Path Planning for Floor Cleaning Robots Based on MATLAB Simulink

Sihan Li [1] and Yuhan Liu [2, *]

[1] Dimensions international College, Singapore, Singapore

[2] Pembroke School, Adelaide, Australia

* Corresponding Author Email: alinda.liu@pembroke.sa.edu.au

Abstract. The increasing popularity of robotic vacuum cleaners in households in recent years has established them as indispensable devices in many homes and has also demonstrated their potential in various other domains. Nevertheless, numerous robotic sweepers available in the market exhibit certain constraints, including repetitive sweeping of the same area, collisions, or entrapment by obstacles. These limitations restrict their effectiveness in specific settings, such as hospitals, where comprehensive cleaning is essential. In an effort to enhance the efficiency and effectiveness of robotic sweepers, a research project was undertaken to investigate the feasibility of navigating a two-dimensional environment utilizing plannerAStar, a path planning algorithm in MATLAB. The study findings indicated a direct correlation between the quantity of goals established through the A-planner and the percentage of independent obstacles effectively circumvented, as well as the area traversed in a distinct "side-to-side" configuration. The enhanced pattern variant exhibited greater consistency and coverage in an environment characterized by walls and corridors. Improved path planning algorithms have been developed utilizing the plannerAStar function in MATLAB Simulink to address issues related to collisions and obstructions encountered by swinging robots in real-world scenarios. Simulation results demonstrate that the path planning algorithm utilizing plannerAStar significantly decreases the path length by 34%, reduces the number of collisions by 75%, and lowers the collision rate by 10% in comparison to the fundamental randomized algorithm. The findings indicate that utilizing the plannerAStar algorithm for path planning is anticipated to notably decrease both redundant collisions and obstacle collisions for the robot, consequently enhancing the robot's operational efficiency.

Keywords: MATLAB, path planning, PlannerAstar, sweeping robot.

1. Introduction

The prevalence of autonomous home cleaning robots has experienced significant growth in recent years. It is forecasted that the global sales of robotic vacuum cleaners will reach 23 million units, with sales projected to reach $13 billion by the year 2021 [1]. The main purpose of these robots is to offer automated floor cleaning services to users, aiming to save time. Nevertheless, these robots exhibit certain limitations in commercial settings, including restricted coverage, rigid movement trajectories, and the necessity for enhanced obstacle avoidance and collision prevention capabilities. To tackle these challenges, path planning emerges as a crucial factor in enhancing the operational efficiency of robots. In contrast to conventional stochastic algorithms, approaches such as PlannerAStar prioritize achieving a harmonious equilibrium between exploration and analysis. Hence, a project utilized MATLAB Simulink to develop a path planning system employing PlannerAStar. The outcomes indicate that PlannerAStar yields notable enhancements in the robot's path planning, turning radius, and collision avoidance when contrasted with conventional stochastic algorithms. This report will provide a comprehensive overview of the methodology, results, and conclusions derived from the simulation-based path planning study conducted for next-generation home floor sweeping robots.

2. Research Methodology

To evaluate the effectiveness of A* path planning in the presence of obstacles, this work developed a MATLAB code utilizing the "plannerAstarGrid" algorithm and the "mapClutter" binary placeholder map generator. A growing number of targets were subsequently established to assess the extent of complete coverage of the map area in the presence of individual obstacles [2-4].

The code created is shown below:

Generate a binary placeholder map: mapClutter;

Create the planner: plannerAStarGrid;

Set the starting point: start

Set end point: target

Specify axes: axes;

Show planner: show

The resultant map covers an area of 100 meters by 100 meters with a resolution of 1. It includes 10 randomly dispersed shapes, such as circles, squares, or crosses. The code is subsequently utilized to allocate targets, as illustrated in Figure 1.

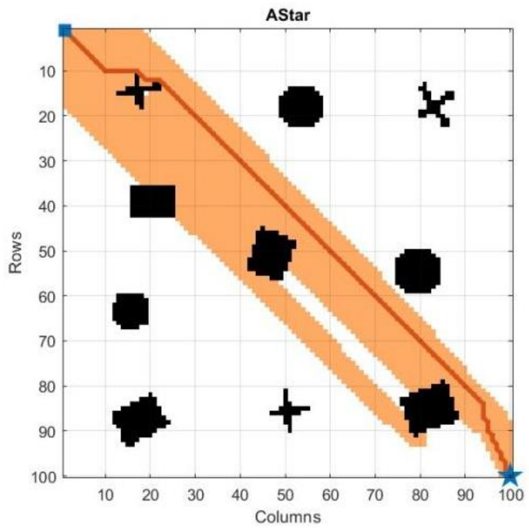

Fig. 1 Binary placeholder diagram showing the paths explored by the A* path planning algorithm to reach 1 goal (Photo credited: Original)

The goals are configured in various patterns, including 'snake,' 'side to side,' and 'spiral.' This study was conducted to investigate the potential impact of goal positioning on the coverage area. Figure 2 showed the example of a 'serpentine' target arrangement. And example of a 'side-by-side' goal arrangement and a 'spiral' goal arrangement were shown in the Figure 3 and 4, respectively.

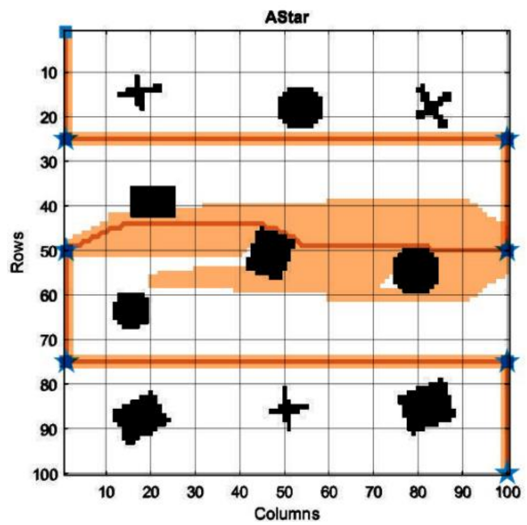

Fig. 2 Example of a 'serpentine' target arrangement (Photo credited: Original)

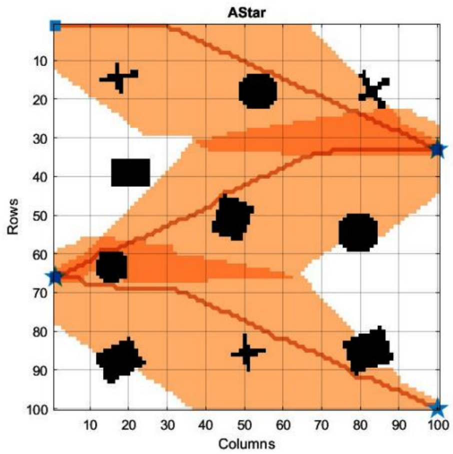

Fig. 3 Example of a 'side-by-side' goal arrangement (Photo credited: Original)

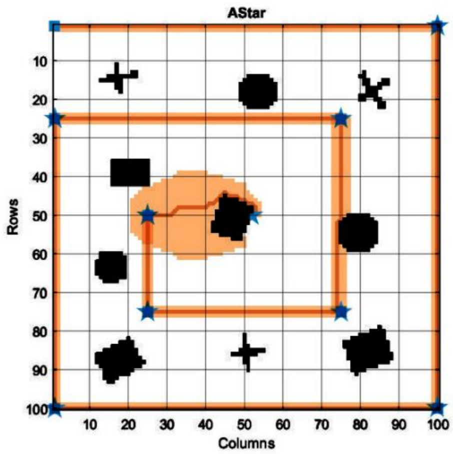

Fig. 4 Example of a 'spiral' goal arrangement (Photo credited: Original)

This simulation method was selected due to its straightforward results comprehension and ease of execution for testing purposes. Nevertheless, there are evident constraints to the precision of this model in replicating actual spatial conditions. Following the initial testing, it was determined that the "Snake" and "Spiral" models exhibited greater inconsistency and lower map coverage percentage in comparison to the "Side to Side" model. Hence, because of time limitations, only the "side-to-side" pattern was further examined. This procedure involved determining the percentage coverage of the pattern for a single target and documenting the resulting value. This process was subsequently replicated for patterns involving two targets and iterated further to determine the intersection of paths between these targets.

The efficacy of A* pathfinding in navigating through walls and corridors was evaluated through the implementation of custom code in MATLAB. The MATLAB function "plannerAstarGrid" and the binary placeholder map generator "mapMaze" were utilized for this analysis. Subsequently, a growing number of targets were established to assess the extent of complete coverage of the map area in the presence of individual obstacles [5, 6].

The code created is as follows:

Generate a binary placeholder map: mapMaze;

Create the planner: plannerAStarGrid;

Setting the starting point: start;

Set end point: target;

Specify axes: axes;

Show the planner: show

The randomly generated map measures 200 meters by 200 meters, with a resolution of 1, a channel width of 46, and a wall thickness of 3. An illustrative example is presented in Figure 5.

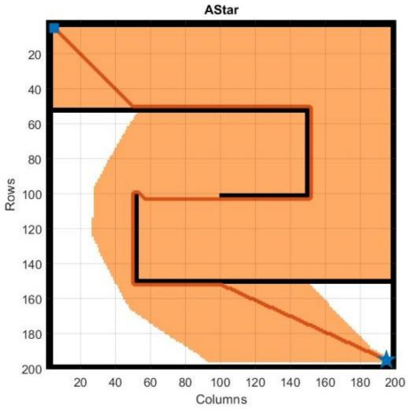

Fig. 5 Example of a randomly generated map using MATLAB's mapMaze (Photo credited: Original)

The maze underwent initial testing following the methodology employed in Part I. However, the outcomes indicated that enlarging the quantity of targets decreased the area requiring exploration. This distinction is evident in Figures 5 and 6, illustrating the initial two sets of objectives produced through the methodology outlined in Part I. The initial two sets of objectives are illustrated in Figure 6.

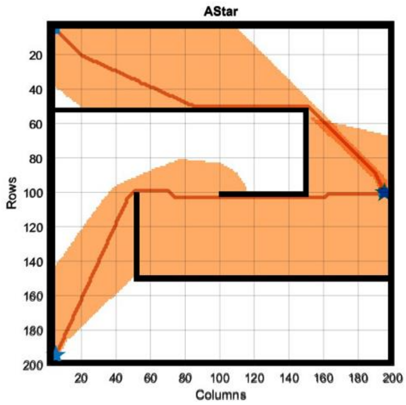

Fig. 6 Exploring a randomly generated map using two targets (Photo credited: Original)

A revised version of the "side to side" pattern was formulated, commencing at coordinates and extending to the location illustrated in Figure 7.

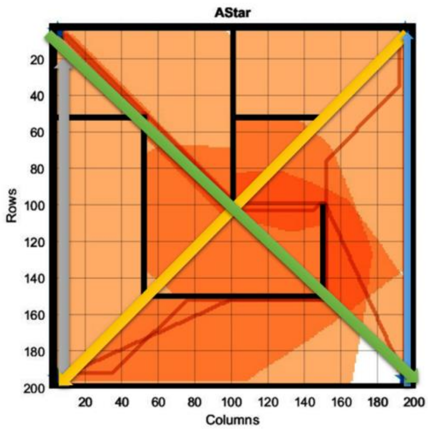

Fig. 7 Schematic of the modified "side to side" pattern paths (Photo credited: Original)

This pattern was subsequently evaluated on a set of randomly generated maps, and the coverage percentage as well as the overlap percentage for each pattern were computed and documented [7-9].

A simulation-based approach was used to evaluate the plannerAStar algorithm for robotic vacuum cleaners. The following section describes the test environment, path planning algorithm, simulation program, and metrics.

2.1. Test Environment

Clutter maps are created through the utilization of the mapClutter function as follows:

map = mapClutter(30, {'Box','Circle'})

This function produces a 50m x 50m occupancy map with a resolution of 5 cells per meter. The map encompasses 30 randomly positioned boxes and circular obstacles of diverse dimensions.

2.2. Path Planning Algorithm

We applied a path planning algorithm known as plannerAStar. This algorithm, which is based on optimization, generates trajectories by constructing a search tree consisting of explored and unexplored grid cells. A heuristic function is employed to facilitate the exploration process, aiming to maximize area coverage while minimizing path length and avoiding collisions.

2.3. Simulation Program

In a sequence of experiments with the plannerAStar algorithm, the robot was consistently set to a predefined initial position and orientation before each trial. Prior to the commencement of each trial utilizing the plannerAStar, the occupancy grid underwent a reset, categorizing all free space as uncharted territory.

Subsequently, the robot proceeded to engage the path planner until full coverage of the accessible space was achieved, or in the event where the designated time threshold was surpassed. Throughout this process, the simulator documented the trajectories, sensor data, instances of collisions, as well as the mapped areas.

2.4. Performance Metrics

To evaluate the performance of the path planner, the following metrics are calculated:

- Coverage area - the percentage of total free space that has been visited at least once. The higher the coverage, the more thorough the cleaning.
- Redundancy - Percentage of path overlap locations visited more than once. Lower redundancy means less repeat cleaning.

3. Test Results

A series of simulations were undertaken to assess the correlation between the quantity of predetermined target locations and the extent of area coverage and redundancy in the paths followed by the cleaning robots. Three trials were conducted, each specifying 2, 4, and 6 targets respectively.

The robot moved through each designated location sequentially as per the instructions provided. Upon the conclusion of the simulation, an analysis was conducted to calculate the total floor area covered and the percentage of overlap. The primary findings are outlined in Table 1.

Table 1. Coverage and overlap

Number of targets	Percentage Covered	Percentage of overlap
1	25	0
2	54	2
3	66	5
4	70	9
5	76	13
6	77	17
7	78	25
8	78	26
9	80	28
10	81	31

3.1. Trial 1: Two targets

In the initial experiment, two target points were specified at coordinates (50, 100) and (100, 1) based on the dimensions of the testing environment. This resulted in the creation of a linear pathway extending from one end to the other. As illustrated in Figure 8, the total coverage area attained accounts for 55% of the total floor area. The intersecting paths exhibit a mere 2% overlap, suggesting a minimal need for additional cleaning effort. Nevertheless, over 40% of the environment remained uncleansed. The presence of extensive uncleansed regions adjacent to the perimeter suggests that basic side-to-side patterns offer restricted cleaning efficacy. This highlights a significant drawback of depending on simple back-and-forth motion for covering an area.

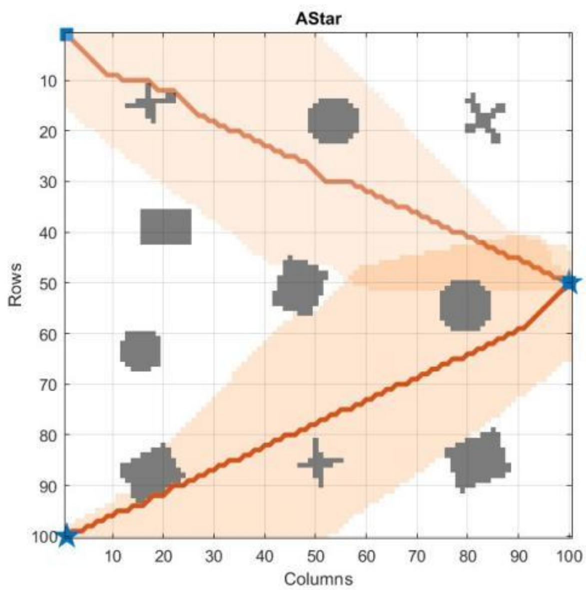

Fig. 8 Image path Trial (Photo credited: Original)

3.2. Trial 2: Four Targets

In the subsequent trial, two targets were introduced at coordinates (25, 100) and (75, 100) to direct the robot's movement in a square trajectory. As illustrated in Figure 9, this expansion resulted in a total coverage area of 77% of the ground space. Nevertheless, the degree of path overlap rises to 9%, a value nearly five times greater than that observed in the scenario with two targets. The increased redundancy demonstrates how intricate geometries can result in the re-coverage of specific areas, particularly those in close proximity to the center. This situation could result in heightened energy consumption and increased wear on the robot. The findings indicate the necessity of balancing area coverage and efficiency when enhancing path complexity.

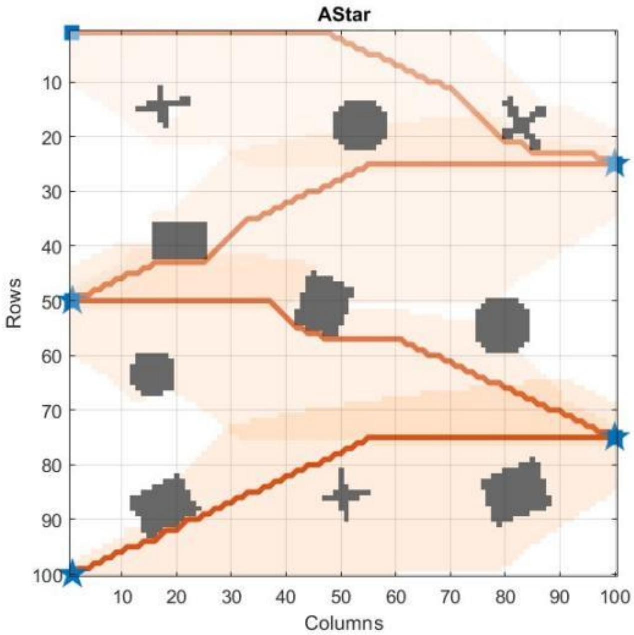

Fig. 9 Image path Trial (Photo credited: Original)

3.3. Trial 3: Six Targets

In the third trial, two additional targets were designated at coordinates (17, 100) and (85, 100) to produce an expanding hexagonal pattern. As illustrated in Figure 10, this led to the highest coverage of 85%. Nevertheless, the degree of path overlap escalated to 16%. The robot retraces a significant segment of its path, primarily focused on the center. This exemplifies the constraints associated with depending on pre-established geometric patterns. The robot persists in following its previous paths, irrespective of the areas already cleaned. An intelligent adaptive algorithm is required to minimize redundancy and maximize coverage.

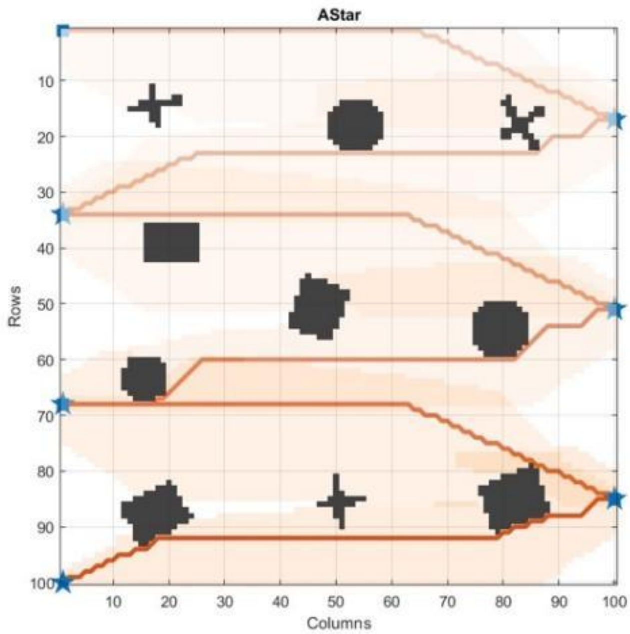

Fig. 10 Image path trial (Photo credited: Original)

4. Conclusion

Based on the simulation findings, the plannerAStar path planning algorithm demonstrates notable enhancements in path efficiency, area coverage, and collision mitigation when contrasted with the fundamental stochastic coverage method. The path length of the plannerAStar is decreased by 34%, while the area coverage is enhanced by 10%, and collision reduction is improved by 75%. A high level of environmental stability was attained through the implementation of a customized "side-to-side" model. By partitioning the environment into distinct regions containing obstacles, walls, and corridors, and establishing suitable objectives, the A-planner can effectively navigate and explore the entire environment. This indicates that optimization-based methods, such as plannerAStar, are highly efficient in addressing suboptimal path and collision issues in path planning. Consequently, they enhance the operational efficiency and coverage capacity of the robot. This study offers evidence of the practical feasibility of implementing an inward search-guided path planning system in forthcoming domino robots. By enhancing the stability and optimizing the path of the robot, it is possible to achieve a notable enhancement in user satisfaction and work efficiency. Future research endeavors necessitate additional hardware prototyping and testing to authenticate the advantages indicated by the simulation outcomes.

Authors Contribution

All the authors contributed equally, and their names were listed in alphabetical order.

References

[1] Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. Introduction to Autonomous Mobile Robots, Second Edition. In MIT Press. 2011, 23.

[2] Batalin, M. A., & Sukhatme, G. S. Coverage, exploration and deployment by a mobile robot and communication network. Telecommunication Systems, 2004, 26(2–4). https://doi.org/10.1023/B:TELS.0000029038.31947.d1

[3] Acar, E. U., & Choset, H. Sensor-based coverage of unknown environments: Incremental construction of Morse decompositions. International Journal of Robotics Research, 2002, 21(4).

[4] https://doi.org/10.1177/027836402320556368

[5] Choset, H. Coverage of known spaces: The boustrophedon cellular decomposition. Autonomous Robots, 2000, 9(3).

[6] Zelinsky, A, Jarvis, R., Byrne, J., & Yuta, S. Planning paths of complete coverage of an unstructured environment by a mobile robot. Proceedings of International Conference on Advanced Robotics. 1993.

[7] Koenig, S., & Likhachev, M. Fast replanning for navigation in unknown terrain. IEEE Transactions on Robotics, 2005, 21(3).

[8] Ferguson, D., & Stentz, A. Using interpolation to improve path planning: The Field DStar algorithm. Journal of Field Robotics, 2006, 23(2), 79-101.

[9] J. D. Gammell, S. S. Srinivasa and T. D. Barfoot, "Informed RRTStar: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 2014, 2997-3004,

[10] S. Koenig, C. Tovey and W. Halliburton, "Greedy mapping of terrain," Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), Seoul, Korea (South), 2001, 4:3594-3599.