search algorithm based on the Dijkstra algorithm [Shaher 2022]. A* [Xunyu 2020] is a well-known grid-based algorithm typically chosen to address planning difficulties for low-dimensional mobile robots. Dang et al. introduced the idea of optimal navigation based on an improved A* algorithm for AMRs [Dang 2023]. Moreover, Yonggang et al. developed an optimized A algorithm combined with the three-time Bezier curve to address the issues of numerous turning points, huge turning degrees, and lengthy running times [Yonggang 2022]. However, because grid edges confine the path, A* does not always find the shortest path. Also, its memory needs for difficult tasks grow Sampling-based Planning (SBP) methods exponentially [Alejo 2015], such as RRT* (Rapidly exploring Random Tree Star) [Eshtehardian 2022] are effective for solving high-dimensional complicated issues [Noreen 2016]. Slow convergence is a major shortcoming of sampling-based algorithms [Noreen 2016].

Among the numerous algorithms for motion planning, sampling-based techniques, such as PRM (Probabilistic Roadmaps) and RRT, were the most practical [Noreen 2016]. Lavalle et al. [Lavalle 1998] proposed the RRT method in 1998. These algorithms have advantages, such as suitability for high-dimensional problems and lower processing costs [Noreen 2016, Eshtehardian 2022, Lavalle 1998]. On the other hand, the RRT group is frequently one of the best options for single-query situations in static and dynamic environments. Typically, RRT*-based algorithms are suitable for finding mobile robot path. For the 2D finite space of an underground mobile robot, there is a greater likelihood of creating a route through tight points and curves, which is closer to underground specifications. Consequently, the RRT* method was chosen as the fundamental algorithm for this research.

After carefully examining the challenges above, the study introduces the improved RRT* algorithm with a safety coefficient. Refining the search point strategy and eliminating unneeded path points decrease the path length and computation scale. The algorithm improves the efficacy of obstacle avoidance and shortens the distance while maintaining efficiency, deriving a viable solution for mobile robots' path planning. The third-degree B-spline curve will be used to smooth the path of a mobile robot at steering angles, enhancing stability and track trajectory robustness [Wang 2021]. The following sections constitute the structure of the study. In Section 2, the improved RRT* algorithm is explained. Section 3 demonstrates the simulation results to prove the feasibility of the proposed method. In the last phase, conclusions are shown, and and further improving its performance.

2 PROPOSED METHOD

2.1 RRT algorithm*

The Rapidly exploring Random Tree (RRT) approach [Lavalle 1998] is the sample-based path planning strategy of random sampling to extend a tree with a root node at the starting node x_{int} in Fig. 1.

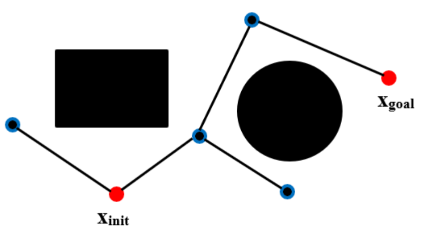


Figure 1. The RRT's path search algorithm.

RRT algorithm supports to non-holonomic constraints and a high degree of freedom. While generating a random sample in the configuration space, an attempt is built in order to connect at a local region separated by a predefined step length from the tree node with the step length closest to the random sample. If possible, tree connections exist. Expand the tree by adding nodes. As stated in the introduction, this sampling-based path planning algorithm uses randomly generated sample points to choose one path that can reach the objective as quickly as possible. However, RRT algorithm is difficult to optimize and ensure completely accuracy.

Therefore, the Rapidly-examining Random Tree Star (RRT*) algorithm [Noreen 2016] functions by rapidly exploring the configuration space of a system to find a path between a start and destination configuration. The initial step in employing the RRT* algorithm is determining the path planning problem's starting and ending points. Once the start and end points have been identified, the method randomly selects a node in obstacle-free space using the "sample function" at each iteration. This randomly selected node, designated x_{rand}, is then compared to the nearest node among the graph's nodes. Their distance is examined if there are no obstructions between xrand and the nearest node. If the Euclidean distance between the x_{rand} and the closest node is less than a constant value, a direct line is added to the graph, and x_{rand} is replaced with a new point, x_{new}. If the distance is larger than this constant amount, the algorithm utilizes the "steer function," as depicted in Fig. 2, to select a new point, x_{new}, on the straight line connecting from x_{rand} to the nearest node at a constant distance. This method is performed until a path from the start node to the goal node is discovered, or a certain number of iterations is attained.

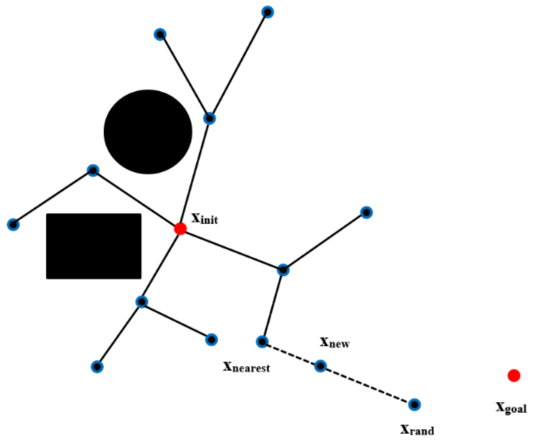


Figure 2. Steering function in the search process.

The "Parent function" is responsible for selecting the parent node, one of these processes. This function identifies the node with the lowest cost after connecting to the random node, x_{rand}, from among its neighbors. This procedure is depicted in Fig. 3.

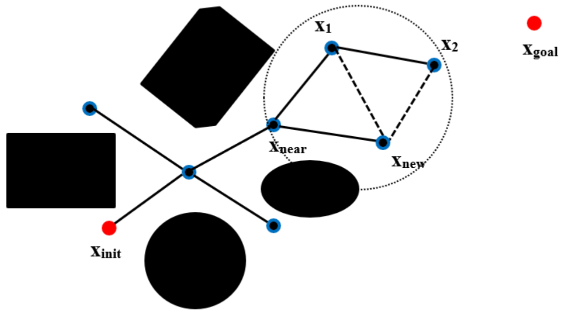


Figure 3. Re-select the parent node.

The "Rewire function" is the last stage of the RRT* algorithm. This function is responsible for rewiring the tree by examining the nearby nodes to determine whether a path traveling through x_{new} and one of the nearby nodes has a lower cost than the current path. If this is the case, the algorithm disconnects this node from its parent and links it to x_{new}. This procedure is depicted in Fig. 4.

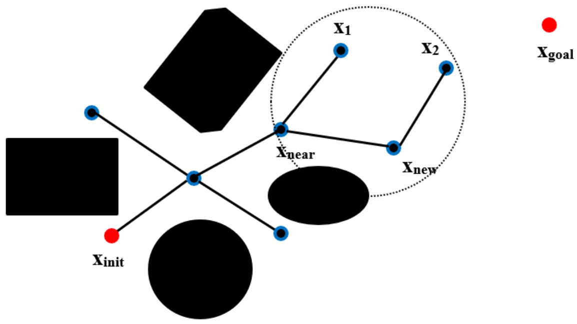


Figure 4. Pruning and re-writing process.

The Parent and Rewire functions enable the RRT* algorithm to efficiently explore the configuration space while considering the cost function. This strategy identifies the optimum path between the start and destination configurations.

2.2 Obstacle avoidance

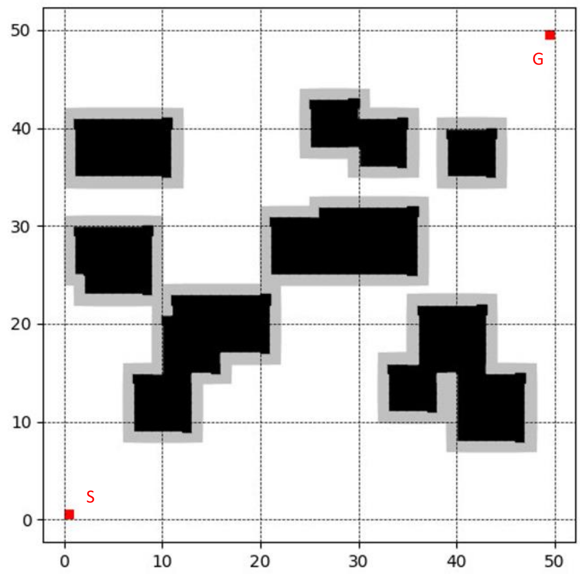


Figure 5. Obstacles with risk grey zones.

To ensure a safe collision distance from the obstacle, Fig. 5 introduces a new environment model with additional risk grey zones surrounding the black areas. A safety zone is the region of space surrounding an obstacle that an autonomous mobile robot (AMR) must not enter to avoid a collision. The obstruction's characteristics and the system's capabilities will decide the safety zone's size and shape. To avoid the smooth path colliding with the original obstacles, the distance between the obstacle's risk zones and the path point is measured to maintain the samples for the safe path. For instance, a complex-shaped obstacle or a limited-agile autonomous system may demand a larger safety zone. Hence, our AMR's path planning will be improved with suitable safety coefficients.

Beside, the traditional RRT algorithm will generate many nodes in a complicated environment, increasing computing size, memory usage, and inefficiency. The movement distance and computational scale will grow due to redundant path points. Therefore, removing redundant path nodes and smooth the path is vital while ensuring MOOPs.

2.3 Smooth RRT star algorithm

To address this issue, a technique is presented that reduces the number of redundant sample locations along the path while ensuring that the mobile robot avoids obstacles. The technique comprises utilizing the traditional RRT algorithm to design the path points. Then, connect the starting point x_{init} to the next path point x_{near} to determine whether having any obstacle between these two points. If there are no obstacles, x_{init} will be connected to the subsequent path point x_{1near}, and the process has been repeated. If having an obstacle, the initial path is the connection path between x_{init} and x_{near}, and the loop operation continues with x_{near} as the new point x_{1init} until the final destination is reached. In addition, if the path between x_{init} and x_{1near} does not cross through the obstacle, the redundant path point xnear is removed, and the mobile robot connects the subsequent path point until the end. Fig. 6 depicts the process of eliminating unnecessary path points. Combining the traditional RRT algorithm with obstacle identification and redundant path point reduction can considerably enhance the mobile robot's response time.

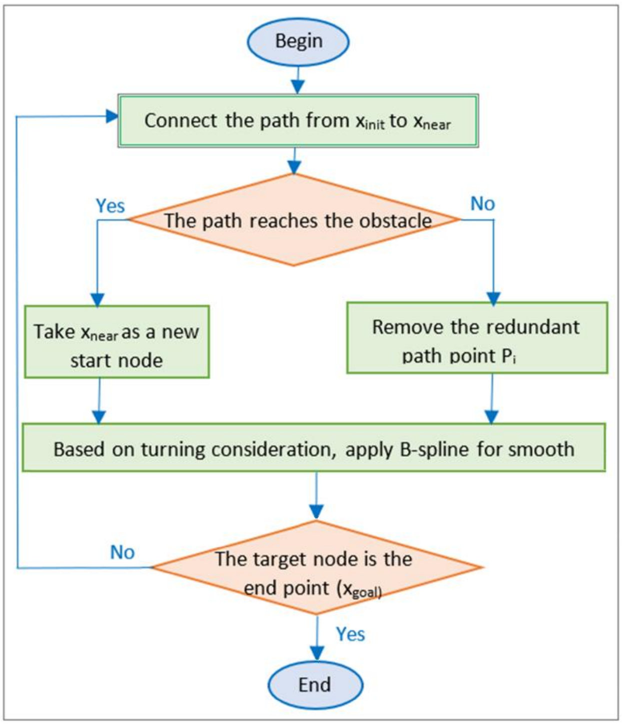


Figure 6. The flow chart of improved RRT* algorithm.

Smoothness in mobile robot route planning is the absence of sudden changes in direction or velocity along the path of a robot. Smooth roads are more efficient and easier to navigate for mobile robots. It may also be safer, as sudden changes in direction or speed may cause the robot to lose equilibrium or collide with obstacles. By lowering the frequency of sudden changes in direction or speed, these strategies can be utilized to smooth the path. This can be achieved by fitting the path to a smooth curve or by minimizing the "roughness" of the path using optimization techniques in Fig. 7.

To facilitate the mobile robot's navigation, a B-spline curve and a straight line will be employed to build a trajectory-creation algorithm. The third-degree B-spline curve is depicted by Equation 1 as follows :

$$B(t) = \sum_{i=0}^{4} N_{i3}(u) Pt_i. \tag{1}$$

where P_{ti} are the control path points, and $N_{i3}(u)$ are the B-spline basis functions of the third degree. Therefore, smoothness in mobile robot path planning is crucial to autonomous system design. It can improve the robot's motion efficiency, dependability, and safety.

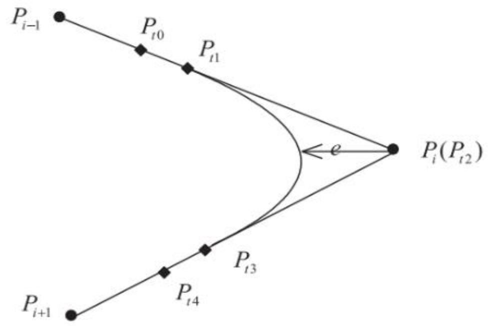
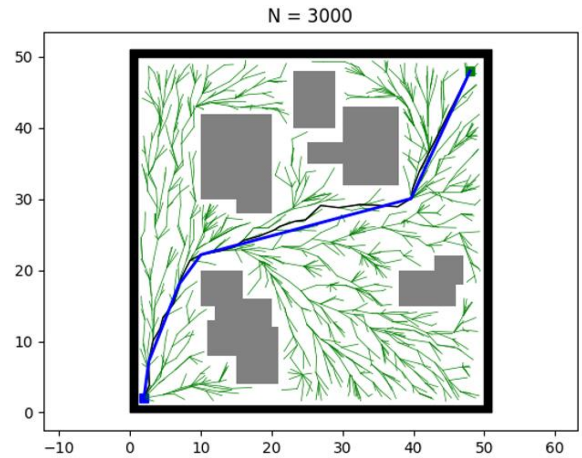


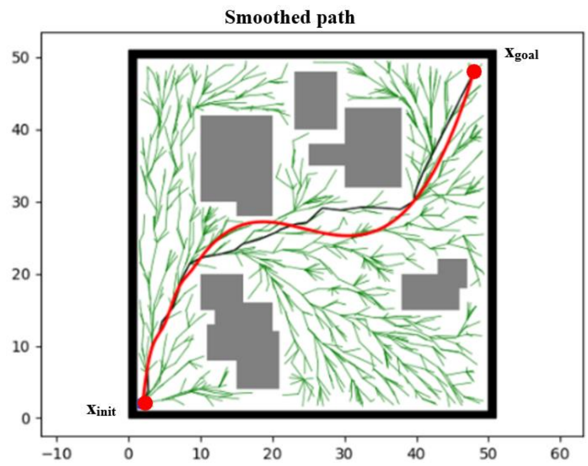
Figure 7. The B-spline transition curve.

3 SIMULATION RESULTS

Initially, the moving environment is constructed using a mobile robot outfitted with Lidar and a Jetson nano-embedded computer. To verify the effectiveness of the improved procedure, the following tests are performed: Two scenarios of 50x50 grid environments with different positions of obstacles are used for simulated test maps. In the moving environments, the start node x_{init} (0,0) and goal node x_{goal} (50,50) are utilized to compare the traditional RRT and improved RRT* algorithms with the number of interactions N=3000 in Figs. 8 and 9. Intel(R) Core(TM) I7-8750h CPU @2.20ghz 2.21ghz, Ram 8.00GB, 64-bit operating system, Windows 10 home English version, and Visual Studio 1.74 are applied for experimental simulation. Then, two scenarios are set up with five obstacles in the environment in Fig. 8 and more complex with six obstacles in Fig. 9. Next, using an improved RRT* algorithm, optimal path planning for a mobile robot is designed from start node x_{init} to goal node x_{goal} (see black lines). As a result of the findings, an revised path removing redundant path nodes will be developed (see blue lines). Finally, the B-spline curve of third degree is utilized to construct the smoothed trajectory (see red lines in Figs. 8b and 9b). Fig. 10 depicts the observation of the ROS mobile robot in an indoor setting.



(a)



(b)

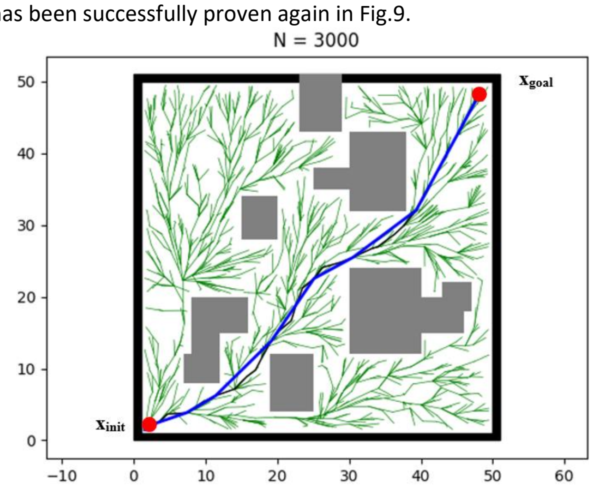
Figure 8. Mobile robot path planning from the start node x_{init} to goal node x_{goal} with five obstacles: (a) improve RRT* removing redundant path points, (b) proposed RRT* with smooth path algorithm.

In scenario 1, with five obstacles: the authors found completely the mobile robot path from the start node x_{int} to goal node x_{goal}. Certainly, the search processing time and length of the RRT* path is much smaller. Then, the RRT* path was continuously optimized by removing redundant path points and a smooth path algorithm (see red line in Fig. 8b). Table 1 proves the proposed RRT* performance is better than the RRT algorithm in the time processing and path length.

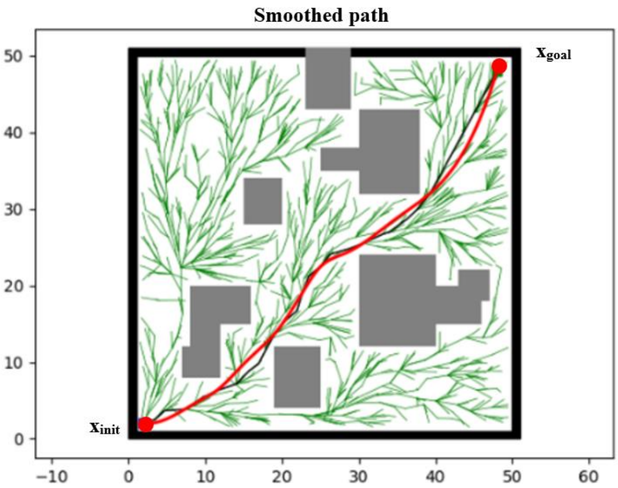
Method	Time processing	Path length
RRT [Noreen 2016]	28.82	1275.68
Improved RRT*	14.24	1325.23
Proposed RRT*	7.16	1012.16

Table 1. The comparison between traditional RRT, improved RRT* removing redundant path points, and proposed RRT* with removing redundant path points and smooth path algorithm in scenario 1.

When verifying the proposed RRT* algorithm, the authors change scenario 1 to be more complex in scenario 2 with six random obstacles. The performance of the proposed method has been successfully proven again in Fig.9.



(a)



(b)

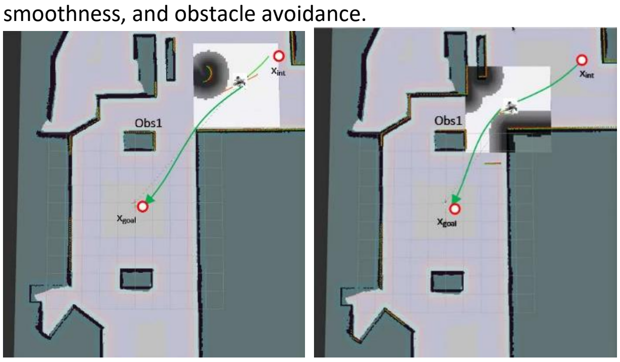
Figure 9. Mobile robot path planning from the start node x_{init} to goal node x_{goal} with six obstacles: (a) improve RRT* removing redundant path points, (b) proposed RRT* with smooth path algorithm.

Moreover, Table 2 illustrates that the proposed RRT* ensured the performance is better than the RRT algorithm in the time processing and path length.

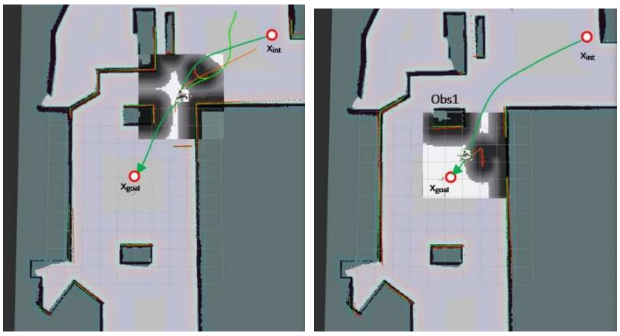
Method	Time processing	Path length
RRT [Noreen 2016]	29.12	1296.48
Improved RRT*	15.34	1315.13
Proposed RRT*	7.21	1022.16

Table 2. The comparison between traditional RRT, improved RRT* removing redundant path points, and proposed RRT* with removing redundant path points and smooth path algorithm in scenario 2.

In both scenarios presented in Figs. 8 and 9, the RRT* algorithm with a safety cost function (gray zones) guarantees a safe collision distance from the barrier (see the back path). In addition, the new blue path will be formed by removing redundant path nodes after removing redundant path points. A smoothing algorithm based on a third-degree B-spline curve will improve the performance of the steering angles of mobile robots. At the steering angles, the smoothing method permits the mobile robot to enter the risk region (gray). Trajectory design has continued to meet all MOOPs, including shortest length, smoothness, and obstacle avoidance.



(a) (b)



(c) (d)

Figure 10. The observation of the ROS mobile robot from the start node x_{init} to goal node x_{goal}.

Fig. 10, with four snapshots, depicts a mobile robot following a course determined by an improved smooth RRT* algorithm. In Fig. 10a, at the start point x_{int}, the global path is completely built from x_{int} to x_{goal} by traditional A* path planning. However, at the conner, if moving according to traditional A* path planning, AMR can be avoid because of being too close to the wall. Fig. 10b presents improved RRT* algorithm with safety coefficient to adjust the global path by connecting search local regions. Figs 10c and 10d prove the abilitie of AMR's to avoid Obs 1 and the right wall. With the support of local path search algorithm, AMR reaches to x_{goal} successfully. In addition, the search area's safe cost ensured the mobile robot's robust movement while avoiding impediments.

4 CONCLUSIONS

The paper proposes the improved RRT* algorithm by removing redundant path points and creating a smooth algorithm for mobile robot path planning. Path length and computing complexity are decreased by optimizing the search point technique and deleting redundant path points. In addition, the movement of mobile robots has been made safer by establishing risk zones around obstacles. A smoothed algorithm will enhance the performance of mobile robot steering angles based on a B-spline curve of the third degree. Simulations conducted in grid-based systems reveals the algorithm's advantages in terms of reduced memory consumption and optimization of computational calculation speed. Ultimately, path planning has been accomplished in an actual ROS environment. In addition, AMR's perception will be developed with sensor and camera systems to deal with MOOP of global and local path planning in complex environments.

ACKNOWLEDGMENTS

The authors express grateful thankfulness to Vietnam-Japan International Institute for Science of Technology (VJIIST), School of Mechanical Engineering, HUST, Vietnam and Shibaura Institute of Technology, Japan.

REFERENCES

[La 2023] La Thi Ngoc Anh et al. 2023 "Research MS Access Software for Application in Smart Garment Factory Machine Management." *MM Science Journal:* 6561-6567, https://doi.org/10.17973/MMSJ.2023_06_202304.

[Dang 2023] Dang, T. V., and Bui, N. T. 2023. "Multi-Scale Fully Convolutional Network-Based Semantic Segmentation for Mobile Robot Navigation." *Electronics* 12(3): 533. https://doi.org/10.3390/electronics12030533.

[Han 2017] Han, J., and Seo, Y. 2017. "Mobile robot path planning with surrounding point set and path improvement." *Application of Software Computer* 57: 35-47. https://doi.org/10.1016/j.asoc.2017.03.035.

[Tran 2023] Hoai-Linh Tran, Thai-Viet Dang. "An Ultra Fast Semantic Segmentation Model for AMR's Path Planning." Journal of Robotics and Control 4(3):424-430. https://doi.org/ 10.18196/jrc.v4i3.18758.

[Noreen 2016] Noreen, I., Khan, A. Habib, and Z. 2016. "Optimal Path Planning using RRT* Based Approaches: A Survey and Future Directions." *International Journal of Advanced Computer Science and Applications* 7: 97-107. https://doi.org/10.14569/IJACSA.2016.071114.

[Shaher 2022] Shaher, A., Sahbi, B., and Lioua, K. 2022. "Improved Dijkstra Algorithm for Mobile Robot Path Planning and Obstacle Avoidance." *Computers, Materials & Continua* 72(3): 5939-5954. https://doi.org/10.32604/cmc.2022.028165.

[Xunyu 2020] Xunyu, Z., Jun, T., Huosheng, H., and Xiafu, P. 2020. "Hybrid Path Planning Based on Safe A* Algorithm and Adaptive Window Approach for Mobile Robot in Large-Scale Dynamic Environment." *Journal of Intelligent & Robotic Systems* 99(2): 65-77. https://doi.org/10.1007/s10846-019-01112-z.

[Dang 2023] Dang, T. V., and Bui, N. T. 2023. "Obstacle Avoidance Strategy for Mobile Robot based on Monocular Camera." *Electronics* 12(8): 1932. https://doi.org/10.3390/electronics12081932.

[Yonggang 2022] Yonggang, L. et al. 2022. "A Mobile Robot Path Planning Algorithm Based on Improved A* Algorithm and Dynamic Window Approach." *Journal of Robotics* 2022(2): 1-9. https://doi.org/10.1155/2022/2183229. .

[Alejo 2015] Alejo, D., Cobano, J.A., Heredia, G., Martinez-de Dios, J.R., and Ollero, A. 2015. "Efficient Trajectory Planning for WSN Data Collection with Multiple UAVs." In: Cooperative Robots and Sensor Networks; Springer International Publishing: New York, NY, USA, vol. 604, pp. 53-75. https://doi.org/10.1007/978-3-319-18299-5_3.

[Eshtehardian 2022] Eshtehardian, S. A., and Khodaygan, S. 2022. "A continuous RRT* based path planning method for non-holonomic mobile robots using B spline curves." *Journal of Ambient Intelligence and Humanized Computing* 14(5): 8693-8702. https://doi.org/10.1007/s12652-021-03625-8.

[Noreen 2016] Noreen, I., Khan, A., and Habib, Z. 2016. "A Comparison of RRT, RRT* and RRT*-Smart Path Planning Algorithms." International Journal of Computer Science and Network Security 16: 20-27.

[Lavalle 1999] Lavalle, S.M. 1999. "Rapidly-Exploring Random Trees: A New Tool for Path Planning." Technical Report, Computer Science Dept., Iowa State University: Ames, Iowa, 1998.

[Wang 2021] Wang, W., Zhao, J., Li, Z., and Huang, J. 2021. "Smooth Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm." *Journal of Robotics* 2021: 1-10. *Journal of Robotics* 2021(3): 1-10. https://doi.org/10.1155/2021/4109821

CONTACTS:

Dr. Thai-Viet Dang

School of Mechanical Engineering/Hanoi University of Science and Technology, Department of Mechatronics

No 1 Dai Co Viet Rd. Hai Ba Trung, Hanoi, 10000, Vietnam

(+084) 0989458581, viet.dangthai@hust.edu.vn, https://www.researchgate.net/profile/Viet-Dang-Thai-2.