Journal of
Engineering Sciences
Faculty of Engineering
Assiut University
JES
مجلة
العلوم الهندسية
كلية الهندسة
جامعة أسيوط
journal homepage: http://jesaun.journals.ekb.eg

Development of Path Planning Approach for Mobile Robot in Static Environment

Received 8 December 2024; Revised 7 January 2025; Accepted 7 January 2025

Khalil Ibrahim[1]
Ahmed Saad[2]
Aly S. Abo El-Lail[3]
Issa Zidane[3]

Keywords
Path planning; Image processing; A_star algorithm;
Wave front method.
Matlab software

Abstract: Self-motion Robots and their static environments are the main topics for trajectory planning researchers, especially when the environments contain static and movable obstacles that make the robots' mission harder to reach their target. Therefore, accurate and comprehensive knowledge of autonomous robots' control strategies makes us take the basic steps to build our approach in mobile robot trajectory planning. Accordingly, in this research paper, we will provide an accurate detail of two path planning algorithms A star and 4_sector wave front algorithms. Some modifications and improvements to its software structure for each algorithm that will reduce the time it takes for the robot to reach the desired goal, taking into account the presence of obstacles in the map that is dealt with and how to avoid collision with it, especially dead-end obstacles, which cause the robot to enter into a state of making wrong decisions that can lead to a collision or a significant waste of time required to reach the specified goal. The simulation results of the A-star algorithm and 4_sector wave front methods using MATLAB program are achieved in different ways using inputs via camera, then image processing is applied to determine the locations of obstacles, starting point, free space, and target point for the robot without any position error. The wave front algorithm takes a shorter time than the A-star method to reach the final position.

1. Introduction

Most industrial fields these days have become dependent on automobile robots or robots that make their own decisions notably that robots have become very effective and valuable in risky work environments or that can cause harm to humans if they are present in them, such as radioactive environments or containing Chemical materials that can lead to cancers. Moreover, the availability of these robots in production lines increases production capacity

[1] Dept. of Mechatronics Eng., Faculty of Engineering, Assiut University 71515, Assiut, Egypt / Faculty of Industry and Energy Technology, New Assiut Technological University (NATU), New Assiut city, Assiut University khalil.ibrahim@aun.edu.eg
[2] Dept. of Mechatronics Eng., Faculty of Engineering, Assiut University 71515, Assiut, Egypt. a.saad1@aun.edu.eg
[3] Dept. of Mechatronics Eng., Faculty of Engineering, Assiut University 71515, Assiut, Egypt, ali.abouelail@eng.au.edu.eg
[4] Dept. of Mechatronics Eng., Faculty of Engineering, Assiut University 71515, Assiut, Egypt, Issa-zidane@hotmail.com

https://doi.org/10.21608/jesaun.2025.342727.1388

and gives a product with manufacturing accuracy greater than the human element, in addition to economic feasibility as it reduces the cost of labor [1].

Decision-making in mobile robots mainly depends on the software algorithm used to guide the robot on the map it is moving on and identify ways to avoid the obstacles it encounters. Therefore, researchers have found many methodologies that will give robots the ability to move independently to reach their goal. These methodologies are separated into heuristic [2], deterministic and randomized algorithms. Algorithms that use the Deterministic approach are commonly preferred when the representative has limited degrees of freedom or the size of the map is low, that makes the quality of the paths less smart. The environment plays an important role in route planning issues. Depending on the nature of the job site, route planning can be categorized as offline and online. The robot's motion planning depends on finding a chain of activities that transform the primary state into the desired target state. We can decide whether the robot path is the best or not by the cost, so the lower the cost, the better the path. Also, the movement strategy becomes more accurate if it produces the optimal path which means the shortest path with the least time without hitting obstacles when the path already exists, or the algorithm make a note telling us that there is no path available [3].

Many years ago, trajectory planning based on the map used various methods and methodologies, we can divide them into: classical approaches such as artificial potential field (APF), roadmap approach (RA), cell decomposition (CD); and reactive approaches such as fuzzy logic (FL), cuckoo search (CS), shuffled frog leaping algorithm (SFLA), firefly algorithm (FA), artificial bee colony (ABC), neural network (NN), bacterial foraging optimization (BFO), genetic algorithm (GA), ant colony optimization (ACO), other miscellaneous algorithms (OMA) and A Star algorithm (A*) [4,5]. But among the above algorithms, A_star is the simplest and fastest algorithm and will be adopted in this paper. The A* (A_star) algorithm is a high level of Edger Dijkstra's algorithm in [6]. The wave front transmission algorithm has been performed as the controlling technique for path planning in separate grid maps. By using heuristics, the A* Algorithm produces an improving implementation of the shortest robot's trajectory to reach its goal. Specifics of absolute heuristic search and uniform-cost search are combined in A* to calculate the perfect solutions for path planning. The distance traveled between the starting point of the robot and the target location is calculated by the cost-function F(x) which is the basis of the A* algorithm [7]. The environment surrounding the robot constitutes the income data that will be dealt with, such as the locations of obstacles and empty spaces the robot can be moved through, the current location of the robot, and the location of the target point [8]. That's why vision sensors play a major role as a data source [9]. So, a huge amount of information about the working space can be obtained by using vision sensors like cameras, compared with touch sensors, ultrasonic sensors, or IR sensors, which provide a limited amount of data like colors or distances for the working environment [10]. The cameras are the primary part of the vision sensors, which capture an image of the environment using either a horizontal camera that monitors the environment from a high place or an overhead camera mounted to the front of the robot [11].

This paper depends on input images that come from a camera that captures pictures horizontally, covering the entire working environment, these images send wirelessly to a computer device that uses MATLAB software which will use the pictures as an input to the image processing toolbox to produce a clear map with specific places for the robot' current location, obstacles, free spaces, and target point. After that, an accurate simulation using the wave front algorithm will be utilized to find the robot's optimal path.

The path produced from the A* approach should be optimal, that is mean minimal energy consumption, proceeds less time, and limits the effect of a collision between the robot and the obstacle. The great defiance for optimal path planning of Robots is as follows:

Path length: The path length is the total distance traveled by the robot from the origin point to the target location.

Optimality: means that the algorithm must be efficient for time, cost, and energy. Therefore, it can be classified within three criteria: optimal, suboptimal, and non-optimal.

Completeness: It is the criterion used to determine if the path produced by the system is optimal or not. It can also provide appropriate solutions to reach the optimal path if the algorithm produced many paths.

Time-efficiency: it means that the time required to reach the target location must be minimum. That could be correct if and only if the robot’s path was the shortest path for completing the required operation.

Robustness: it means that the robot can deal with external or internal errors like position sensitive device errors, rotation driving errors, linear driving errors during motion planning. Collision avoidance: Collision avoidance is defined as the mechanism in which robot has the capability to detect the obstacles and avoid collisions without physical damages to robot’s body. These parameters are considered for our robot path planning algorithm [12]. The drawbacks of each conventional path planning technique including several points like Computational Complexity (Graph-based methods) - Scalability Issues - Memory Consumption - Static Environment Assumption

In this paper, the wave front method and A-star methods are presented. The proposed methods will improvements for the outcome of the optimal path which helps the robot to move from its position to the target position without collision with any obstacle and also consider all possible considerations such as the surrounding environment, dead end, U_shape, the building of the algorithm and the necessary time for finding the optimal route. The consideration of choosing these algorithms according to some reasons:

- The proposed algorithms could work with a small number of sensors (one camera), so the hardware costs are small.
- The wave front technique has easy code, which creates it faster than the earlier algorithms and does not need a large memory.
- The A-star method provides us with the greatest probable optimal route.

This paper will present several strategies for robot's trajectory planning based on path planning method that contains finding the optimal path for the robot to reach its destination safely with respect to the previous parameters. Series of operations will apply on the environment's map by using image processing techniques to remove noise and unwanted information, and extract the main features from the images, to introduce a map with accurate locations for the robot, obstacles, the target and start point. After image processing, path planning methods will apply based on the inputs come from image processing by using MATLAB software, then we will show the time spent on creating the optimal path and how we reduce it. This paper is organized as follows: Section 2 presents an explanation of image processing and its main role to get a clear image which captured by a horizontal camera, and the methodology of path planning methods explained in section 3. Section 4 shows results and analysis of the simulation work. Section 5 presents the conclusion. Finally, future work and conclusion in section 6.

2. Image Acquisition and Processing

The digital model of a location or map is called Image acquisition, so the resulting representation is the image or the picture, and its components are known as pixels (picture elements). Also imaging sensors are electronic instruments used to capture the environment [13]. The details of the image represent the main input elements for the MATLAB Software and its image processing toolbox. The image processing outputs could be letters or numbers, describing the basic picture of the environment. These days, the robotics field's applications depend mainly on vision sensors more than other sensors, that because vision sensors provide us with huge information about the environment like distances, colors and obstacles' positions, compared with the other traditional sensors. In addition to their decreasing cost, which makes them ideal sensors for robots [14, 15]. A horizontal image for the map will obtain by a Camera (Specifications: Digital Camera FHD 1080P with 32GB Card, 2 Batteries, Lanyard, 16X Zoom Anti Shake 44MP Compact Portable Small Point Shoot Camera), then this image will send to MATLAB software by wireless connection. In this stage, the image processing toolbox starts its work by filtering the image and removing the noise to produce a digital picture with clear positions of the robot, target and the obstacles [16]. The methodology of image processing is illustrated in the flow chart as shown in Fig. 1. For example, the image acquisition process as follow:

First step, the image of the environment is captured by using a horizontal camera.

Second step, this captured image is sent to the MATLAB software by using a wireless connection.

Third step, in the image processing toolbox, perform filtering operations like "open and close image" to eliminate salt and pepper noise [17] as shown in Fig.2. Salt-and-pepper noise is a form of noise sometimes seen on images also known as impulse noise, this noise can result from sharp and unexpected disturbances in the signal of the image. It shows as sparsely occurring white and black pixels as Fig. 2 illustrated, also, use the "edge detection" model based on Sobel operator.

The fourth step, the image converts into binary form, zeros '0' represent obstacles and ones '1' as surrounding space.

Fifth step, the black and white pixels will be reversed, then detecting obstacles in the map will have the black color.

Sixth step, dividing the picture into sectors, each one involves a group of pixels so that values of pixels of the image will distribute on sectors.

The last step, the final image is ready for applying the proposed algorithm on it.

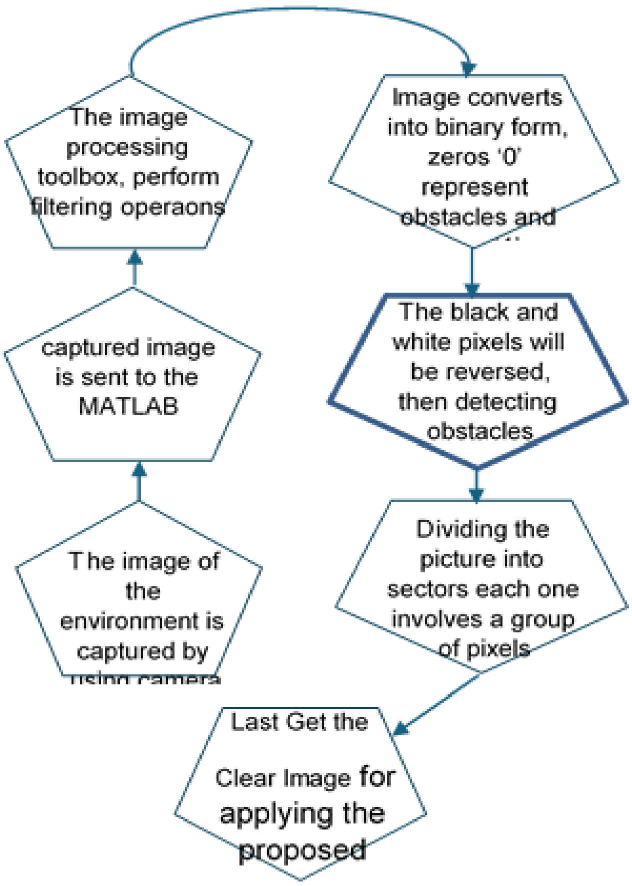


Fig. 1. Flowchart of image processing

2.1. An Edge Detection Model Based on Sobel Operator

One of the various kinds of the orthogonal gradient operators is Sobel operator. The first derivative corresponds to the Gradient, and the derivative operator is the gradient operator. The gradient of the position (x, y) could be represented as a vector, and its components are the two first derivatives along the X and Y direction respective [18]:

$$\nabla f = [G_x G_y]^T = \left[\frac{\partial f}{\partial x} \frac{\partial f}{\partial y}\right] \tag{1}$$

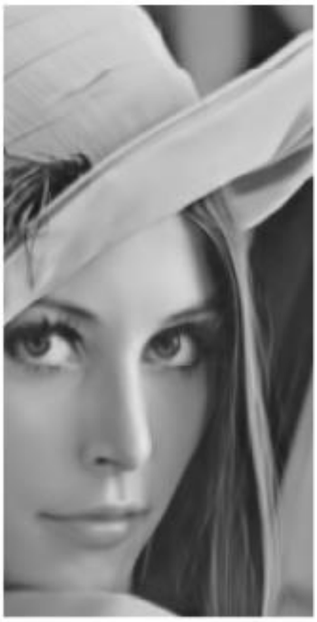
The equation (2) represents the magnitude, and the equation (3) represents the direction angle of the vector:

$$mag(\nabla f) = \left|\nabla f_{(2)}\right| = \left[G_x^2 G_y^2\right]^{1/2} \quad (2)$$

$$\emptyset(x, y) = arctan\left({G_x}/{G_y}\right) \quad (3)$$

Approximation often applied on small area template convolution, each one of G_x and G_y require a template, so these two templates joint in an operator of gradient. In Fig. 3 (a) and (b) are the two used templates 3*3 by Sobel. The two 3x3 templates used by Sobel are showed in Fig. 3 (a) and (b) are the two used templates 3*3 by Sobel. These two kernels should be used by each point to apply convolution. The maximum response to the vertical edge and the level edge is obtained by using these two kernels, so the output bit represents the maximum value of the two convolutions, which is the edge image amplitude

a) An image contaminated salt and pepper noise.



b) An image without salt and pepper noise.

Fig. 2. The methodology of image processing

.

-1	-2	-1
0	0	0
1	2	1

(a) Convolution template S1

-1	0	-1
-2	0	2
-1	0	1

(b) Convolution template S2

Fig. 3. Sobel edge operator

Their convolution is as follows:

$$G_1(x,y) = \sum_{k=-1}^{1}\sum_{l=-1}^{1} s_1(k,l)f(x+k,y+l) \tag{4}$$

$$G_2(x,y) = \sum_{k=-1}^{1}\sum_{l=-1}^{1} s_2(k,l)f(x+k,y+l) \tag{5}$$

$$G(x,y) = G_1^2(x,y) + G_2^2(x,y) \tag{6}$$

$if\ G_1(x,y) > G_2(x,y)$ It means that there is an edge.

Example: Let’s take an image for a virtual environment and apply the previous steps as Fig.4 shown.

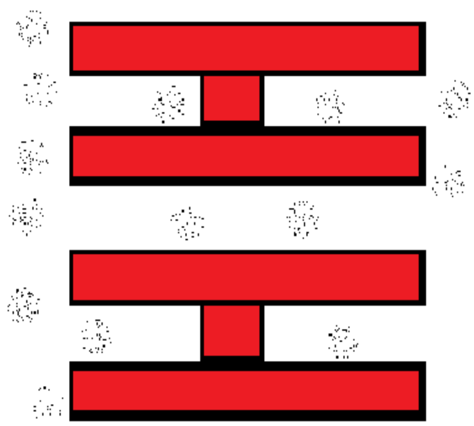


Fig.4 virtual environment with salt and pepper noise

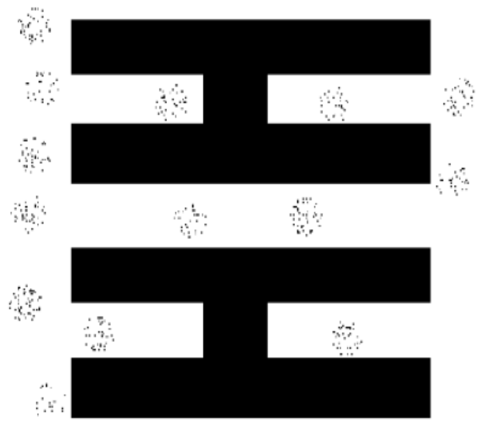


Fig. 5. Binary image with salt and pepper noise.

Now by using the image processing toolbox we can convert the image into binary form as Fig.5. We have to filter the image out of salt and pepper noise because this noise will lead to loss of the algorithm where the algorithm will consider each noise zone as a barrier and thus will make the road longer and will take a longer time to process the map. After using open image, close image, and edge detection techniques we get a clean image without distortion or noise as Fig. 6 shown.

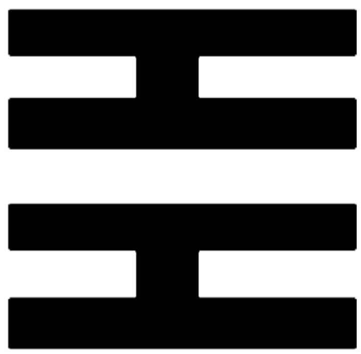


Fig. 6. An image without salt and pepper noise by using filters and edge detection techniques

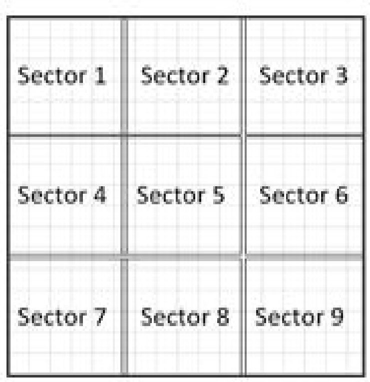


Fig. 7. Segmentation the image into sectors

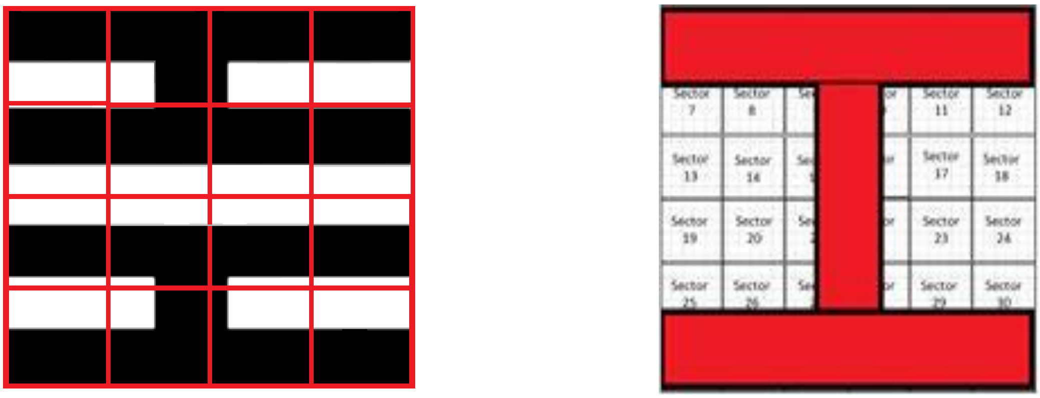


Fig. 8. An image after segmenting into sectors

The last step is to segment the image into sectors to be ready for applying the path planning algorithm as Fig. 7 illustrated. This segmentation is done by assembling the pixels in separate groups and each sector will be given an intermediate value for the pixels which contained in as illustrated in Fig.8

3. Methodology of Path Planning Methods

This section covers the path planning method for a mobile robot in different environments. Several methods of path planning techniques allow a mobile robot to navigate through obstacles and find the path to reach the target without collision.

3.1. A_Star (A*) Algorithm

A-star is a search algorithm that deals with specific inputs from the map of the environment and then makes a lot of calculation processes upon possible paths to produce the perfect track. It's one of the various computer science algorithms specialized in trajectory planning and graph traversal. The approach of A* applies a best-first search and produces a trajectory with the least cost from a beginning sector to the wanted one [19]. This algorithm performs a full map scan and generates a sequential and ordered list of sectors which have the lowest traffic cost, also makes a list of the previously scanned sectors so as not to be scanned again, which wastes time. These considerations make A_star one of the best search algorithms, if not the best.

3.1.1. The Elements of A_Star

The A_Star algorithm depends on the sum of two main functions, g(x) and h(x). The first one expresses the distance traveled from the starting point to the current location, and the second function represents the distance required to reach the target from the current location, where "X" is a random sector on the map of the environment. These components produce the following equation:

$F(x) = g(x) + h(x)$ as illustrated in Fig. 9.

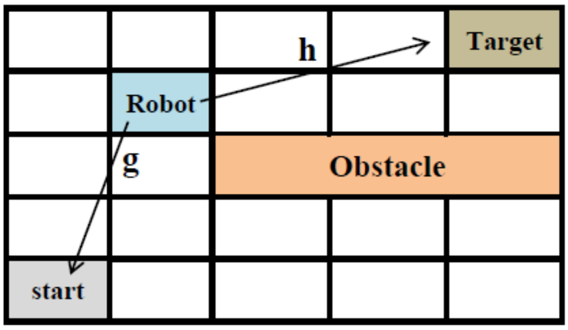


Fig. 9. The elements of A-star algorithm

3.1.2. The principal work of A_star algorithm

This algorithm provides the robot with an auxiliary search methodology, which enables it to explore available paths and avoid paths that lead to dead ends. That is done by creating two lists called the open list and the closed list. These two lists represent the basic properties of building the A_star algorithm [20], [21]. The scanned sectors are recorded and saved in the closed list, while in the open list, sectors adjacent to the scanned sectors are recorded, and then the distance of these sectors from the origin point and the target point is calculated. The addresses of sectors with the most appropriate and shortest distances are sequentially grouped, which will form the desired path [22], [23].

3.1.3. The Stages of A_Star Algorithm

First step: initialize the start sector 'n' and determine its position with X&Y axes, then put it on the open list and calculate G & H by using the two Eq (7,8):

$$G = \sqrt{(x_c - x_s)^2 + (y_c - y_s)^2} \tag{7}$$

$$H = \sqrt{(x_c - x_g)^2 + (y_c - y_g)^2} \tag{8}$$

Where:

$(\mathbf{x_c}, \mathbf{y_c})$ are the coordinates of the current sector. $(\mathbf{x_s}, \mathbf{y_s})$ are the coordinates of the start sector. $(\mathbf{x_g}, \mathbf{y_g})$ are the coordinates of the goal sector.

Second step: calculate the cost function F by using the next formula

$$F(n) = H(n) + G(n) \tag{9}$$

Third step: remove from the open list the sector ‘n’ which has the smallest cost function F and put it on the closed list and save the index of this sector.

Fourth step: check if ‘n’ is the target sector then terminate the algorithm and use the pointers of indexes to get the optimal path, if 'n' is not the target sector, then detect all the successor sectors of 'n' which do not exist on the closed list.

Fifth step: determine G & H for each sector, then go to step two. Figure 10 represents a flow chart for A_star path planning algorithm for mobile robot.

3.1.4. The MATLAB Functions of A_Star Algorithm

The file A-star.m represents the algorithm with all sub-functions, these sub-functions are:

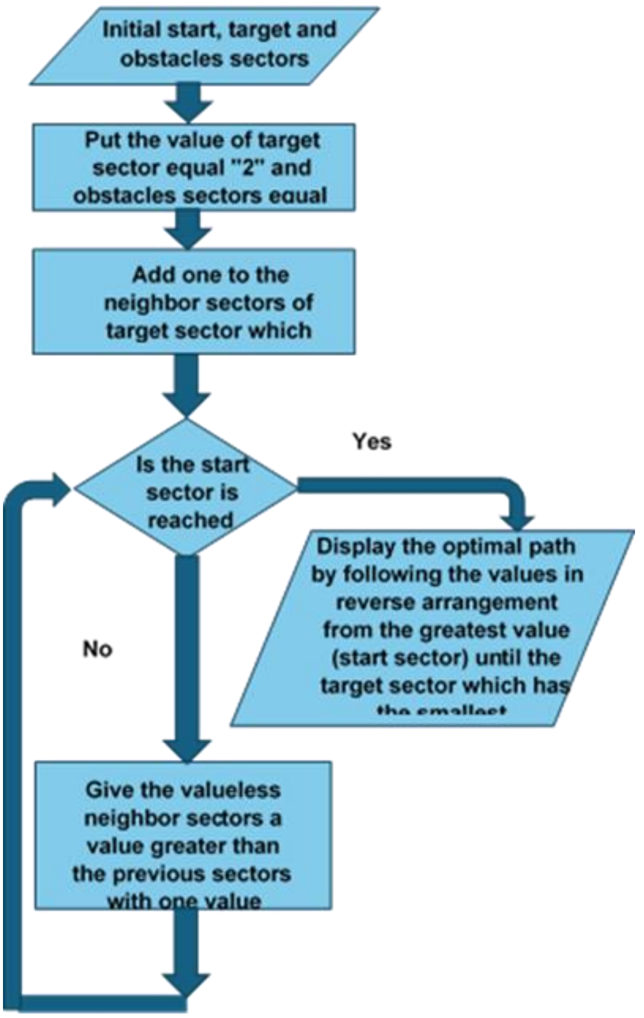


Fig. 11. The flow chart of wave front method

- The 'distance' function: its mission is to calculate the distance associated with the sectors.
- The ‘Expan_array’ function, which calculates the cost function for the current sector and its successors, returns that in an extended list.
- ‘Insert_open’ this function fills the list OPEN with values that have been passed to it.
- ‘min_fn’ this function checks the list OPEN and returns the address of the sector which has the smallest cost function.
- The 'index' function, this function is used to record the address of the sectors that have the smallest cost function and order them to produce the optimal path.

3.2 Wavefront Algorithm

The wave front map technique gives us a straightforward and operative execution that creates a track for the robot with suitable time fulfillment. This algorithm includes a width first search of the graph starting at the target sector until it reaches the start sector. First, sectors of obstacles are indicated with a ‘1’ and the destination sector is marked with any minor number other than ‘1’ such as ‘2’. Wavefront sectors matrix is created either by using 8 sector connectivity or 4 sector connectivity. The obstacle avoiding track is specified by any continuous sequence of decreasing matrix element values that lead to the target [23-25] as shown in Fig.11.

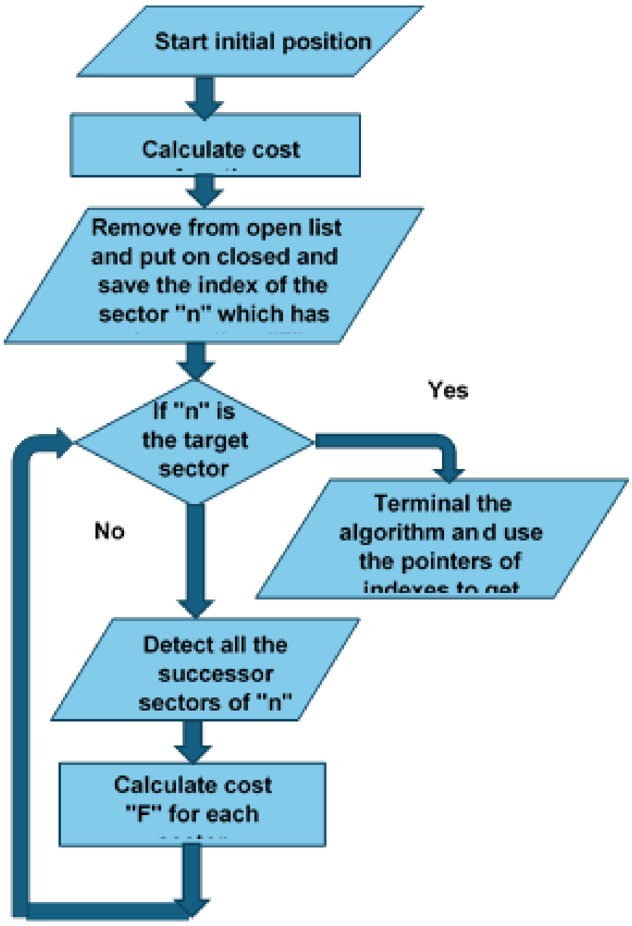


Fig. 10. The flow chart of A_Star algorithm

Step one: give an amount more than ‘1’ to the target position such as ‘2’. Then give the value ‘3’ to four neighbor sectors of the target and value ‘4’ to the neighbor sectors of value ‘3’ [26-28].

Step two: keep increasing the value for each 4 adjacent sectors until reach to start position.

Last step: when the start position is reached, the route will be found by following the values in reverse arrangement from the greatest value (reference position) until the lowest value (end position) except the value ‘1’. All steps illustrated in Fig. (12-14)

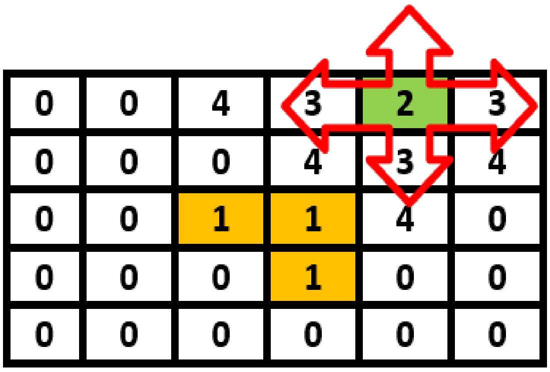


Fig. 12. First step of 4-sector wave front method

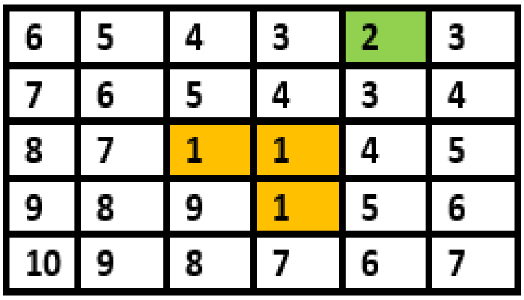


Fig. 13. Second step of 4-sector wave front method

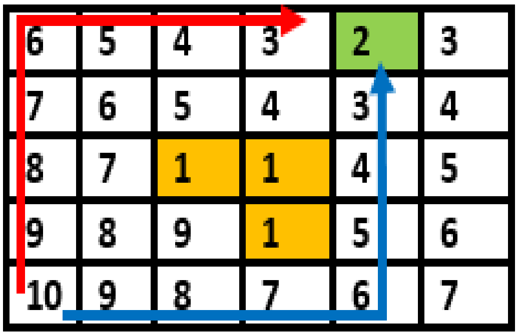


Fig. 14. Last step of 4-sector wave front method

4. Simulation Results and Discussion

In this section, simulation work conducted in this study will be presented. Also, different environments will be shown with A-star. Let's examine A-star algorithms and the 4_sector wave front method with different shapes of the mazes as illustrated from Fig. 15. To Fig 20.

4.1. Different Maze Test

As we can see from Figure 15 to 20, A-Star and 4_sector wave front methods respectively that can handle a circular and hexagonal maze, considering the flexibility of movement and bending angles, the algorithm was able to find the best path after examining the available paths without collision.

Table 1 shows the path time for each maze shape with different path planning methods. Also, the time required to do A- star is longer time than the 4_sector wave front. But the path length of 4_sector wave front is shorter than the A star method.

Table: 1 Time comparison between path planning algorithms with different maze

Method	Circular maze	hexagonal maze
A-star	Time =317.7 sec	Time =254.9 sec
4_sector WAVEFRONT method	Time =2.11 sec	Time =2.11 sec

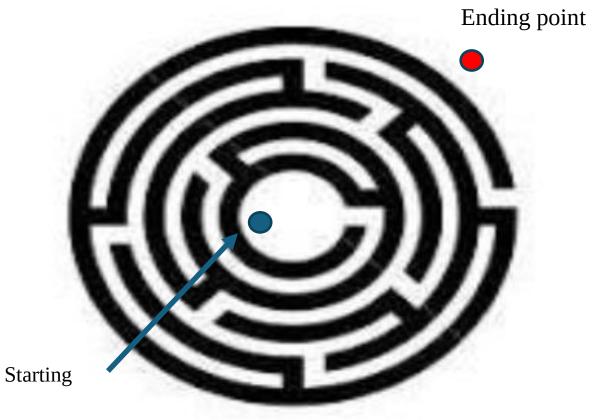


Fig. 15. A maze with a circular shape

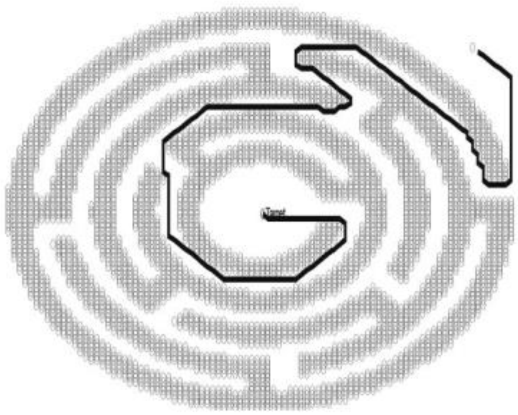


Fig.16. The path of a circular maze using A-star algorithm

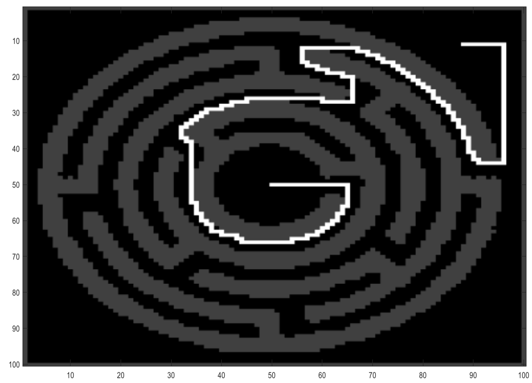


Fig.17. The path of a circular maze using 4_sector wave front method

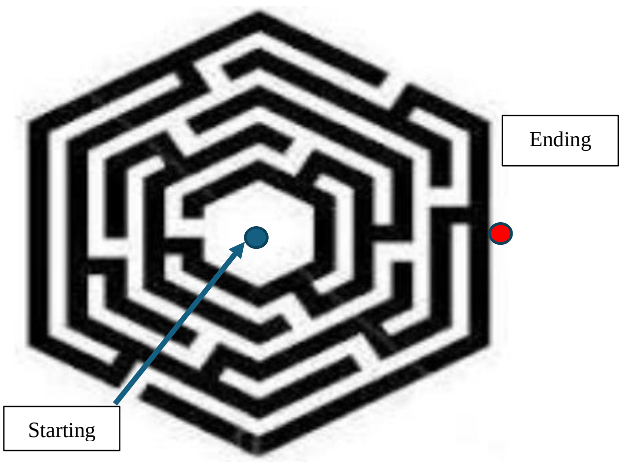


Fig. 18. A maze with a hexagonal shape

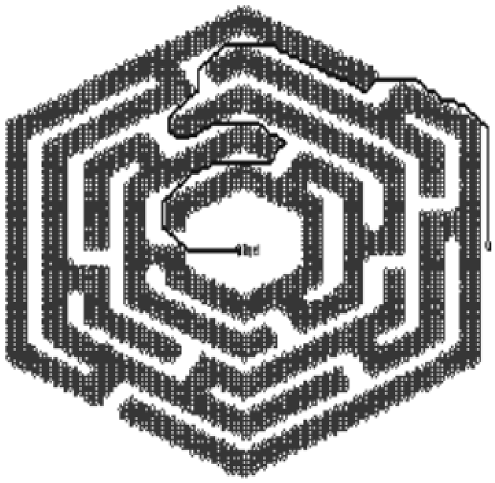


Fig. 19. The path of a hexagonal maze using A-star method

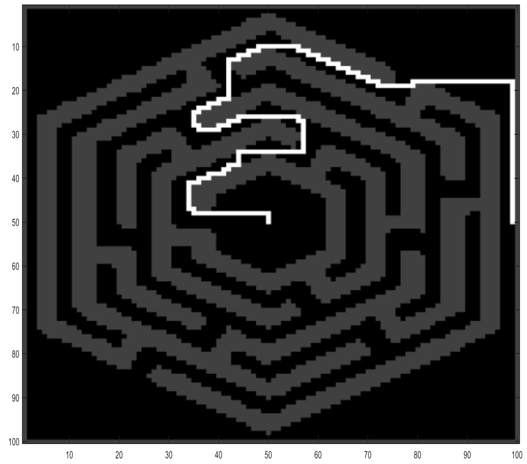


Fig. 20. The path of a hexagonal maze using 4_sector wave front method

4.2. Different Static Environment

Labyrinth-shaped environments are considered one of the most difficult environments for any specialized algorithm to find a movement path due to the presence of a large number of paths, these paths are divided into dead-end paths, long paths far from the target, or short paths close to the target, that makes the algorithm calculations larger and more difficult and it need more accuracy to find the path that leads to the desired goal. The first environment consists of a V shape with some horizontal obstacles as shown in (figure 21 to 23).

The purpose of the second environments (figure 24 to 26) is to show the algorithm's ability to overcome the collision problem, especially when the target is close to the starting point but there is an obstacle that makes the robot choose a path to go around the barrier, which proves that the algorithm chooses the priority of roads based on avoiding collision first and then Choose the shortest route.

Table 2 shows the path time for each static environmental with different path planning methods. Also, the time required to do A- star is biggest time than 4_sector wave front, but the path length of 4_sector wave front is shorter than a star method.

Table: 2 Time comparison between path planning algorithms with different static environmental

Method	1st Environmental	2nd Environmental
A-star	Time =1688.2sec	Time =389.9 sec
4_sector wave front method	Time =7.525 sec	Time =7.665 sec

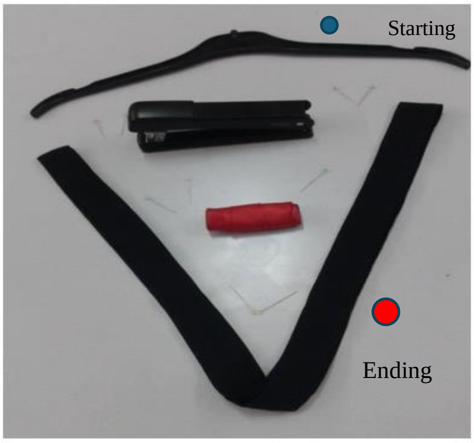


Fig. 21. The first random environment captured by using a camera

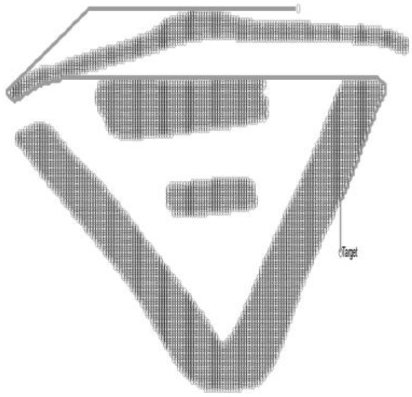


Fig. 22. The path of first environment using A-star algorithm

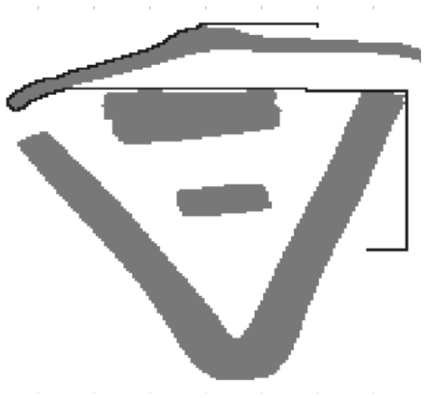


Fig. 23. An optimal path for the fourth environment using 4_sector wave front method

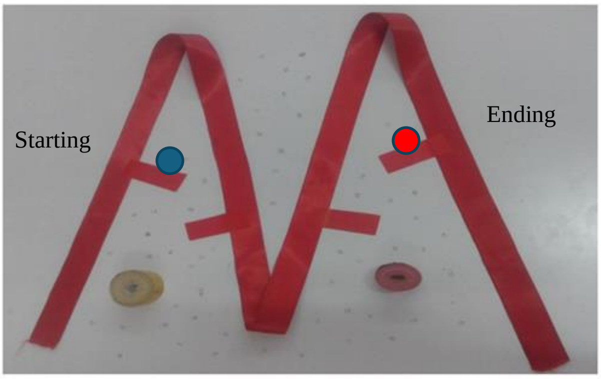


Fig. 24. The second random environment captured by using a camera

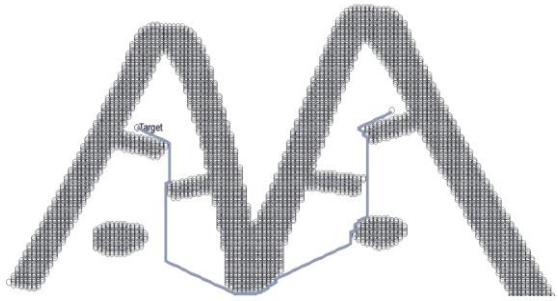


Fig. 25. An optimal path for the second environment using A-star algorithm

Fig. 26.An optimal path for the fourth environment using 4_sector wave front method

5. Conclusions

The potential impact on autonomous robot navigation in real environments is very important. MATLAB simulation for mobile robot trajectory planning has been used in this paper to provide us with the optimal path considering all obstacles that the robot could face. This simulation is based on the A_STAR algorithm, and 4_sector wave front methods are presented effective solutions for basic problems such as static environments, which have different maze, dead ends paths, the shortest path issue and the desired time to find the proper way. The A* algorithm and 4_sector wave front methods are stimulated to achieve the locations of obstacles, starting points, free space, and target points for the robot with zero position error. The wave front method is achieved all paths with less time than A star method, but the A star is shorter path length than the wave front method.

6. Future Work

In the future paper, we will work not only on one robot but also a group of robots that need a monitor system and must use online control algorithm, so cameras will not be enough, other sensors like ultrasonic, infrared and touch sensors will fix on each robot to prevent collision issue, beside that the algorithm should use machine learning and deep learning techniques for better decision-making be smart enough to send orders to all robots in the same time. We still have some limitations like:

- Combining pixels into sectors leads to deform the picture or miss some features of it, these features could be important for the used algorithms, especially when that related to small obstacles which won't consider and lead to collision problem.
- Depending on a horizontal Camera only could cause a problem, when the background and the obstacles have the same color, that make trouble for image processing which based on the colors to extract the features of the image, in this case the supplied image from the camera will not be enough, so we have to use other sensors to overcome this issue.

References

1. Hachour, O.: The proposed genetic FPGA implementation for path planning of autonomous mobile robot. Int. J. Circ. Syst. Sig. Process. 2(2), 151–167 (2008)
2. Jinyu Dai, Jin Qiu, Haocheng Yu, Chunyang Zhang, Zhengtian Wu and Qing GAO: Robot Static Path Planning Method Based on Deterministic Annealing. Machines, Machines 2022, 10(8), 600; www.doi.org/10.3390/machines10080600. (2022)
3. Ferguson, D., Likhachev, M., Stentz, A.: A guide to heuristic-based path planning. School of Computer Science, Carnegie Mellon University Pittsburgh, PA, USA. American Association for Artificial Intelligence (2005). www.aaai.org.
4. B.K. Patle a, Ganesh Babu L b, Anish Pandey c, D.R.K. Parhi d, A. Jagadeesh: A review: On path planning strategies for navigation of mobile robot. Production and hosting by Elsevier B.V. on behalf of China Ordnance Society, 2214-9147, (2019). https://doi.org/10.1016/j.dt.2019.04.011.
5. Sariff, N., Buniyamin, N.: An overview of autonomous mobile robot path planning algorithms. In: Proceedings of 4th Student Conference on Research and Development, pp. 183–188 (2006).
6. Chao Liu , Lei Wu , Guangxin Li, Hao Zhang, Wensheng Xiao, Dengpan Xu, Jingjing Guo, Wentao Li " Improved multi-search strategy A* algorithm to solve three-dimensional pipe routing design" Expert Systems with Applications Volume 240, 15 April 2024.
7. Reddy, H.: Path Finding - Dijkstra's and A* Algorithm's, 13 December (2013).
8. David Ball,Ben Upcroft, Gordon Wyeth, Peter Corke, Andrew English, Patrick Ross, Tim Patten, Robert Fitch, Salah Sukkarieh, Andrew Bate: Vision-based Obstacle Detection and Navigation for an Agricultural Robot. Journal of Field Robotics 33(8), 1107–1130, DOI: 10.1002/rob.21644. Australia, (2016).
9. Y. Wang, B. Du, Y. Shen, K. Wu, G. Zhao, J. Sun, and H. When: "EV-gait: Event-based robust gait recognition using dynamic vision sensors," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), June 2019, pp. 6351– 6360. Doi: 10.1109/CVPR.2019.00652.

10. Yuncheng Lu, Zhucun Xue, Gui-Song Xia & Liangpei Zhang,” A survey on vision-based UAV navigation”, Geo-spatial Information Science, 21:1, 21-32, DOI: 10.1080/10095020.2017.1420509, (2018).
11. Chandak, A., Gosavi, K., Giri, S., Agrawal, S., Kulkarni, P.: Path planning for mobile robot navigation using image processing. Int. J. Sci. Eng. Res. 4(6), 1490–1495 (2013).
12. S. Aggarwal and N. Kumar: “Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges”, Computer Communications, S0140-3664(19)30853-9, (2019), Doi: https://doi.org/10.1016/j.comcom.2019.10.014.
13. Fernando Perez-Sanz, Pedro J. Navarro and Marcos Egea-Cortines,” Plant phenomics: an overview of image acquisition technologies and image data analysis algorithms”, Giga Science, Volume 6, Issue 11, November 2017, gix092, Doi: 10.1093/gigascience/gix092.
14. Shojaeipour, S., et al.: Vision-based mobile robot navigation using image processing and cell decomposition. In: IVIC 2009. LNCS, vol. 5857, pp. 90–96 (2009).
15. Campbell, J., Sukthankar, R., Nourbakhsh, I., Pahwa, A.: A robust visual odometry and precipice detection system using consumer-grade monocular vision. In: Proceedings of ICRA 2005, Barcelona, Spain (2005).
16. Fábio Celestino Pereira, Carlos Eduardo Pereira,” Embedded Image Processing Systems for Automatic Recognition of Cracks using UAVs”, IFAC-Papers Online, Volume 48, Issue 10, 2015, Pages 16-21, https://doi.org/10.1016/j.ifacol.2015.08.101.
17. Chen Suna, Chen Tanga, Xinjun Zhu, Xiaoyu Li, Linlin Wangb, “An efficient method for salt-and-pepper noise removal based on shearlet transform and noise detection”, AEU - International Journal of Electronics and Communications, 1434-8411, Volume 69, Issue 12, December 2015, Pages 1823-1832, http://dx.doi.org/10.1016/j.aeue.2015.09.007.
18. Wenzhou GAO, Lei Yang, Xiaoguang Zhang, Huizhong Liu, “An Improved Sobel Edge Detection”, 978-1-4244-5540-IEEE, 3rd International Conference on Computer Science and Information Technology, Chenddu, China, pp. 67-71, 2010.
19. Akshay Kumar Guruji, Himansh Agarwal, D. K. Parsediya,” Time-Efficient A* Algorithm for Robot Path Planning”, 3rd International Conference on Innovations in Automation and Mechatronics Engineering, Ahmedabad, India, Published by Elsevier Journal, pp.144-149, ICIAME 2016.
20. Basem M. ElHalawany, Hala M. Abdel-Kader, Adly Tag Eldeen, Alaa Eldeen Elsayed, ZakiB. Nossair,” Modified A* Algorithm for Safer Mobile Robot Navigation”, U2013 Proceedings of International Conference on Modelling, Identification & Control (ICMIC) Cairo, Egypt, 31st Aug-2ndSept,pp.74-78,2013.
21. FrantišekDucho, Andrej Babineca, Martin Kajana, Peter Beoa, Martin Floreka, TomášFicoa, LadislavJurišicaa, “Path planning with modified A star algorithm for a mobile robot”, Published by Elsevier Journal, Procedia Engineering 96, pp. 59-69, 2014.
22. C.Wang, L.Wang, and J.Qin, X.Su, W.Li, Z. Lu, and M.Li, Z.Wu, L.Duan, Z.Li, M.Cao and XicuiOu, Y.Wang, J.Long, M.Huang, Y.Li and Q.Wang, “Path Planning of Automated Guided Vehicles Based on Improved A-Star Algorithm”, Proceeding of the 2015 IEEE International Conference on Information and Automation, Lijiang, China, pp.2071-2076, August 2015.
23. Haifeng Wang, Jiawei Zhou, GuifengZheng, Yun Liang, “HAS: Hierarchical A-Star Algorithm for Big Map Navigation in Special Areas”, International Conference on Digital Home, pp.222-225, Guangzhou, China, 2014.

24. Abdel-Nasser Sharkawy "Task Location to Improve Human–Robot Cooperation: A Condition Number-Based Approach" Automation, Vol. 4, pp. 263–290, 2023.
25. Martin Psotka, František Ducho , Mykhailyshyn Roman, Tölgyessy Michal and Dobiš Michal " Global Path Planning Method Based on a Modification of the Wavefront Algorithm for Ground Mobile Robots " Robotics, Vol. 25,pp.1-16 2023.
26. Zhang, B.; Li, G.; Zheng, Q.; Bai, X.; Ding, Y.; Khan, A. Path planning for wheeled mobile robot in partially known uneven terrain. Sensors 2022, 22, 5217.
27. Looi, C.Z.; Ng, D.W.K. A Study on the Effect of Parameters for ROS Motion Planer and Navigation System for Indoor Robot. Int. J. Electr. Comput. Eng. Res. 2021, 1, 29–36.
28. Aggarwal, S.; Kumar, N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. Computer Communications, 2020, 149, 270–299.