International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 5532-5539

Safe Navigation of Mobile Robot Using A* Algorithm

Hasan Mujtaba
Assistant Professor, Electronics & Communication Engineering Department, Ganeshi Lal Bajaj Institute of Technology & Management, Greater Noida-201308, India.

Orcid Id: 0000-0002-4834-4350

Dr. Gajendra Singh
Professor & Head, Electrical and electronics engineering Department, Sharda University, Greater Noida-201308, India.

Orcid Id: 0000-0001-6882-5082

Abstract:

Path planning contributes a very significant role in navigation of a mobile robot. The accuracy of path depends on mapping and localization of an indoor environment. A number of path planning approach are already in applications like A* Algorithms, D* Algorithm (Heuristic Approach), Dijkstra's Algorithm (Deterministic approach), Cell Decomposition Technique etc. The conventional A* algorithm method does gives the shortest path planning but having high probability of collision, if the mobile robot travels or approaches nearer to the obstacle. Under these conditions, it will be difficult for mobile robot to negotiate the clear path through close obstacles in the environment. In this paper a new method of safe and shortest navigation technique of path planning has been introduced using modified A* algorithm by placing the virtual obstacle in the environment. As the size of the obstacles increase in the environment it reduces the 98% probability of collisions with robot as well as obstacle. Simulation results for indoor static environment shows the improvement in safe and the shortest path.

Keywords: Safe navigation, A* algorithm, Path planning, Virtual obstacles.

INTRODUCTION

Path planning is the analytical capability of a mobile robot in static or dynamic environment, that how a robot makes a long term decision to achieve its objective or goal.The trajectory planning of an autonomous mobile robot plays important task for navigation. The trajectory planning is to determine which route the robot has to be taken up or decided towards the target location when robot running or moving. The safe navigation of autonomous mobile robots more to be focused on the task of collision avoidance. The safety of robots has been an issue of discussion in the mobile robots. The safety of a mobile robot depends upon how much the clearance area has been found when it moves from the trajectory of a sharp or closer obstacles. Richardo acatillo et.al has designed modular robotic system using different terrain and obstacle encounter in the path [1]. But, it is not suitable for surveillance in the indoor or acute obstacle environment. Jong-Hun Park and Uk-Youl Huh has given safe global path planning method that is the combination of classical A* and PRM (probabilistic roadmap) algorithms [2]. Which facilitates the analysis of navigation, time and length of the robot path compared to the classical A* algorithm for safe navigation. This added feature in the classical A* is useful for being employed as an obstacle avoidance method. Borenstein and Koren has given new approach for collision free navigation using potential field method based on the attraction and repulsion forces. Which facilitates convenient mathematical analysis and application, it is useful for being employed as an obstacle avoidance method [3]. Christos Alexopoulos et. al. and J. Crowley has been developed a method to improve the practical application of the potential field method. Using the local minimum method, we found a learning technique in which the robot developed a comprehensive model and places close to the grid. An advanced approach planning system is used for the constructing the graph of arena and this so-called graph vision popularly known as "visibility graph method"[4]. In the vision graph approach,(Visibility graph method) the system plans and implements paths by the sequence of right online movements where the path is not always guaranteed shorter. Another problem with this method is that border barriers of an obstacle were also seen as a means for a mobile robot. Vladimir J. Lumelsky et al has suggested algorithms for mobile robot path planning [5]. In this technique, the robot moves directly to the target and follows an obstacle circumference if robot has found an obstacle in the path of motion. While tracking obstacles on the border that the robot is moving towards the target, it stops after the obstacle limits and goes towards the target. The problem with this technique is that it does not meet the shortest path planning requirements. Soh Chin Yun, S.Parasuraman; Velappa Ganpatty worked on the new approach of dynamic path planning algorithm of mobile robot [6]. The study was focused on exploring the algorithms for acute obstacle. They used genetic algorithm for replanning of path, when a dynamic obstacle has been encountered during the motion. This algorithm does not validate about the collision free path planning, if the size of robot has been increased. Dijkstra's algorithm [7-8] is considered an competent method for the shortest path search. Peter E. Hart et.al proposed the algorithm A*, which is identical to that of the Dijkstra's algorithm. But it uses the heuristic values to improve its efficiency in terms of calculating the shortest path [9]. Later, many modifications are made to the A* algorithm and is also modified according to the application for finding the optimal path between two points [10-15]. German A.Vargas et.al has proposed A* path planning algorithm using webot [17]. The paper defines initial or goal positions and obstacle avoidance using simulation environment and diagonal movement of a mobile robot on cell decomposition platform. The paper does not clarify real time path planning of a mobile robot. Yudi Pratama and Wasolim have developed a mobile robot for logistic transport of material in indoor [18].

Wall following algorithm for a mobile robot has been used for the transportation of material using the corner or sides of a room in navigation, the combination of sensor on robot give the exact location of initial or target points. But does not satisfy the obstacle avoidance methods of a mobile robot. The proposed study gives a safe and shortest path planning approach using A* algorithm and virtual obstacles. It avoids the collision of a mobile robot in the environment with different shaped obstacles. Virtual obstacle leads to increase the safe area of environment.

PATH PLANNING METHOD

A. Classical Method of A Algorithm*

A* search algorithm is mainly applicable to a grid-based environment. In the case of A* algorithm the heuristic function H(n) is used that make it more practical than Dijkstra's algorithm [6]. The heuristic function relate to continually changing distance between target and the present location. This heuristic function does not get influenced by various obstacles in forward path of robot which makes the algorithm more effective for identification of the grid. The search for A * algorithm begins with the expansion of the starting node and the stacking of all its neighbors. The stack calculate minimum value of F(n) with heuristic function H(n). The lower cost cell is extracted and expanded. This operation continues until the target node is cleared.

$$F(n) = G(n) + H(n) \quad (1)$$

Where,

N = (Xn,Yn) is the current or expansion cell/ node

$F(n)$ = The minimum value of the cell entry estimated by all paths starting node (S) to the goal node (G)

$G(n)$ = The function of the real cost of the starting node $S(X_0,Y_0)$ is current expansion of the node n..

$H(n)$ = The indicative estimate of the minimum path costs from the current expansion node N to the target node $G(x_G, y_G)$

Now $G(n)$ can be calculated as

$$G(n) = \begin{cases} distance(prev(n), n) & \text{if } prev(n) = S_0 \\ g(prev(n)) + distance(prev(n), n) & \text{else} \end{cases} \quad (2)$$

Where $prev(n) = S_k$, (read as the previous n) and S_k is the current expanding node of the k^{th} step, and in (2)

$$distance(prev(n), n) = 10\sqrt{(x_k - x_n)^2 + (y_k - y_n)^2} \quad (3)$$

And H(n) is expressed as

$$H(n) = 10\sqrt{(x_n - x_G)^2 + (y_n - y_G)^2} \quad (4)$$

Now the lowest cost solution can be found using equations above and set back from goal position G to start position S. The flow chart of classical A* algorithm shows in Fig.1 give the detail about calculating the steps for shortest path using heuristics value H(n) [15]. The experiment carried out for the indoor environment which consists of various obstacles of different forms. The mapping of the network of occupation of the environment is reached. The possibility of cells is converted into the value of '1' for obstacles and '0' for empty cells. For the simulation, the robot's purpose and starting position are defined manually on the grid map. When the goal position is set by setting the numeric value '-2', and the initial position is set by setting '-1'. Now, the traditional A * algorithm is applied to find the path between the goal and the starting node. The simulation results show that the robot passes very close to the obstacles, resulting in a collision with the walls and another object in the environments. If the obstacles are precise or narrower, the probability of collision is very high.

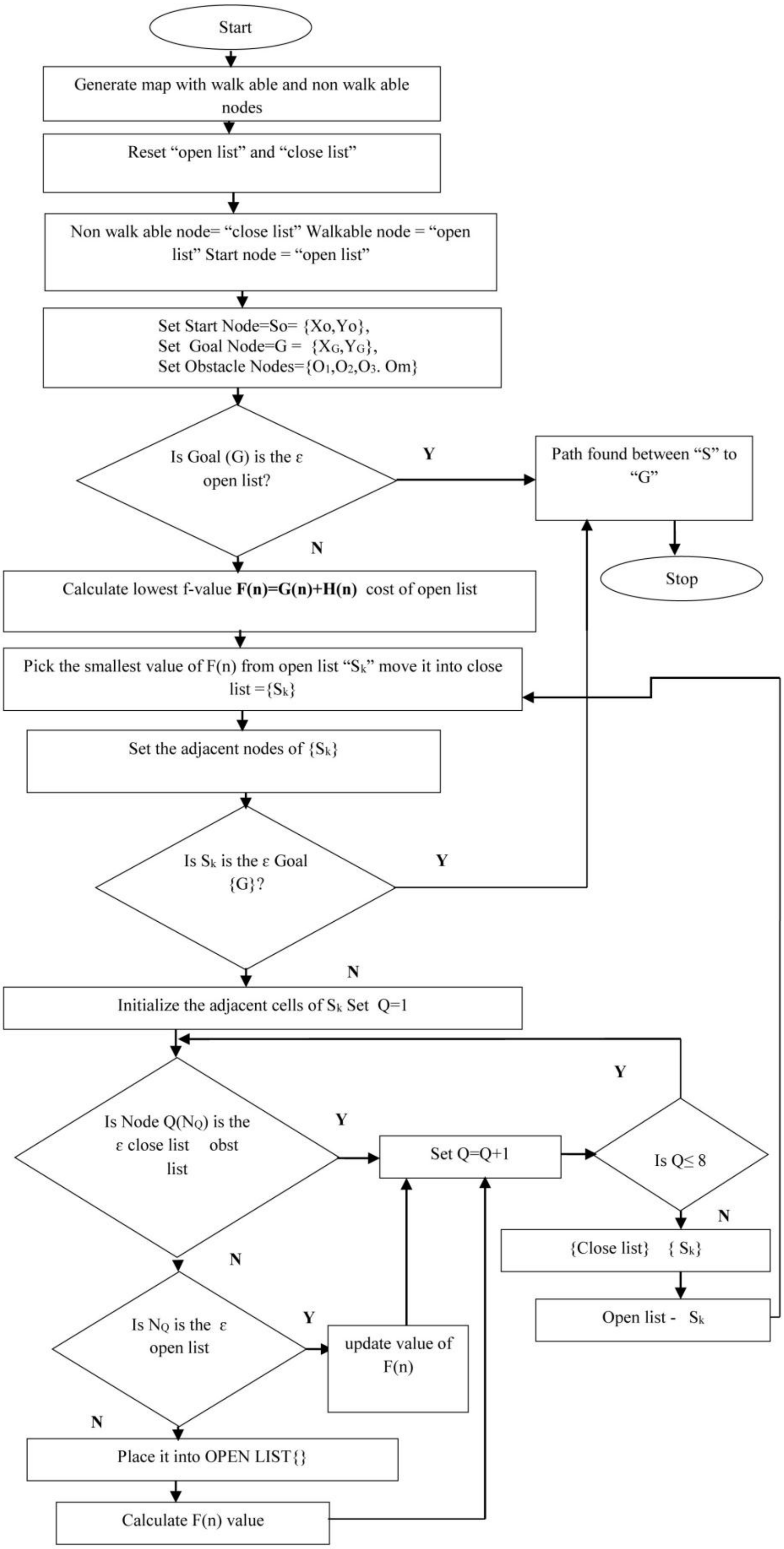

Start
Generate map with walk able and non walk able nodes
Reset "open list" and "close list"
Non walk able node= "close list" Walkable node = "open list" Start node = "open list"
Set Start Node=So= {Xo,Yo},
Set Goal Node=G = {XG,YG},
Set Obstacle Nodes={O1,O2,O3. Om}
Is Goal (G) is the ε open list?
Y
Path found between "S" to "G"
Stop
N
Calculate lowest f-value F(n)=G(n)+H(n) cost of open list
Pick the smallest value of F(n) from open list "Sk" move it into close list ={Sk}
Set the adjacent nodes of {Sk}
Is Sk is the ε Goal {G}?
Y
N
Initialize the adjacent cells of Sk Set Q=1
Is Node Q(NQ) is the ε close list obst list
Y
Set Q=Q+1
Is Q≤ 8
Y
N
{Close list} { Sk}
Open list - Sk
N
Is NQ is the ε open list
Y
update value of F(n)
N
Place it into OPEN LIST{}
Calculate F(n) value

Figure 1:

B. Back Tracking method from source to goal

To find out the optimum Path from source to goal a subroutine is being followed from goal node (G) to start node(S).The process of backtracking in A* algorithm is describe as follow:

Step1:Let Po=G(Goal node), N=0

Step2: IF P ε S(source node) go to step no 5 otherwise go to next step

Step3:P $_{n+1}$=prev(P_n)

Step4:n=n+1 go to step no2

Step5:N=n (Stop)

After completing the above action robot able to draw the shortest path from source to destination.

C. Proposed modified method of safe navigation

For save navigation of a mobile robot some modification has been done in the environment by placing the virtual of obstacle in the closed set marked as “2” at the boundaries of static environment so that the possibility of hitting the obstacles is reduced as shown in Fig2

1	1	1	1	1	1	1
2	2	2	1	2	2	2
0	0	2	1	2	0	0
0	0	2	1	2	0	0
0	0	2	2	2	0	0

Figure 2

D. Modified A algorithm for safe navigation*

In the proposed method, following steps has been used

Step1: Create a map with walkable and non-walkable nodes.

Step2: Consider virtual obstacles value ‘2’ around non-accessible nodes.

Step3: Create “open list” and “closed list” that is initially Empty or reset

Step4: Place all the non-walkable and virtual obstacles node in closed list. Create starting node and target node. Place the starting node on open list

Step5: IF Goal (G) is ε open list, then go to step no: 17 otherwise go to next step

Step6: Calculate the F(n) cost of adjacent cells .

Step 7: Pick the smallest value of F(n) from open list “S_k” move it into close list ={S_k}

Step 8:IF “S_k” ε Goal (G) go to step no: 17 otherwise go to next step

Step9: Set Q=1.

Step10:IF Q ε close list ∪obstacle list ∪ virtual list then Go to next step otherwise go to step no14.

Step11:Set Q= Q+1

Step12:IF Q<= 8 then go to step no 10 otherwise goto next step

Step13: Close = Close ∪{S_k}, Open list =Open -{S_k} and go to step no 7

Step14:IF Node(Q) ε open list the go next step otherwise go to step no16

Step15: Update F(n) value and go to step no11

Step16: Place it in “open list” and calculate F(n) value got step no 11

Step17: Calculate the path using backtracking method

Step18: Stop

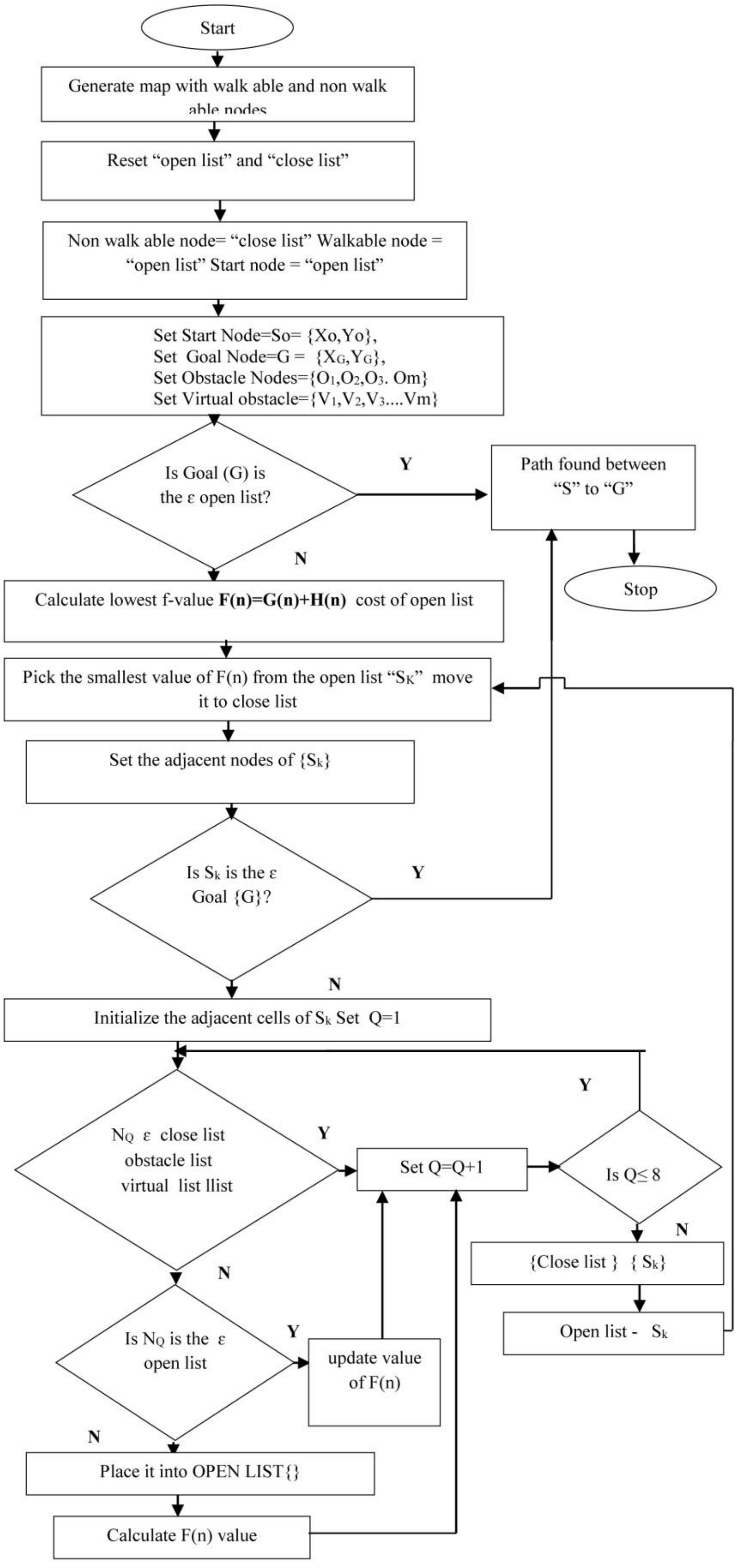

Start
Generate map with walk able and non walk able nodes
Reset "open list" and "close list"
Non walk able node= "close list" Walkable node = "open list" Start node = "open list"
Set Start Node=So= {Xo,Yo},
Set Goal Node=G = {XG,YG},
Set Obstacle Nodes={O1,O2,O3. Om}
Set Virtual obstacle={V1,V2,V3....Vm}
Is Goal (G) is the ε open list?
Y
Path found between "S" to "G"
Stop
N
Calculate lowest f-value F(n)=G(n)+H(n) cost of open list
Pick the smallest value of F(n) from the open list "SK" move it to close list
Set the adjacent nodes of {Sk}
Is Sk is the ε Goal {G}?
Y
N
Initialize the adjacent cells of Sk Set Q=1
Y
NQ ε close list obstacle list virtual list llist
Y
Set Q=Q+1
Is Q≤ 8
N
{Close list } { Sk}
Open list - Sk
N
Is NQ is the ε open list
Y
update value of F(n)
N
Place it into OPEN LIST{}
Calculate F(n) value

Figure 3:

RESULTS

In the experiment of an autonomous mobile robot we consider an indoor static environment in form of occupancy grid with different shaped obstacle. The grid has been constructed and mapped with X=50(cells) and Y=50(cells) size of fixed boundary. The grid is transformed into a logical array by replacing the black cell "1" with obstructions or (occupied cells) and "0" white cells with a free space (cells that are empty). Now locate the robot's initial position and the intended target position on the grid map and the changes (the virtual obstacle lines) are applied to the map using for the purpose of safe navigation of the mobile robot.

A. Simulation Results

The result of classical A* algorithm is shown in Fig.4, Fig5. In the simulation made on MATLAB programming shows the high probability of collision to hit with an obstacle. The results of safe algorithm shown in the Fig. 6,7, and Fig8, by putting virtual obstacle in the environment In Fig. 4, it can be seen that the path required by the classical algorithm A* has enabled the mobile robot to get closer to the obstacle which is an edge of the dangerous environment. In Fig.6,7, result shows that the safe path planning method for the mobile robot increased the clearance area and safety between the obstacle and robot. Now, if the virtual obstacle placed to the static obstacle exactly equal to cell size shown as red line Fig 6, 7, then robot gets the safe path of propagation without collision. If the obstacle size is increased more than the cell size then another path executed and the result is displayed in Fig.7. But, the walkable area get reduced because, it closes the path of closure obstacle due to virtual cells. In Fig.8, it is observed that robot calculate the shortest path and allows the mobile robot to move in the safe environment to the closed obstacles. In such situations the mobile robot may choose only the path which is safe otherwise, it gives the result that no path exists in a environment. Two other paths are compared by using classical A* algorithm and the secure propagation of mobile robot method, as shown in Fig9. In Fig8, it is found that the safe area of robot has been found where robot can navigate easily without the collision.

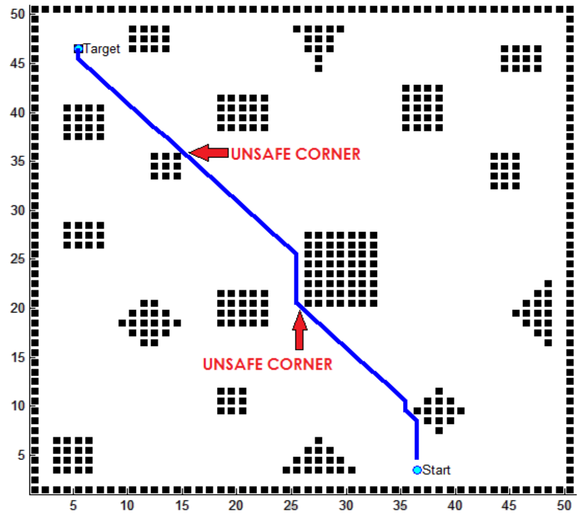
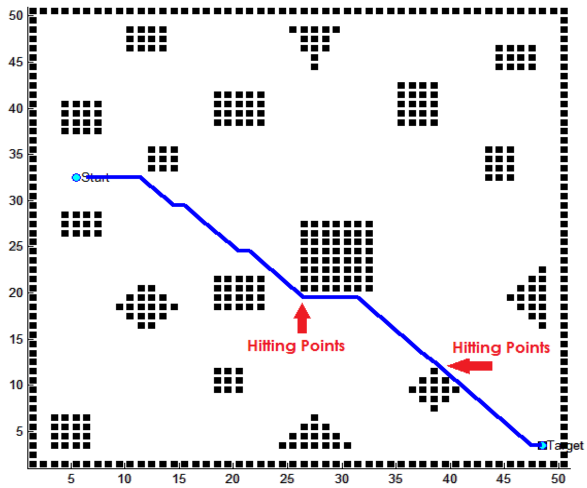


Figure 4: Unsafe Path Planning Using classical A*



F

Figure 5: Unsafe Path Planning Using classical A*

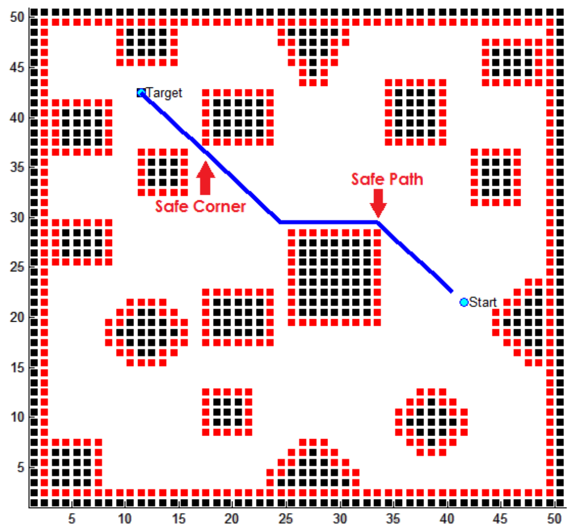


Figure 6: Safe Path Planning by Placing Virtual Obstacle

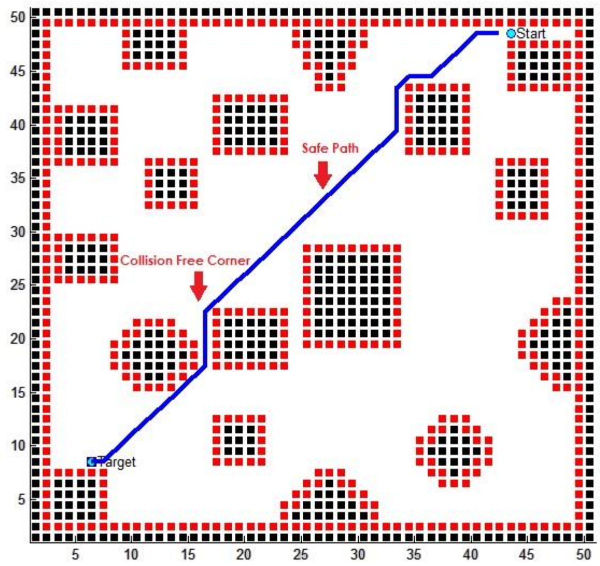


Figure 7: Safe Path Planning Cell Size Increased One Time

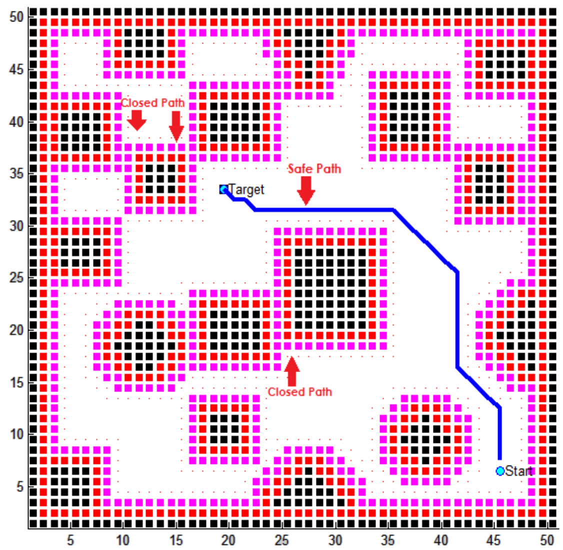
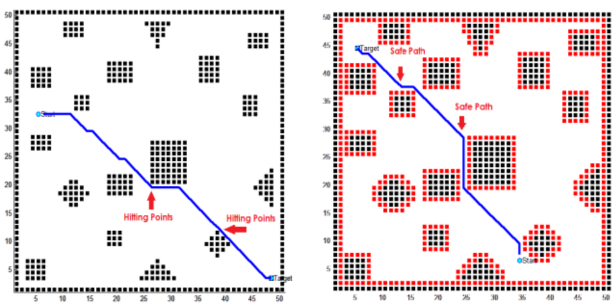


Figure 8: Unsafe Path Closed by Increasing Virtual Obstacles



(a) Classical A* algorithm (b)Safe path planning

Figure 9: Comparison of Classical A* and Safe path planning method

CONCLUSIONS

In this article, the method of path planning of a mobile robot is proposed based on the spread of safely using the A* algorithm to find out the safest and shortest path for mobile robots in a static environment with different shaped obstacles. The technique has been used by placing virtual obstacles to reasonably increase the volume of obstacles and closes the dangerous routes. With the implementation of the proposed method, the robot does not encounter with the stationary obstacles found in the environment of the grid. It provides an example of how to evaluate a safe route planning and a classical A* simulation algorithm. Although the A* algorithm can find out the shortest routes but, it can allow the mobile robot to collide with obstacles in a real environment. In the proposed method the path is safe and shorter which is a good advantage that makes this algorithm even more relevant than another techniques or methods.

REFERENCES

[1] Richardo acatillo et.al "Implementation and assembly of a robotic module for mecabot-3 reconfigurable system" International Journal of Applied Engineering Research 2016 vol-11 no.21 pp 10681- 10684

[2] Jong-Hun Park and Uk-Youl Huh "Path Planning for autonomous mobile robot based on safe space" Journal Electrical Engineering Technology.2016; vol .11 pp1921-718

[3] Bornstein, J. and Y. Koren, Real-time obstacle avoidance for fast mobile robots. Systems, Man and Cybernetics, IEEE Transactions on,1989. 19(5): p.p 1179-1187.

[4] Christos Alexopoulos and Paul M. Griffin, "Path planning for a mobile robot," IEEE transactions on Systems, Man and Cybernetics, vol.22, no.2, pp.318 – 322 ,1992

[5] Vladimir J. Lumelsky and Alexander A. Stepanov, "Dynamic path planning for a mobile automaton with limited information on the environment," IEEE transactions on Automatic Control, vol.3, no.11, pp.1058 – 1063, 1986

[6] Soh Chin Yun ,S.Parasuraman and Vellapa Ganpatty "Dynamic path planning algorithm in mobile robot,"IEEE symposium on industrial electronics application ,and vol 11, pp.364-369,2011

[7] M. Barbehenn, "A note on the complexity of dijkstra's algorithm for graphs with weighted vertices" IEEE Transactions on Computers, vol.47, no.2, 1998

[8] E. W. Dijkstra, Numedsehe Yathematik "A Note on Two Problems in Connecting with Graphs,", vol.1, pp.269-271, 1959

[9] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost

paths in graphs", IEEE Transactions of Systems Science and Cybernetics. vol.4, no.2, pp.100-107, 1968

[10] M. Fu and B. Xue, "A path planning algorithm based on dynamic networks and restricted searching area," IEEE International Conference on Automation and Logistics, pp.1193-1197, 2007

[11] I. Chabini and L. Shan, "Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks," IEEE transactions on intelligent transportation systems, vol.3, no.1, pp.60-74, 2002

[12] M. Nakamiya, Y. Kishino, T. Terada and S. Nishio, "A route planning method using cost map for mobile sensor nodes," International Symposium on Wireless Pervasive Computing, pp.168-174, 2007

[13] E. P. F. Chan and N. Zhang, "Finding shortest paths in large network systems," 9th International Conference on Advances in Geographic Information Systems, ACM Press New York, NY, USA, 2001

[14] M. Hashemzadeh, "A fast and efficient route finding method for car navigation systems with neural networks," 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06), pp.423-426, 2006

[15] G.E. Jan, K. Y. Chang, and I. Parberry, "Optimal path planning for mobile robot navigation,“ IEEE/ASME Transaction on Mechatronics vol.13, no.4, pp. 451-459, 2008

[16] Chia-Jun Yu; Yi-Hong Chen; Ching-Chang Wong, "Path planning method design for mobile robots", Proceedings of SICE annual conference, pp.1681 – 1686, 2011

[17] German A. Vargas et.al "Simulation of e-puck path planning of Webot" International Journal of Applied Engineering Research 2016 vol-11 no.19 pp 9772-9775

[18] Yudi pratama & Wansu lim "Development application of indoor logistics transportation using amr" International Journal of Applied Engineering research 2016 vol-11 no.20 pp 10183-10189.