



Optimizing Indoor Path Planning for Autonomous Mobile Robots with the A* Algorithm

Tsegaye Alemu Tola¹, Jing Mi², Yanqiu Che¹

¹School of Automation and Electrical Engineering, Tianjin University of Technology and Education, Tianjin, China

²School of Vocational Education, Tianjin University of Technology and Education, Tianjin, China

Email: tsegayealemu623@gmail.com

How to cite this paper: Tola, T.A., Mi, J. and Che, Y.Q. (2024) Optimizing Indoor Path Planning for Autonomous Mobile Robots with the A* Algorithm. *Open Access Library Journal*, 11: e12431.

<https://doi.org/10.4236/oalib.1112431>

Received: October 5, 2024

Accepted: November 16, 2024

Published: November 19, 2024

Copyright © 2024 by author(s) and Open Access Library Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Autonomous robotics is one of the key subjects of this generation of research and has grown to be more and more popular in recent years, especially in such industries, hospitals, hotels, and plenty more. Path planning is a critical issue in the applications of these autonomous mobile robots. This means the path planning task is to find a collision-free path for the robot, in an indoor environment that has cluttered obstacles, from the specified beginning or start point to the end or desired goal destination while satisfying certain optimization criteria. This paper explores the optimization of path planning of autonomous mobile robots in indoor environments using A star algorithm methods in different cluttered environments. The main contribution of this paper aimed to improve the path planning of autonomous mobile robots in the indoor environment on the basis of the A star algorithm and offer a unique capability field approach for time efficiency, moving the robots in the complex obstacle to getting the goal, and obtain better path for robots as well as for solving the problems that the traditional A* algorithm method often converges of much time, long distance, and some oscillatory movement of the robot. The simulation results show that the performance of the methods of path planning algorithm is getting good results which are applied in different environments and obtained better results on finding a short and safe path planning for the autonomous robot.

Subject Areas

Engineering

Keywords

Autonomous Mobile Robot, A* Algorithm, Path Planning, Optimizing

1. Introduction

In recent decades, robotics has become one of the most important areas of engineering research, with the use of robots enabling humans to perform critical, difficult, and dangerous processes with greater precision, reliability, and safety. Furthermore, with the development of current technology, robots are becoming not only cheaper but also more efficient, faster, more flexible, and smarter. In order to perform tasks, autonomous robots must be intelligent and must dictate their actions. The term path planning has evolved and developed in many fields, such as consisting of control theory, synthetic intelligence, and robotics. Path or route planning research of autonomous mobile robots has attracted attention in autonomous robots since the 1970s. In robotics, path planning concerns the problem of how to move a robot from one point/ starting point to another point/desired goal point. In robotics or artificial intelligence, path planning is a search for a sequence of logical actions that transform a starting point/initial robot state into a desired target/goal state. Such planning may include many decision-theoretic ideas such as Markov decision processes [1], imperfect state information, learning methods, or game-theoretic equilibrium [2]. In the control theory, path planning deals with issues of stability [3], feedback, and optimality [4] [5]. Generally, in robotics, path/route planning is focused on designing algorithms that generate beneficial motions by processing simple or extra-complex geometric models. Over the past several years, studies in this topic area have expanded because of the reason that mobile robots are now carried out in diverse applications.

Path planning algorithms are usually based on configuration space representations such as the potential Field method, voronoi diagram, visibility graph, regular grids [6], the cell decomposition method [7], generalized cones, quad-tree, and vertex graph, where the c-space is full of records systems that display the placement and orientations of gadgets and robots inside the workspace location together with the free space regions and forbidden areas with obstacles or mazes. In computing, statistics structures play a critical position and significantly have an impact on the computational complexity and efficient implementations of the algorithm [8]. For the time being, domestic and foreign scholars have put forward many algorithms after a larger number of research on path planning. Artificial potential field method [9], using sonar sensors [10], ant colony optimization algorithm, genetic algorithm [11], etc. are commonly used for path planning algorithms and also SLAM autonomous robots [12].

The path-planning robot uses algorithms, eliminating the need for complex electronic circuit boards, ground wires, ground straps, etc. The robot's planning has been miniaturized, minimizing the processing cost of robot path planning to meet general manufacturing needs. Functional requirements, the technical pursuit of simplicity and practicality, and the desire to allow users to recoup the application in the shortest possible time.

2. Applications of the Autonomous Mobile Robot Path Planning

High performance, high quality and time-saving of autonomous mobile robots initiate various human actions. The application of the robot path planning, mapping and localizing is shown in **Figure 1**.

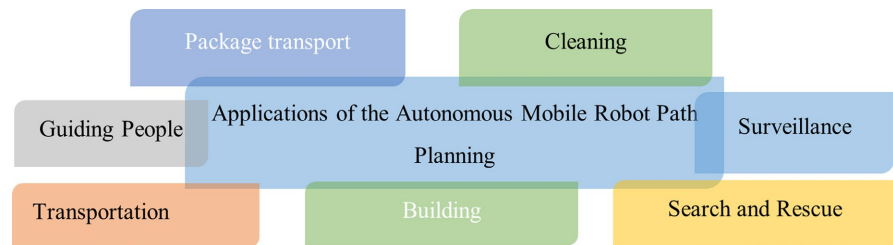


Figure 1. Applications of the autonomous mobile robot path planning.

- **Package transport:** This is used to send small packages of fast food, bills, files /documents, books, and other items using autonomous mobile robots. They are used for carrying medicinal drugs and documents in hospitals.
- **Cleaning:** Cleanliness and order are vital in any advanced civilization. However, cleaning and maintaining order have usually been heavy and monotonous activities. Robots can carry out those duties for us at domestic and in public locations.
- **Surveillance:** Algorithms that understand and version the environment are probably used to reveal human conduct.
- **Search and Rescue:** Search and rescue have several demanding situations mobility on land with debris, enough power for long missions, and figuring out victims.
- **Building:** The construction industry can gain from robotics as the manufacturing industry has done. Robots would liberate humans from heavy labor, improving their lifestyles greatly.
- **Transportation:** Most goods you use were brought from distant places, possibly by vehicles and human force.
- **Guiding People:** It could additionally be utilized in guiding blind people or for advertising in malls, offices, and concerts as well as tourism. Consider an income robot that is attractive to the public and engages humans to offer them a product.

3. Problems of Path Planning to be Solved

Autonomous robots could catapult the productivity and nice of various human activities. The path planning trouble of mobile robots is a warm spot in the discipline of mobile robotic navigation studies [13], mobile robots can find a premiere or close to-most effective path or route from the starting/beginning point to the target /goal point that avoids barriers primarily based on one or some overall performance signs (along with the lowest working cost, the shortest walking route,

the shortest walking time, and so on.) in the motion space. Path planning selects the shortest distance of the route; that is, the shortest length of the path from the starting point to the target point as a performance indicator. Some applications are package delivery, cleaning, hospitals, hotels, surveillance, search & rescue, building, and transportation. However, a primary challenge is that robot path planning in indoor environments to achieve these applications. The aim of this paper is to contribute to solving some problems of autonomous mobile robot path planning in indoor environments.

Autonomous path planning is the capacity of a robot to navigate in unstructured environments. An unstructured environment is an environment that is not adapted to facilitate robot path planning, for example, offices, factories or production industry, etc. The mobile robot path planning technique can be divided into two types consistent with the recognized degree of environmental records and environmental information *i.e.*, path planning based on global map information or local map information [14]. These two path planning techniques are referred to global path planning as well as local path planning. The global path planning approach can generate the path under a completely known environment (the position and shape of the obstacle are predetermined). With the global map version of the environment wherein the mobile robots are located, the search is performed on the established global map model. The most effective algorithm can obtain the optimal path. Consequently, global path planning includes two parts: the establishment of the environmental model and the path/route planning strategy.

Path planning calls for constructing an environmental map. Environmental map creation refers to the established order of a correct spatial location description of numerous items in the surroundings in which the robotic is placed, including obstacles, road signs, and so on: that is, the establishment of a spatial model or map. The purpose of constructing the environmental map is to assist the mobile robot for the best path from the start /beginning point to the target/goal point within the model of the mounted surrounding with obstacles. There are many mature methods for establishing an environment model for mobile robot path planning. After the environmental map is built, global route planning is finished. Algorithms of global path planning are, in particular, divided into two types: heuristic search methods and intelligent algorithms. The initial representation of the heuristic search is the A* algorithm developed by the Dijkstra algorithm. The A* algorithm uses a heuristic graph search algorithm for solving problems regarding the path planning of robots in indoor environments. Many scholars have improved the A* algorithm and obtained other heuristic search methods [15] [16]. Intelligent algorithms have lots of studies, including ant colony [17], particle swarm [18], genetic [19], bat, simulated annealing, and so forth [20].

Exclusive from the global path planning technique, the local path planning approach assumes that the position of the obstacle in the surroundings is unknown, and the mobile robot perceives its surrounding environment and its state only through the sensor. Due to the fact that global information of the environment

cannot be obtained, the local path planning focuses on the contemporary nearby surroundings information of the mobile robot and uses the local surroundings statistics received via the sensor to discover an optimal path from the starting point to the target point that does not touch the obstacle within the surroundings. The path planning strategy desires to be adjusted in actual time. Commonly used methods for local path planning include the rolling window [21], artificial potential field [22], and various intelligent algorithms [23]. The mobile robot path planning problem is typically formulated as follows: Given a mobile robot and a description of an environment, plan a path between two specified locations, a start point, and a destination point. The path should be freed from collision and satisfies positive optimization criteria (*i.e.*, shortest cost path). According to this definition, a path planning problem is labeled as optimization trouble.

3.1. Classifications of Robot Path Planning

Path planning for autonomous mobile robots is one of the most essential aspects of robotic navigation techniques. Robotic path planning can be classified into different kinds based on various situations. Path planning classifications are classified based on the algorithm, based on the environment, and based on completeness.

In this topic, it can be classifying the various issues that arise when designing the path of a mobile robot. As shown in **Figure 2**, it can be classified into three categories based on the robot's knowledge of the environment, the nature of the environment, and the method used to solve the problem.

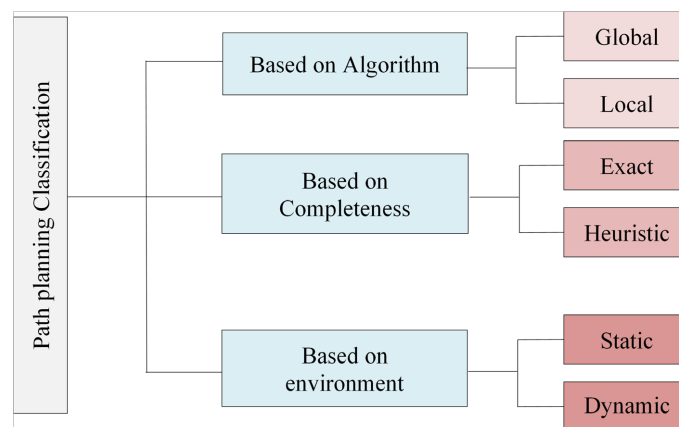


Figure 2. Path planning categories.

3.1.1. Path Planning Based on Algorithm

Path planning based on algorithm can be divided as global and local path planning. The global path planning commonly includes strategies that generate a hazard-free path, primarily based on a recognized environmental map or its contemporary and past perceptive statistics of the environment, to bring the robot to a predetermined destination. In the case of unforeseen obstacles blocking the pre-planned path, replanning the path based on the current environment would be

required. Hence, a certain extent of map building must be incorporated. The outcome is a slower reaction to an unexpected obstacle and computationally cost particularly whilst the instead of obstacles are dynamic.

The local path planning does not need a priori information about the environment. The mobile robot reacts to the detected obstacle and changes its heading direction in real-time to avoid the obstacle. These types of path-planning strategies are very appropriate for a dynamic environment. However, the local path planning technique is inefficient while the goal is far away and the environment is cluttered. The primary motivation in this topic is to develop a path-planning algorithm that has the advantages of the above global path-planning strategies.

3.1.2. Path Planning Algorithm Based on Completeness

The path planning algorithm based on completeness is categorized as either exact or heuristic. If an optimal solution exists, an exact algorithm finds it or proves that no feasible solution exists, while heuristic algorithms look for a good solution in a short amount of time.

3.1.3. Path Planning Algorithm Based on Environment

Based on the environment, it is possible to solve the path planning problem in both static and dynamic environments. In a static environment, the starting and destination point is fixed, and obstacles do not change their locations over time. However, in a dynamic environment, the place of obstacles and the point of the goal may change throughout the search, due to the uncertainty of the environment, so path planning in dynamic environments is typically more complex than in static environments. As a result, the algorithms must adapt to any unpredictable events, such as the implementation of new moving obstacles in the planned advanced path or when the target is moving continuously. When both obstacles and targets move, the path planning problem becomes even more difficult because it must react to both goal and obstacle movements in real-time. In this manner, we used the static environment of path planning. Path planning for mobile robots is based on using an existing map (map knowledge) as a reference to determine the initial and destination locations, as well as the link between them. The amount of information in the map plays an important role in the path planning algorithm's design.

4. Methods of Optimizing Path Planning

By overview the most commonly used path planning or route-making plans methods that can be carried out for autonomous mobile robotics in static indoor environments using the A-star algorithm. The system needs the following modules to function efficiently to create a robot that can intelligently make path planning without human involvement. Each module's function is as follows; the first module is image processing for the environment module, and the captured image which is taken cannot be used directly to find the shortest path planning through in the environment. This is since the image captured by the camera is not binary

and may contain noise, resulting in processing errors. As a result, the image must be processed before applying the A-star (A*) algorithm, the detail stated in the next topic environmental mapping model. The second module is the shortest pathfinder algorithm which is the A* search algorithm that is used to find the shortest path through the environment while avoiding various obstacles. The most reliable and efficient path will be calculated, and the resulting data will be sent to the autonomous robot.

4.1. Environment Mapping Model

When the conventional A* searching technique is applied for searching path planning, the environment is divided into two states: passable regions and forbidden regions by using the complex grid map modeling method and free space method [24] [25]. While our method is by using the vision sensor/camera to get the environment according to the demand needed of environment for gathering the environmental area and then processing the image for the system is an important aspect of the system. The design methods are shown in **Figure 3**, which shows the environmental map procedure for path planning.

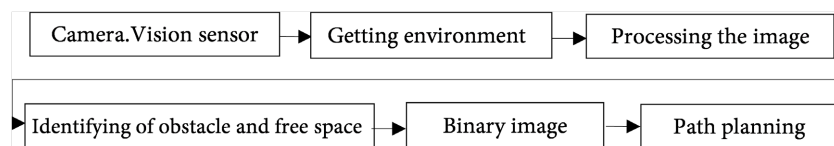


Figure 3. Environmental mapping procedure.

4.2. Traditional A* (A Star) Algorithm

The A* algorithm [26] is the searching algorithm effectively in terms of pathfinding optimization and thoroughness, it nonetheless has a bent to be a slower planner because of the quantity of space that is searched. It means that it will spend much of the walking time to complete the entire challenge if the range to be searched is big and concurrently, the time-ingesting problem will come about within the robot systems. A* algorithm is developed by Bertram Raphael and Peter Hart. of Stanford Research Institute, it solves the path planning hassle and answering attempting to find the nearest distance to journey from starting place to the destination. A* algorithm of rules may be summarized in the subsequent steps:

- Identify admissible robot movements
- Express a cost function:

$$f(n) = g(n) + h(n) \quad (1)$$

In Equation (1): where $f(n)$ represents the estimated path distance from beginning through node n to destination, $g(n)$ denotes the actual path length from beginning to n , and $h(n)$ is the function of the heuristic which represents the guess distance path length from n to destination.

- Calculate total distance $f(n)$ and move to the node with the lowest cost value.

- Again, repeat the same method until going to the destination point.

The A* algorithm is a heuristic searching system, its objective is to discover the nearest path in a graph from a starting node to a destination node [27].

4.3. Improved A* Algorithm

The A* algorithm is one of the most commonly used path planning algorithms because of its simplicity and speed. The algorithm sees the environmental map as a transform to a 2D occupancy grid. Each occupancy grid is assumed to be a node. From the robot's current position, all the cells surrounding it, *i.e.*, the neighbouring cells are checked for the presence of an obstacle and the cost. The optimal node will be chosen for the next step of the robot. In this way, the algorithm is run until the robot reaches the last node or the target. While creating the occupancy grid, the information regarding the cells' occupancy is provided. The knowledge of the start points and the goal point is needed which is given or provided by the user. Based on this data, the cells are checked to find the optimal next cell and are added to the list of free and optimal cells. A heuristic approach is used for the A* algorithm. Initially, the distance (cost) between the current node and the goal node is calculated. Then, it calculates all the neighbouring nodes which are free (White spaces) and which are not visited beforehand for the heuristic cost *i.e.*, the expected cost from all the neighbouring cells to the target. This heuristic cost is represented as $h(n)$. The cost needs to travel from the current node to all the adjacent cells is also calculated. This is called the path cost and is represented as $g(n)$. Once, the heuristic and the path cost are found, it calculates the entire cost from the current cell to the goal cell. This overall cost is represented by $f(n)$ and is given as Equation (1). For an optimized solution, the total cost, $f(n)$ should be used. Using just the heuristic or just the path cost will not give an optimal path. Thus, the A* algorithm produces a set of waypoints from the robot's current position to the target, which is then fed to the motion control model.

A* is a heuristic searching process, its objective is to find the shortest path in a graph from a node p_{init} to a node p_{goal} . It employs a function $f(n)$ that guides the selection of the next node that will be expanded. $f^*(p)$ is an estimate of $f(n)$ that is the cost of the shortest path that passes through a node p and achieves p_{goal} . These two functions are computed as shown in Equations (1) and Equations (2).

$$f^*(p) = g^*(p) + h^*(p) \quad (2)$$

where $g^*(p)$ is an estimate of $g(n)$, the cost of the shortest path from p_{init} to p , and $h^*(p)$ is an estimate of $h(n)$ the cost of the shortest path from p to p_{goal} . If $h^*(p) \leq h(n)$ for all p in the graph, the heuristic is admissible and the algorithm can be improved optimally. The implementation of the A* algorithm for path planning generally uses two lists named open and closed. The open list stores the node's data that are in the frontier of the search.

The closed list stores the node's data that have already been expanded. The

expansion of any point p is carried out using a matrix called the connectivity matrix, where a number 2 implies the current p position, the number 1 shows where a movement is allowed, and the number 0 indicates movement is not allowed for a mobile robot. Connectivity matrix (1): Only horizontal-vertical movement allowed, Connectivity matrix (2): Complete neighbor movement allowed Connectivity matrix (3) and Connectivity matrix (4): Larger movements allowed. The algorithm removes the node with the smallest f -value from the open list, expands it, inserts its successors who have not yet been expanded in the open list, and marks the node as expanded, inputting it in the closed list in each iteration. It repeats these steps until p_{goal} is removed from the open list or no more nodes are available in the open list. In the first case, A^* used the connectivity matrix constraints to find the shortest route between (p_{init}) and (p_{goal}) . If the second stop situation holds true, there is no way to solve the problem.

In this topic, the concept of the valuation feature of the A^* algorithm is used to enhance the calculation accuracy. The result obtained is a smooth path and time-efficient path planning by increasing the connectivity matrix of the direction of the robot and Euclidean distance is used for heuristic functions analysis. To find the two points which means the current point to the destination/goal point on the map, the length of a segment connecting the two points is measured. The Euclidean distance formula, as its name suggests, gives the distance between two points, such as the current point and goal points (or) the straight-line distance. Let us assume that (x_1, y_1) and (x_2, y_2) are two points in a two-dimensional plane. Here is the Euclidean distance formula.

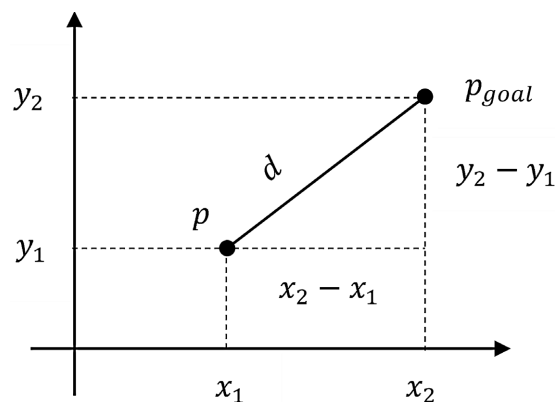


Figure 4. Euclidean distance derivation.

From **Figure 4**, (p, p_{goal}) means two points in Euclidean n -space; (x, y) means Euclidean vectors, starting from the origin of the space (initial point)

The Euclidean distance formula as Equations (3) is:

$$d^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \quad (3)$$

where (x_1, y_1) are the coordinates of one point which is used as p node point. (x_2, y_2) are the coordinates of the other point which is used as p_{goal} node point. d is the distance between adjacent cells p to the target p_{goal} .

Replacing the d^2 by the heuristic formula is below Equation (4).

$$h^*(p)^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \tag{4}$$

$$h(p) = \left[(x_2 - x_1)^2 + (y_2 - y_1)^2 \right]^{1/2} \tag{5}$$

So, this means $h(p) = \sqrt{(p_{goal_x} - p_{current_x})^2 + (p_{goal_y} - p_{current_y})^2}$.

In general, the cost function is calculated as $f^*(p) = g^*(p) + h^*(p)$

Where p is the adjacent cells of the robot point and $g^*(p)$ is the destination of the point of the robot. The movement of the robot is accomplished using a connectivity matrix shown below in **Figure 5**, in which quite a number 2 implies

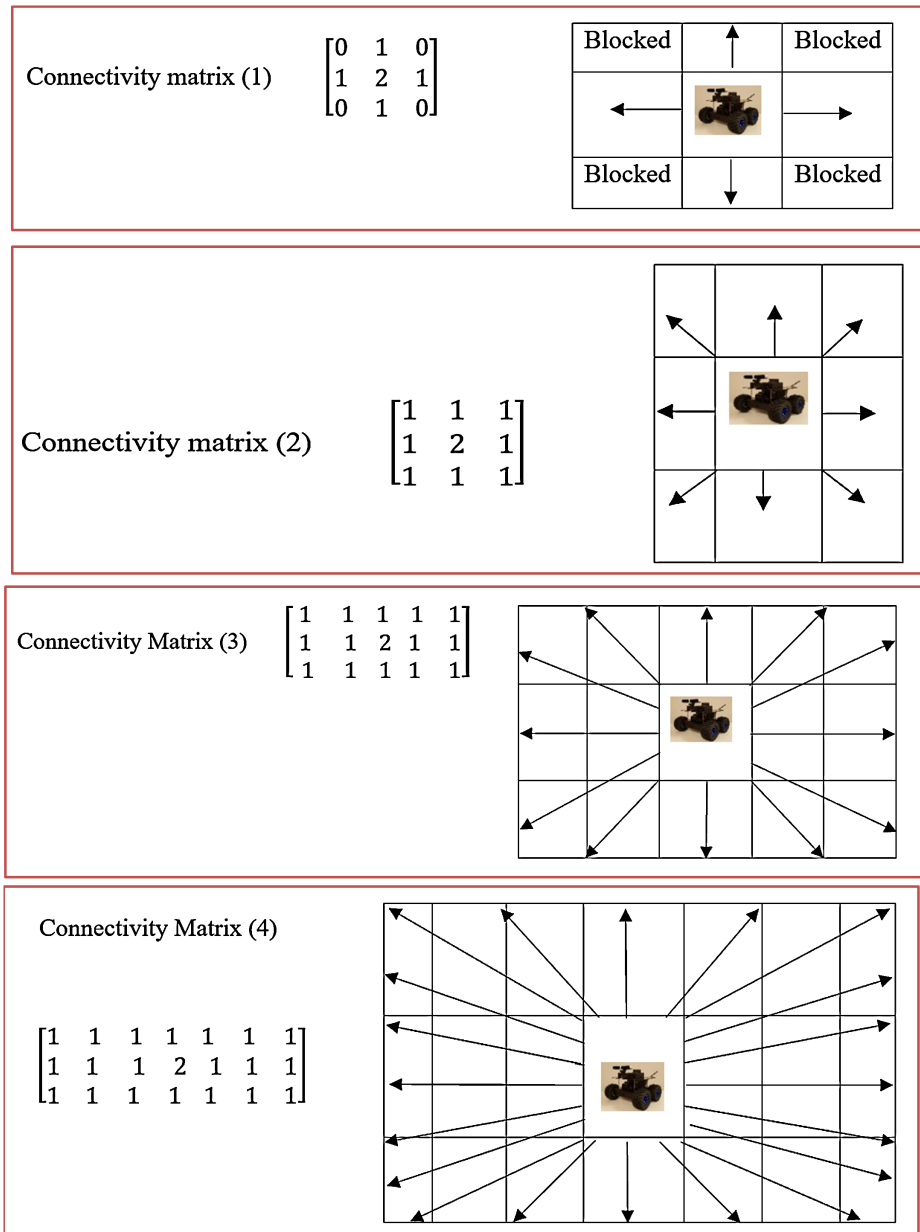


Figure 5. The connectivity matrix for the movement of robot.

robot position, 1 shows where a movement is possible for the robot, and a wide variety of 0 suggests movement blocked. Connectivity matrix (1): only forward-backward and left-right movement possibilities, Connectivity matrix (2): forward-backward, left-right movement possibilities and co-ordinate angles movement. Connectivity matrix (3): Many direction movements are allowed. Connectivity matrix (4): Extra many direction movements.

The detailed process of the Improved A star algorithm working method is shown in **Figure 6**.

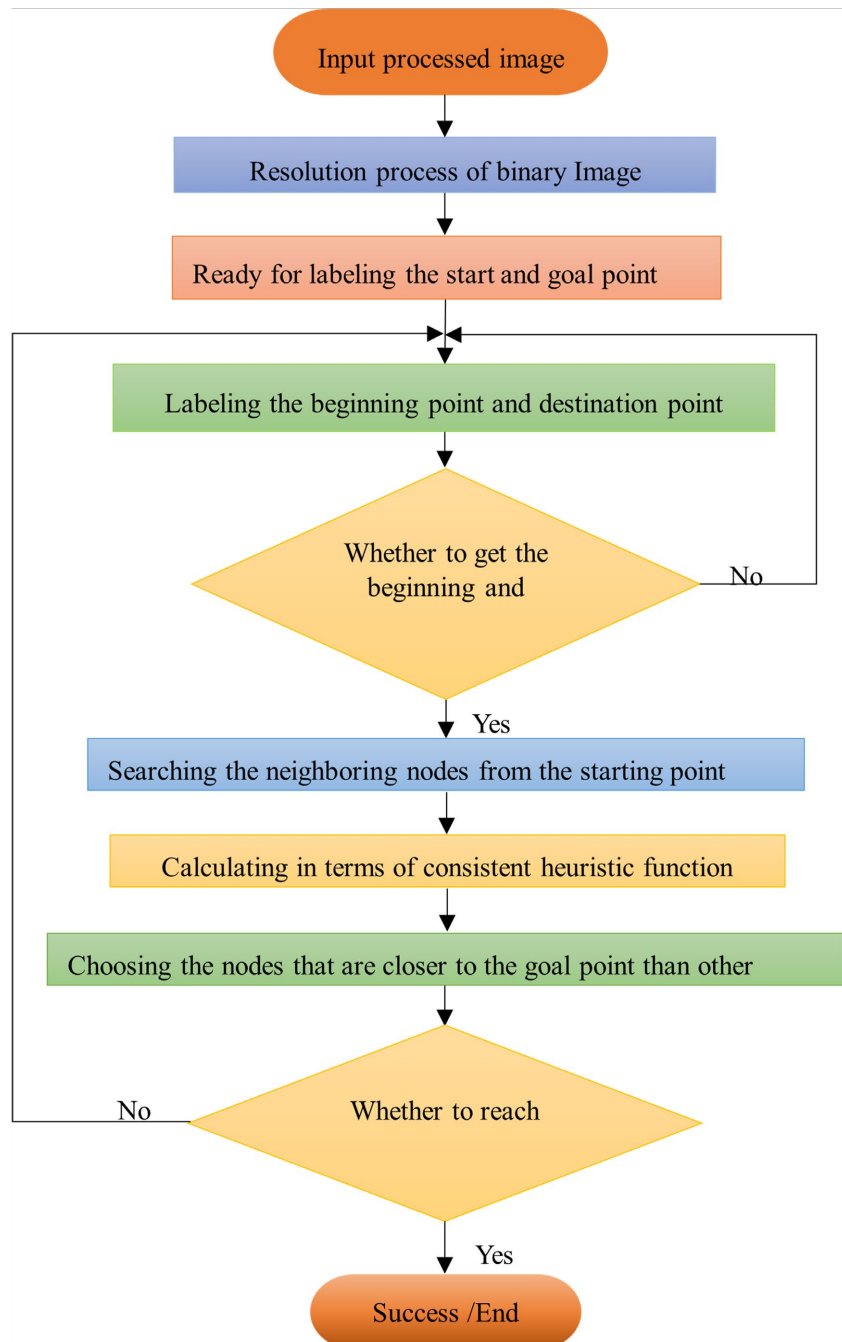


Figure 6. Improved A star algorithm working method.

5. Experimental Results of the Path Planning

In the experiment results, pathfinding for a robot (AMR) is done in indoor different working environment maps with several barriers at various positions. The comparison results of conventional and approved algorithms are below.

Simulation Result in Case 1: There are different obstacles in the working area, and there are different obstacles indoors with many rectangular obstacles like cartons and shelves. The approved algorithm is smoothed, high-speed, and time-efficient. The experiment result shown in **Figure 7(b)** is a good path result compared with **Figure 7(a)**. **Figure 7(b)** is for a robot moving with low oscillation during movement. Generally, the accomplishing time is much reduced and the path length is reduced. The simulation results are shown in **Figure 7**.

Simulation Result in Case 2: In an indoor environment irregular shape of obstacles. In **Figure 8(a)**, the traditional A* algorithm is more oscillated and moves a long way. In **Figure 8(b)**, the approved algorithm has not oscillated and the path is optimal.

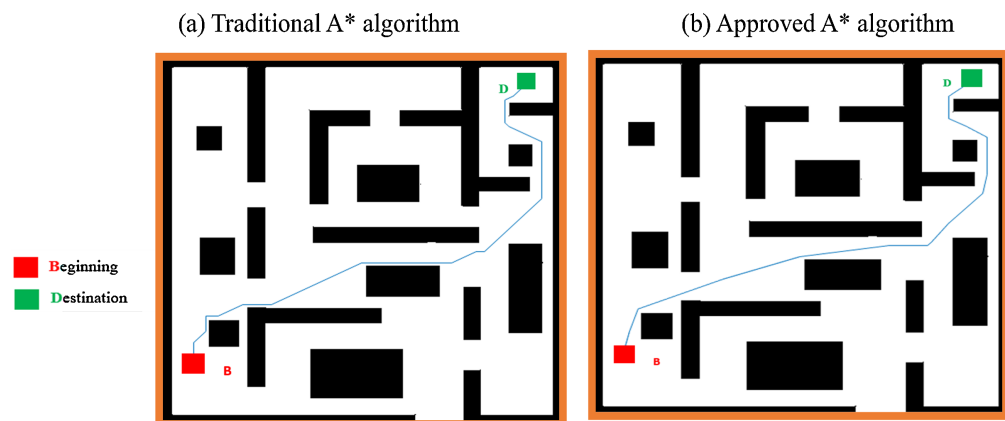


Figure 7. Simulation result in case 1.

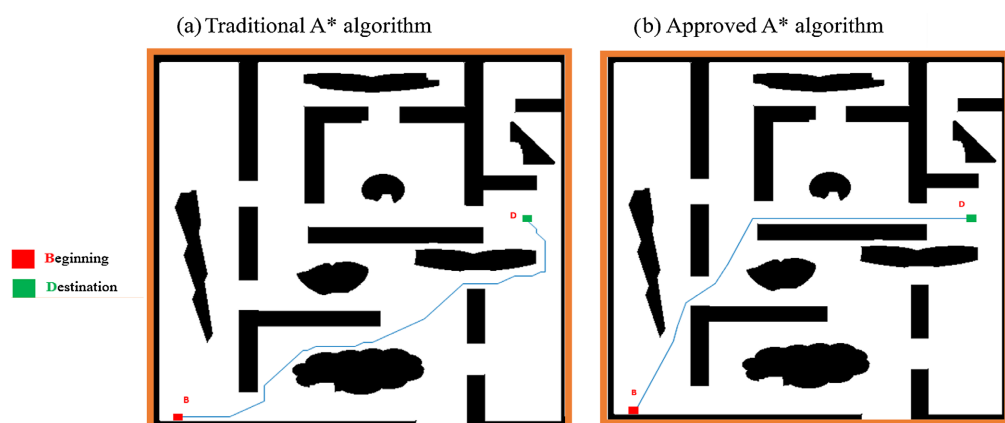


Figure 8. Simulation result in case 2.

6. Conclusion

In conclusion, this paper is Optimizing Indoor Path Planning for Autonomous

Mobile Robots with the A* Algorithm in a static environment, a method utilized for improving the performance time and increasing the efficiency of the moving in the short path. Then, by increasing the admissible connection matrix for the robot, modifying the heuristic, and implementing the environmental image transformation is to improve the traditional A* algorithm for the multiple problems of path oscillating in the find path and time-consuming (losing energy). The simulation results show that the performance of our methods of path planning algorithm is getting good result which is the simulation results are applied in different environments and obtained good results on such as finding a short safe path planning for the robot, it is time efficient path planning, made a path planning for the indoor environment without colliding with obstacles, and improved and optimized the performance of the path planning. Future research may utilize many methods of path-planning algorithms like improving Genetic, and Carrot planners, and others that can be used to modify the path-planning of autonomous mobile robots.

Conflicts of Interest

The authors declare no conflicts of interest.

References

- [1] Al-Sabban, W.H., Gonzalez, L.F. and Smith, R.N. (2013) Wind-Energy Based Path Planning for Unmanned Aerial Vehicles Using Markov Decision Processes. 2013 *IEEE International Conference on Robotics and Automation*, Karlsruhe, 6-10 May 2013, 784-789. <https://doi.org/10.1109/icra.2013.6630662>
- [2] Zhu, M., Otte, M., Chaudhari, P. and Frazzoli, E. (2014) Game Theoretic Controller Synthesis for Multi-Robot Motion Planning Part I: Trajectory Based Algorithms. 2014 *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 31 May-7 June 2014, 1646-1651. <https://doi.org/10.1109/icra.2014.6907072>
- [3] Chen, C., Shih, B., Shih, C. and Wang, L. (2012) RETRACTED: Enhancing Robust and Stability Control of a Humanoid Biped Robot: System Identification Approach. *Journal of Vibration and Control*, **19**, 1199-1207. <https://doi.org/10.1177/1077546312442947>
- [4] Ulusoy, A., Smith, S.L., Ding, X.C. and Belta, C. (2012) Robust Multi-Robot Optimal Path Planning with Temporal Logic Constraints. 2012 *IEEE International Conference on Robotics and Automation*, Saint Paul, 14-18 May 2012, 4693-4698. <https://doi.org/10.1109/icra.2012.6224792>
- [5] Gmitterko, A., Kelemen, M., Virgala, I., Surovec, R. and Vackova, M. (2011) Modeling of a Snake-Like Robot Rectilinear Motion and Requirements for Its Actuators. 2011 *15th IEEE International Conference on Intelligent Engineering Systems*, Poprad, 23-25 June 2011, 91-94. <https://doi.org/10.1109/ines.2011.5954726>
- [6] Miao, H. and Tian, Y. (2008) Robot Path Planning in Dynamic Environments Using a Simulated Annealing Based Approach. 2008 *10th International Conference on Control, Automation, Robotics and Vision*, Hanoi, 17-20 December 2008, 1253-1258. <https://doi.org/10.1109/icarcv.2008.4795701>
- [7] Ashleigh, S. and Silvia, F. (2010) A Cell Decomposition Approach to Cooperative Path Planning and Collision Avoidance via Disjunctive Programming. *49th IEEE*

- Conference on Decision and Control*, Atlanta, December 15-17 2010.
https://lisc.mae.cornell.edu/LISCpapers/CDC_AshleighDPCellDecomposition2010.pdf
- [8] LaValle, S.M. (2006) *Planning Algorithms*. Cambridge University Press.
<https://doi.org/10.1017/cbo9780511546877>
- [9] Yao, J., Lin, C., Xie, X., Wang, A.J. and Hung, C. (2010) Path Planning for Virtual Human Motion Using Improved A* Star Algorithm. 2010 *7th International Conference on Information Technology: New Generations*, Las Vegas, 12-14 April 2010, 1154-1158. <https://doi.org/10.1109/itng.2010.53>
- [10] Lee, J.-S., Kim, C. and Chung, W.K. (2010) Robust RBPF-SLAM Using Sonar Sensors in Non-Static Environments. 2010 *IEEE International Conference on Robotics and Automation*, Anchorage, 3-7 May 2010, 250-256.
<https://doi.org/10.1109/robot.2010.5509635>
- [11] Santiago, R.M.C., De Ocampo, A.L., Ubando, A.T., Bandala, A.A. and Dadios, E.P. (2017) Path Planning for Mobile Robots Using Genetic Algorithm and Probabilistic Roadmap. 2017 *IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, Manila, 1-3 December 2017, 1-5.
<https://doi.org/10.1109/hnicem.2017.8269498>
- [12] Tola, T., Mi, J. and Che, Y. (2024) Mapping and Localization of Autonomous Mobile Robots in Simulated Indoor Environments. *Frontiers*, **4**, 91-100.
<https://doi.org/10.11648/j.frontiers.20240403.13>
- [13] Guan, D., Yuan, W., Jehad Sarkar, A., Ma, T. and Lee, Y. (2011) Review of Sensor-Based Activity Recognition Systems. *IETE Technical Review*, **28**, 418-433.
<https://doi.org/10.4103/0256-4602.85975>
- [14] Porta, J.M. (2005) CuikSLAM: A Kinematics-Based Approach to Slam. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, 18-22 April 2005, 2425-2431. <https://doi.org/10.1109/robot.2005.1570476>
- [15] Borenstein, J., Everett, H.R. and Feng, L. (1996) *Where I Am? Sensors and Methods for Mobile Robot Positioning*. University of Michigan.
- [16] Lagarias, J.C., Reeds, J.A., Wright, M.H. and Wright, P.E. (1998) Convergence Properties of the Nelder—Mead Simplex Method in Low Dimensions. *SIAM Journal on Optimization*, **9**, 112-147. <https://doi.org/10.1137/s1052623496303470>
- [17] Gindele, T., Brechtel, S., Schroder, J. and Dillmann, R. (2009) Bayesian Occupancy Grid Filter for Dynamic Environments Using Prior Map Knowledge. 2009 *IEEE Intelligent Vehicles Symposium*, Xi'an, 3-5 June 2009, 669-676.
<https://doi.org/10.1109/ivs.2009.5164357>
- [18] Aurenhammer, F. (1991) Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys*, **23**, 345-405.
<https://doi.org/10.1145/116873.116880>
- [19] Khatib, O. (1985) Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Proceedings 1985 IEEE International Conference on Robotics and Automation*, St. Louis, 25-28 March 1985, 500-505. <https://doi.org/10.1109/robot.1985.1087247>
- [20] Feder, H.J.S. and Slotine, J.E. (1997) Real-Time Path Planning Using Harmonic Potentials in Dynamic Environments. *Proceedings of International Conference on Robotics and Automation*, Vol. 1, 874-881. <https://doi.org/10.1109/robot.1997.620144>
- [21] Cesarone, J. and Eman, K.F. (1991) Efficient Manipulator Collision Avoidance by Dynamic Programming. *Robotics and Computer-Integrated Manufacturing*, **8**, 35-

-
44. [https://doi.org/10.1016/0736-5845\(91\)90005-d](https://doi.org/10.1016/0736-5845(91)90005-d)
- [22] Azmi, M.Z. and Ito, T. (2020) Artificial Potential Field with Discrete Map Transformation for Feasible Indoor Path Planning. *Applied Sciences*, **10**, Article No. 8987. <https://doi.org/10.3390/app10248987>
- [23] Siciliano, B. and Khatib, O. (2008) Handbook of Robotics. Springer.
- [24] Hertzberg, C., Wagner, R., Birbach, O., Hammer, T. and Frese, U. (2011) Experiences in Building a Visual SLAM System from Open Source Components. 2011 *IEEE International Conference on Robotics and Automation*, Shanghai, 9-13 May 2011, 2644-2651. <https://doi.org/10.1109/icra.2011.5980140>
- [25] Nakhaeinia, D., Tang, S.H., Noor, S.M. and Motlagh, O. (2011) A Review of Control Architectures for Autonomous Navigation of Mobile Robots. *International Journal of Physical Sciences*, **6**, 169-174. <http://psasir.upm.edu.my/id/eprint/22917>.
- [26] Ziliaskopoulos, A.K. and Mahmassani, H.S. (1993) Time Dependent, Shortest-Path Algorithm for Real-Time Intelligent Vehicle Highway System Applications. Transportation Research Record 94-100.
- [27] Patrick Lester. (2005) A* Pathfinding for Beginners. WebSite. <https://csis.pace.edu/~benjamin/teaching/cs627/webfiles/Astar.pdf>