VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF MECHANICAL ENGINEERING

MM Science Journal | www.mmscience.eu
ISSN 1803-1269 (Print) | ISSN 1805-0476 (On-line)
Special Issue | TEAM 2024
Transdisciplinary and Emerging Approaches to Multidisciplinary Science
11.9.2024 – 13.9.2024, Ostrava, Czech Republic
DOI: 10.17973/MMSJ.2024_10_2024063

MM SCIENCE JOURNAL

TEAM2024-00003

AN IMPROVED COVERAGE PATH PLANNING FOR SERVICE ROBOTS BASED ON BACKTRACKING METHOD

TIEU-THANH LE[1], KIM-KHOI PHUNG CONG[1], THAI-VIET DANG[1]*

[1] Department of Mechatronic, School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam

*Corresponding author; e-mail: viet.dangthai@hust.edu.vn

Abstract

Service robots are of paramount importance in both industrial and residential environments. Nevertheless, these robots frequently encounter obstacles in the form of perilous elements present in their surroundings, which may result in their malfunction and hinder their optimal performance. Hence, autonomous robot applications heavily rely on coverage path planning algorithms, which facilitate the execution of area coverage operations in a streamlined and economical manner. The article presents a coverage path planning algorithm for service robots that is based on a backtracking procedure. The proposed method attempts to strike a balance between the safety of the cleaning automaton and the extent of coverage. Following straight-forward instructions, the mobile robot (MR) initially generates a spiral or zigzag trajectory. It examines each position for possible backtracking locations. When obstacles and visited positions obstruct the path, the improved A* algorithm is utilized to return to known unvisited regions. Subsequently, the robot proceeds to sweep the regions until no remaining back-tracking sites remain. Simulation of the method on the Robot Operating System (ROS) verifies that it is possible to restrict the movement of high-risk robots.

Keywords:

A* algorithm, coverage path planning, backtracking method, service robots, tracking trajectory

1 INTRODUCTION

Service robots will become an integral part of every facet of existence due to the fourth industrial revolution's accelerated progression [Dang 2023a]. Autonomous navigation, which is predominantly predicated on the effectiveness of path planning, has brought about a paradigm shift in the field of mobile robotics in recent times. Path planning is a critical component in directing MRs through dynamic and complex environments while maintaining a critical balance between safety and locating the quickest route [Dang 2023b]. Comprehensive coverage path planning (CCPP) is our primary area of emphasis. In contrast to point-to-point path planning, contextually dependent path planning (CCPP) strives to encompass all locations within a workspace, not merely the distance between an entry and exit point [Khan 2017]. Two distinct categories of algorithms are utilized for navigation: local path planning and global path planning [Dang 2023c].

Local path planning utilizes data pertaining to the robot's immediate vicinity to accomplish dynamic obstacle avoidance. In contrast, global path planning functions by utilizing pre-existing environmental information, which encompasses stationary obstacles. Researchers are primarily concerned with two aspects of global path planning: determining the shortest route and ensuring adequate path coverage [Dang 2023d]. A* and Dijkstra are two of the most widely recognized algorithms utilized to address the initial problem [Dang 2023e]. Breadth-first search traversal is utilized by the Dijkstra algorithm to ascertain the shortest path [Alshammrei 2022]. As an improvement over Dijkstra, the A* algorithm estimates the distance between the current position and the target using heuristics. As a result, A* offers solutions more rapidly and requires fewer memories than Dijkstra [Yong 2022]. The second problem has been the subject of numerous proposed works. Brownian motion is employed to generate paths at random for certain sweeping robots [González 2003]. González et al. proposed the method of assisting the robot in traversing the area, but it is extremely time-consuming and energy intensive. As an example, Guo et al. partitions the map into regional maps [Guo 2022]. A unique path planning approach is employed for each local map. Local locations are then covered by the robot in a predetermined sequence. Complication arises when implementing these methods on embedded systems that have limited memory and feeble processors [Dang 2023f, and Tran 2023].

The paper presents the straightforward retracing spiral algorithm. Spiral paths are utilized by the BSA algorithm to generate a coverage map and identify unvisited regions.

Unvisited regions are concealed by a mechanism that tracks backwards. Determining the shortest route back to unobstructed regions is a matter of considerable importance. Backtracking in BSA utilizes the depth-first search algorithm, which is not particularly effective. The paper describes the BSA algorithm by backtracking with A* and designing a zigzag path for this algorithm. For simulation, the Robot Operating System (ROS) and Turtlebot3 on Gazebo are utilized.

2 BACKTRACKING-BASED PATH PLANNING METHOD

It is presumed that the MR can only move in four directions during filling path execution: up, left, down, and right (Figure 1).

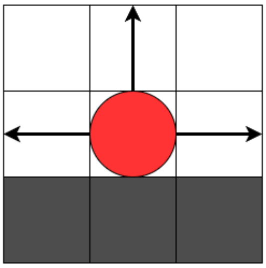


Fig. 1: MR's direction.

Upon being traversed, a grid cell may be regarded as an impediment (Figure 2).

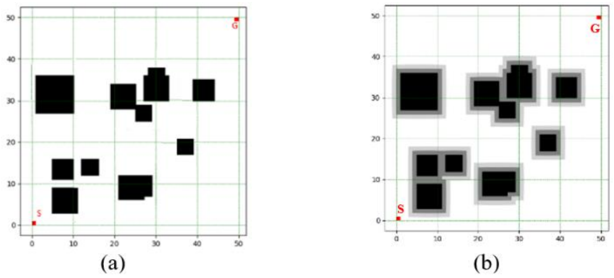


Fig. 2: The known environment is built with (a): a grid map containing obstacles without risk regions and (b): a grid map containing obstacles with risk region.

2.1 Spiral filling

BSA algorithm specifies four principles that, when followed, direct the robot to construct a spiral filling path. Prior to proceeding, the robot must establish the reference lateral side (RLS), which is the right or left side, and the opposite lateral side (OLS).

- Rule 1: IF obstacle_all_around
  THEN ending_spiral_point_detected
- Rule 2: IF NOT obstacle_in_RLS
  THEN turn_to_RLS and move_forward
- Rule 3: IF obstacle_in_front
  THEN turn_to_OLS
- Rule 4: OTHERWISE move_forward

The first rule is used in a while loop as a stop condition. The second and third rule helps the MR move along the obstacles' borders (Figure 3).

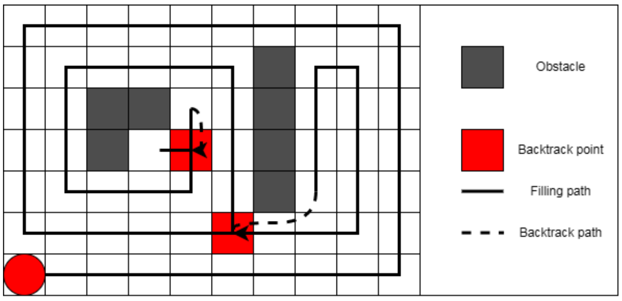


Fig. 3: Spiral filling

2.2 Zigzag filling

When applying the aforementioned spiral filling principles, special attention must be paid to the robot's current direction. This causes an inconvenience when the path must be generated prior to the initiation of the MR's movement. The regulations outlined below do not necessitate taking into account the orientation of the robot.

- Rule 1: IF obstacle_all_around
  THEN ending_zig_zag
- Rule 2: Move to the first free grid cell in map's counter-clockwise direction.

Attention should be given to the fact that if the robot prioritizes clockwise movement, the zigzag path will reverse direction (Figure 4).

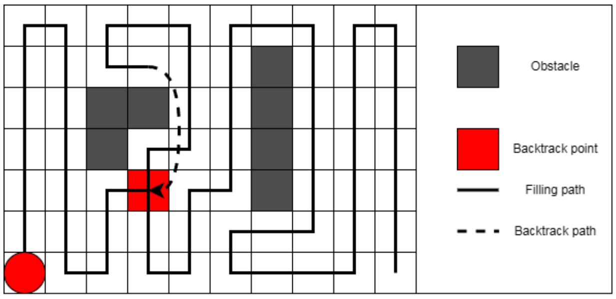


Fig. 4: MR's direction

2.3 Backtracking mechanism

The zigzag and spiral filling methods mentioned earlier are only capable of covering an entire area if the automaton begins at a corner of the map and there are no obstacles present. Occasionally, the automaton might encounter a circumstance in which there are multiple viable courses of action. In order to remedy this issue, the robot's current location will be marked as a backtrack point, and it will continue its intended path. The robot is subsequently able to return to the retrace point and resume its filling path operations. Bear in mind that once the robot completes its coverage of a given area, certain retrace points will be eliminated due to the inability to generate a feasible filling path.

2.4 Backtracking method based on A* algorithm

When returning to unvisited areas, the service automaton must take the shortest route possible in the grid-map (Figure 1). The A* algorithm is chosen as the solution to this problem. Heuristic cost is utilized by A* to direct its investigation. It modifies its f-value after exploring the adjacent nodes and selecting the grid with the lowest f-value, in (1). By repeating the procedure until the objective is achieved.

$$f(n) = g(n) + h(n), \qquad (1)$$

where n: the current grid; f(n): the total cost evaluation of n; h(n): the heuristic cost from the current grid to the goal; g(n): the actual cost from the current grid to the following grid.

While going to the backtrack points, the robot is allowed to move in eight directions (Figure 5). Therefore, the heuristic equation shown in (2) is used.

$$h(n) = D*(dx+dy)+(D_2-2*D)*\min(dx,dy), \quad (2)$$

where D is the length of each grid and D2 is the diagonal distance between two neighboring grids. dx and dy can be calculated as following (3):

$$dx = |x_E - x_S| \\ dy = |y_E - y_S|, \quad (3)$$

where is goal grid's coordinate and is start grid's coordinate.

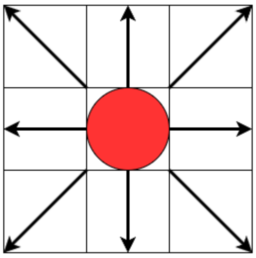


Fig. 5: MR's movement with eight directions.

3 SIMULATION RESULTS

The following experiment is conducted to ensure the feasibility of the MR's path planning (Figure 6).

Fig. 6: Three-wheeled MR

On a computer equipped with 16 GB Ram, Intel Core i5 - 11400H processor and Windows 11, a 30x30 grid map were created with start point S(0, 0). In Figure 7, the 30x30 grid map is represented as a two-dimensional array. Then, the MR tracked CCP algorithm with two scenarios of using the spiral path and the zigzag path.

Furthermore, the zigzag method generates a stream-lined trajectory requiring a reduced number of backtracking locations (Figure 8). Based on the same grid-map and starting and ending points, the spiral method creates three backtrack points whereas zigzag method only needs two backtrack points. Global path planner obtains information from the global cost-map of the navigation stack, which has irregular borders. Due to these zagged borders, the spiral method will create more backtrack points than zigzag method.

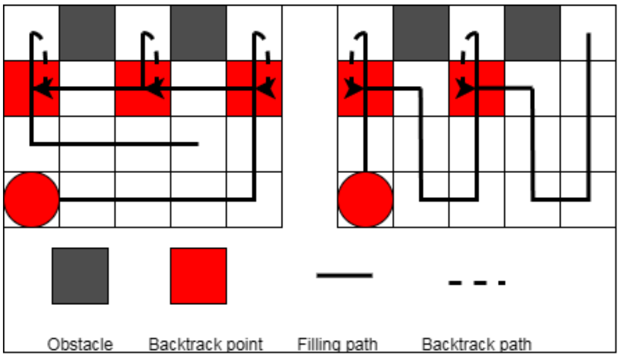


Fig. 8: The comparison of the spiral method zig-zag method based on the number of MR's back-track points in the same grid-map.

The data are illustrated using the colors white and black in Matplotlib. Random obstacles are generated in order to produce a variety of outcomes. To facilitate comprehension, the MR robot will be proportionate to a single grid cell. The subsequently generated contour trajectories, denoted by red lines, will be the result of backtracking calculations. The results indicate that both filing methods are capable of covering the entire map. Next, a simulation was performed on ROS utilizing Turtlebot3 and Gazebo. A simple environment comprises four static obstacles and an empty chamber (Figure 9).

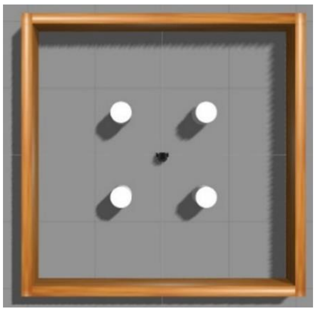


Fig. 9: Simulation environment with four static obstacles

In the beginning, SLAM is utilized to produce the grid map (Figure 10a). The generated map is saved as an image file, which can be accessed by ROS as an occupancy grid message. The map data is organized in row-major order, and the occupancy probabilities range from -1 to 100 (with -1 indicating positions that are undetermined). The generation of coverage paths will occur on grid cells that are entirely vacant. Following the loading of the map into ROS's navigation stack, a cost map featuring an inflation radius is superimposed on the map to prevent the MR robot from approaching the impediments too closely (Figure 10b).

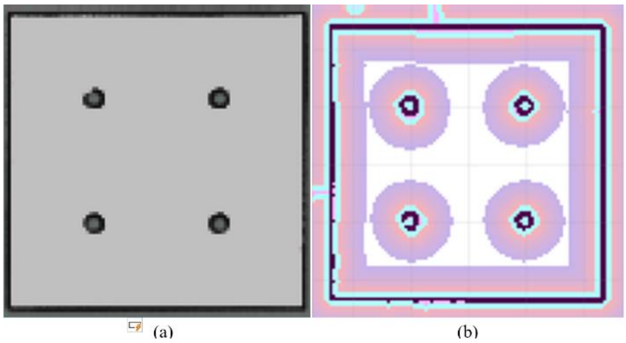


Fig. 10: Simulation environment after using (a): SLAM, and (b): inflation radius surrounding obstacles.

Due to the two-dimensional array format in which the map data is stored, the backtracking algorithm can be implemented with minimal efforts. Continuously, the generated path will be transmitted to the navigation stack via the move-base action of the navigation stack (Figure 11).

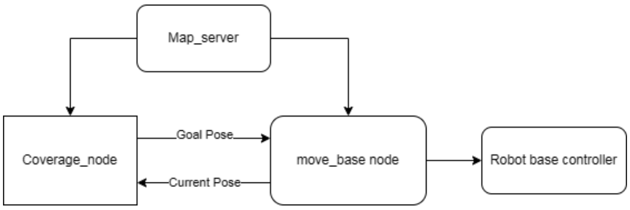


Fig. 11: MR's controller diagram

Turtlebot3 will also change its goal whenever it approaches a threshold relative to its current goal. The rendered path will be presented on RVIZ (Figure 12).

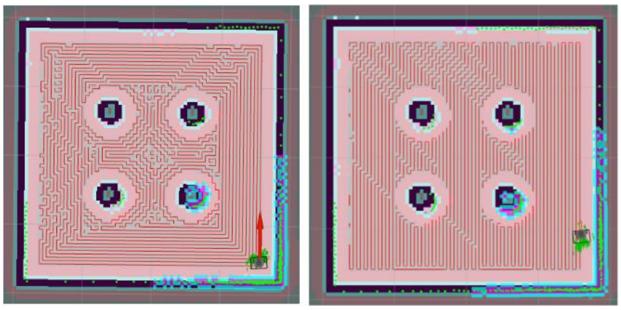


Fig. 12: Turtlebot3's path planning with Gazebo on ROS

The findings presented in Figures. 7, 8, and 12 indicate that the zigzag method produces an optimized trajectory that minimizes the quantity of backtracking locations required, as demonstrated by the ROS simulation.

4 CONCLUSIONS

The paper presents a coverage path algorithm that utilizes a backtracking procedure to enable robots to traverse static obstacles while thoroughly populating the map. Two filled trajectories were provided, along with backtracking using A*. The results generated from the simulations executed on the computer exhibit a notable capacity for rapidity and minimal memory consumption. Furthermore, the zigzag method produces an optimized trajectory that minimizes the quantity of backtracking locations required, as demonstrated by the ROS simulation. In the end, proposed effective coverage path planning has proven its feasibility in practical applications.

5 ACKNOWLEDGMENTS

This research is funded by Hanoi University of Science and Technology (HUST) under project number T2023-PC-008. Lab-506: Computer Vision and Autonomous Mobile Robot (CVAMS), VJIIST, HUST is gratefully acknowledged for providing work location, guidance, and expertise.

6 REFERENCES

[Alshammrei 2022] Alshammrei, S. et al. Improved Dijkstra Algorithm for Mobile Robot Path Planning and Obstacle Avoidance. Computers, Materials & Continua, 2022, Vol.72, No.3, pp 5939-5954. https://doi.org/10.32604/cmc.2022.028165

[Dang 2023a] Dang, T.V. and Bui, N.T. Multi-Scale Fully Convolutional Network-Based Semantic Segmentation for Mobile Robot Navigation. Electronics, 2023, Vol.12, No.3, pp 533. https://doi.org/10.3390/electronics12030533

[Dang 2023b] Dang, T.V. Research and design of a path planning using an improved RRT * algorithm for an autonomous mobile robot. MM Science Journal, 2023, Vol.10, pp 6712-6716. https://doi.org/10.17973/MMSJ.2023_10_2023051

[Dang 2023c] Dang, T.V. and Bui, N.T. Obstacle Avoidance Strategy for Mobile Robot Based on Monocular Camera. Electronics, 2023, Vol.12, No.8., pp 1932. https://doi.org/ 10.3390/electronics12081932

[Dang 2023d] Dang, T.V. et al. IRDC-Net: Lightweight Semantic Segmentation Network Based on Monocular Camera for Mobile Robot Navigation. Sensors, 2023, Vol.23, No.15, pp 6907. https://doi.org/10.3390/s23156907

[Dang 2023e] Dang, T.V. Autonomous mobile robot path planning based on enhanced A* algorithm integrating with time elastic band. MM Science Journal, 2023, Vol.10, pp 6717-6722. https://doi.org/10.17973/MMSJ.2023_10_2023052

[Dang 2023f] Dang, T.V., Nguyen, D.S. Optimal Navigation Based on Improved A* Algorithm for Mobile Robot. Intelligent Systems and Networks, 2023, LNNS, Vol.752, pp 574-580. https://doi.org/10.1007/978-981-99-4725-6_68

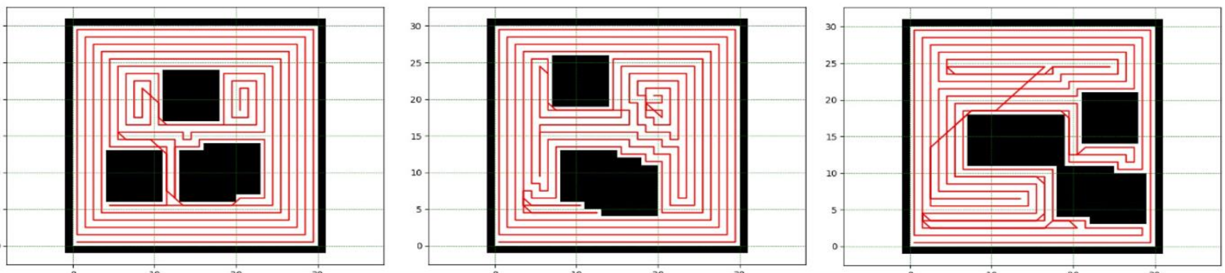
[González 2003] González, E. et al. BSA: a coverage algorithm, Intelligent Robots and Systems, 2003, (IROS 2003), Las Vegas, NV, USA. https://doi.org/10.1109/IROS.2003.1248885

[Guo 2022] Guo, S. et al. Path planning algorithm for sweeping robot full traversal cleaning area. 2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE), Qingdao, China (2022). https://doi.org/10.1109/ARACE56528.2022.00030

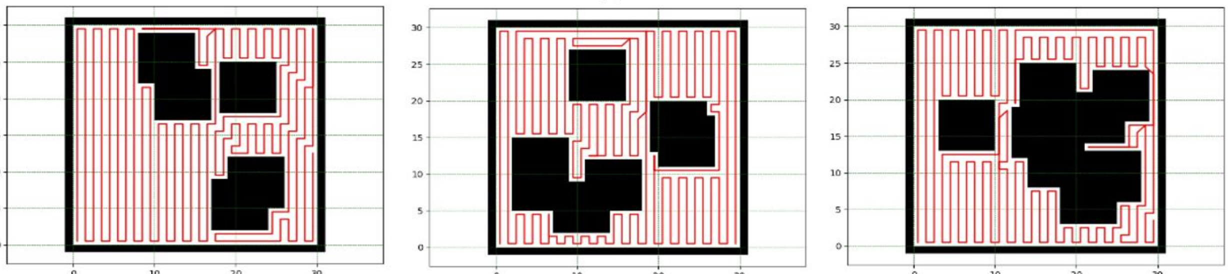
[Khan 2017] Khan, A. et al. On complete coverage path planning algorithms for non-holonomic mobile robots: survey and challenges. Journal of Information Science and Engineering, 2017, Vol.33, pp 101-121. https://doi.org/10.6688/JISE.2017.33.1.7

[Tran 2023] Tran, H.L., and Dang, T.V. An Ultra Fast Semantic Segmentation Model for AMR's Path Planning, Journal of Robotics and Control, 2023, Vol.4, No.3, pp 424-430. https://doi.org/10.18196/jrc.v4i3.18758

[Yong 2022] Yonggang, L. et al. A mobile robot path planning algorithm based on improved A∗ algorithm and dynamic window approach. IEEE Access, 2022, Vol.10, pp 57736-57747. https://doi.org/10.1109/ACCESS.2022.3179397



(a)



(b)

Fig. 7: Improved backtracking method-based MR's path planning with (a): the Spiral path and (b): the Zigzag path.