Path Planning Algorithm in Complex Environment: A Survey

Atikah JANIS & Abdullah BADE*
Mathematics, Graphics and Visualization Research Group (M-GRAVS), Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu, Sabah, MALAYSIA.
*Corresponding author: abb@ums.edu.my; Tel: +6088-320000; Fax: +6088-435324.

Received: *16 June 2015*
Revised: *29 July 2015*
Accepted: *8 August 2015*
In press: *5 November 2015*
Online: *1 April 2016*

Keywords:
Path planning, complex environment, autonomous robot

Abstract

Path finding algorithm is a very challenging problem for navigating autonomous virtual robots in complex environment. A reliable navigation system must be able to identify the virtual robot current location, avoid any collisions and determine the smooth path trajectory of the object. At present, the needs to produce systematic and efficient path finding algorithm with impressive collision scheme has led number of researchers to conduct various experiments to improve and modify the existing algorithms in order to solve several issues in path planning algorithm with collision avoidance for autonomous virtual robot. This paper presents series of path planning algorithms for the last 10 years in order to solve the navigation of autonomous virtual robot in complex environment. We believe that all algorithms reviewed in this paper will give researchers in the field of virtual environment, collision detection and robotic about some fundamental background, issues and challenges on how navigation procedures of autonomous virtual robot in such a complex environment works.

Introduction

In general, path planning requires three vital concept; sensing, learning and reasoning (Gireesh *et al.*, 2010). Navigation of virtual robots is a complicated issue due to the fact that a variety of obstacles have to be detected and an efficient collision free path must be chosen (Amir & Habib, 2010). In real world, navigation of robots requires specific devices such as sensor and camera to determine the reliable distances of robot from any obstacles that comprises the environment (Shyba & Tauseef, 2015). In order to develop a virtual robot navigation system with better collision avoidance function, it requires higher computational time due to the fact that hundreds of testing must be conducted in advance throughout the whole navigation procedures. This is the locus standee to moving forward on the development of smooth path planning algorithm with efficient collision avoidance system in virtual worlds and of course will open hundreds of new potential researches in the field of virtual environment, collision detection between complex 3D model and robotics area. The ultimate aim in this perspective is of course particularly in highly complex virtual environment; to reduce that senseless computational and processing time, and at the same time activating highly efficient algorithms and techniques with zero dependency to specific devices required to running the developed system.

The navigation of virtual robot in complex environment can be performed normally in 2-dimensional environment setting comprising thousands of polygons. Figure 1 shows several example of complex environment in 3D form.

Figure 1: Example of complex environment in 3D form

Literature review

Path planning algorithm with collision avoidance technique had been receiving considerable attention over the past twenty years and many algorithms have been developed. Some of the techniques not only need to detect the obstacles but also need to perform quantitative measurement which concerning dimensions of the obstacles (Borenstein & Koren, 2014). Basically, path planning algorithms can be divided into two categories namely global path-planning and local path-planning. Global path-planning or offline path-planning requires complete information of the obstacles and the environment. Hence, this enables the system to generate complete trajectory from start to goal even before the robot starts motion (Leena & Saju, 2014). This concept is differ from the local path-planning where the environment and obstacles information are completely unknown. Therefore, the robot gets information through sensors as it travels throughout the environment and algorithm is needed to develop path to reach target point.

There are various issues need to be considered in navigation of virtual robots due to various purpose and function of the virtual robot itself. Most of proposed approaches are focusing on finding the shortest path from the initial position to goal. Recently, researches are focusing on reducing the computational time and enhancing smooth trajectory of the virtual robot (Zeyad *et al.*, 2015). Other ongoing issues include navigating the autonomous robots in complex environment, considering movable obstacles, navigation of multi-agent robot and enhancement of natural motion for the robot movement. Figure 2 shows related issues in path planning.

TRANSACTIONS ON SCIENCE AND TECHNOLOGY

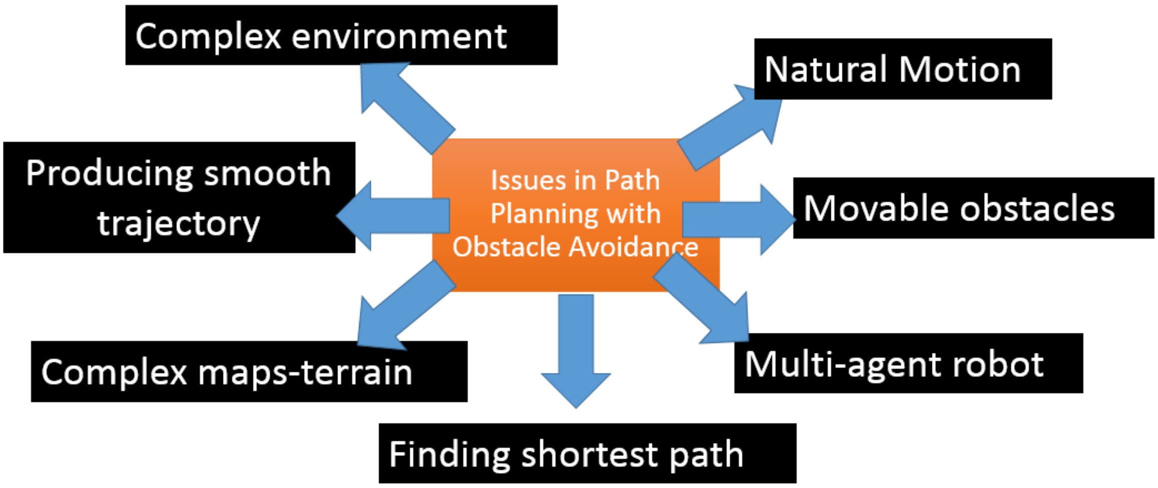


Figure 2: Issues in path planning

Finding shortest path is the issue of finding the optimal path is with minimum path length, with minimum reaching time and minimum control effort. Researchers are focusing on this issue in the first place to ensure the robot to move to goal in very short time. A* search (Peter *et al.*, 1968) is a well-known best-first graph search algorithm for finding a correct and complete path from the initial position to the goal. A* works by expanding the vertices inside the map and searching for the nodes with lower estimated distance to the goal. The ability of this A* search algorithm to be manipulated in many ways leads to the development of many path planning techniques.

Numerous number of graph search algorithm developed over the last decades have been tested for path planning of autonomous robot in complex environment like terrain maps (Ouanezar *et al.*, 2008). D* Search (Anthony, 1994) and focused D*(Anthony, 1994) shows that they can be used for dynamic path planning. However, it requires longer calculation and several replaning operations (Ouanezar *et al.*, 2008). Artificial Potential field algorithm (Khatib, 1986) also being tested for complex environment, however suffer in local minima problem. Basic theta* (Nash, 2007) is reliable to be used for any angle path-planning algorithm but spending long time in path planning calculation. Another approach, Field D* (Anthony, 1994; Ferguson & Stents, 2006) is the extension of D* and D* Lite which uses linear interpolation in order to generate paths with low-cost. Nash (2007) had also develop AP theta* which is better that Theta* and Field D* in finding shortest path.

Several heuristic approaches have been developed to navigate robot in complex environment such as the use of genetic algorithm introduced by Holland (1975). Genetic algorithm uses the concept of chromosome as binary string to code each candidate solution (Holland, 1975). Neural Network was used by Zacksenhouse *et al.*, (1988) which algorithm can be used for dynamic environment (Zelinsky, 1994). The idea of collision avoidance in neural network is that the target attracts the robot while obstacle pushes away the robot (Amir, 2010). Several of these algorithms are combined to achieve maximum result and resolving respective issue (Ouanezar *et al.*, 2008). Combination of neural networks and fuzzy logic has been proposed by Payeur *et al.* (1994) generally developed for high level control of robots. In 1998, genetic-fuzzy approach was introduced which benefits in finding the optimal travel time and controlling the motion of a robot while avoiding moving obstacles (Pratihar *et al.*, 1998).

Most of the previous mentioned techniques are used for navigating autonomous virtual robot in complex environment. The biggest main concern in this paper is the techniques used when the complex environment is completely unknown by the autonomous robot.

Techniques for path planning with collision avoidance in navigating virtual robots in complex environment

Navigating mobile robots in complex environment is one of the challenges in robotics. The latest techniques which is focusing on solving this issue include Bug Algorithm, path planning using Corridor Maps, Follow the Gap Method, New Hybrid Navigation Algorithm, Intelligent Bug Algorithm and Intelligent Follow the Gap Method. Artificial Potential Field Method has been developed few decades ago. Figure 3 shows main classification of path planning algorithm in complex environment.

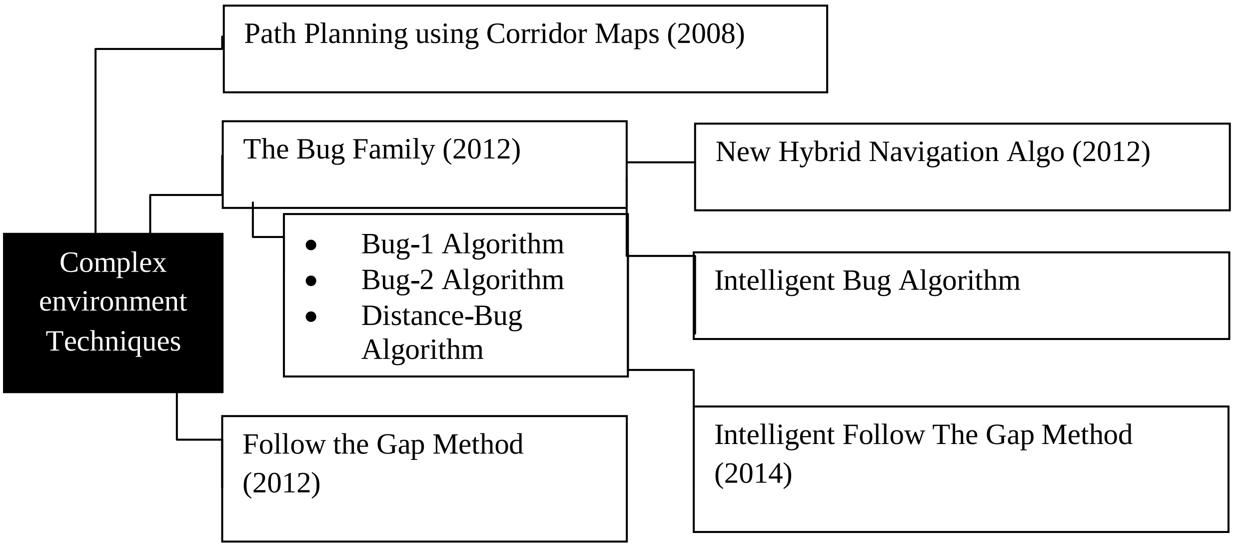


Figure 3: The main classification of path planning algorithm in complex environment

Path planning using corridor maps

Initially, the path planning using corridor maps (Overmars, 2007) is developed to enhance smooth trajectory of the robot moving from the start point to target point. At the same time, it is also beneficial to be applied for solving path planning with complex environment. The corridor map is a map whose edges represent collision-free corridors, consists of a backbone path and a set of boundary circle centered on the path. This method uses one force function which can steer the character toward the goal and keeps it inside corridor. Hence, the corridor map ensures the robot to move smoothly on its path regardless of how complicated the environment is. It was proven that this technique is fast and flexible resulting in a very reasonable final path.

TRANSACTIONS ON SCIENCE AND TECHNOLOGY

This algorithm was then improved in which uses two forces; boundary force and steering force. In this improved method, boundary force will push the character away from the boundary of the corridor and steering force guides the character toward the goal. As a result, this method has produced natural paths for a large number of characters in complicated environments (Overmars, 2008). However, this technique needs reference path to be specified in order for the robot to find from initial position to goal position.

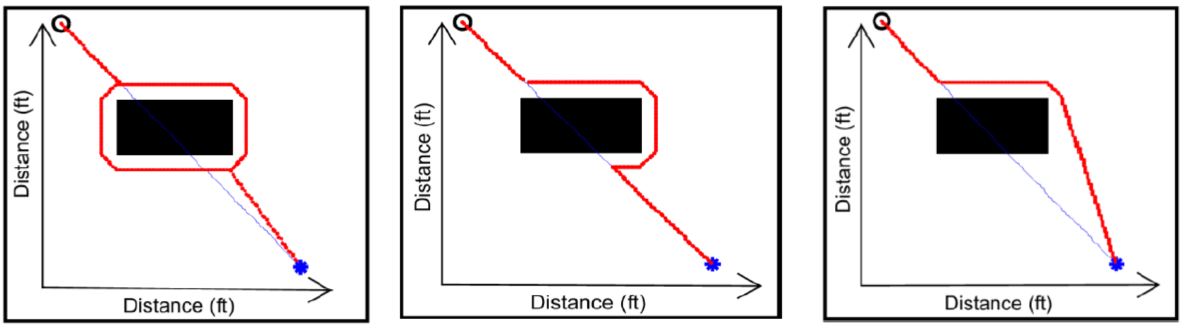
The bug algorithm

The Bug Algorithm is a complete algorithm (Sezer, 2012) used in moving character especially in complex environment where it plans direct path from source to destination until it faces an obstacle. There are 3 versions of this method; Bug-1 Algorithm, Bug-2 Algorithm and Distance-Bug Algorithm. Each version in Bug algorithm carries its own termination property (Zohaib *et al.*, 2012). It has two behaviors; move to goal mode and obstacle avoidance mode. Initially, the robot is in 'move to goal' mode and changes to 'obstacle avoidance' mode when it faces an obstacles (Yufka, 2009). At this stage, the robot will move along the edge of obstacle and compute new path from the leaving point (x_1, y_1) to destination (x_2, y_2). The slope and intercept 'c' can be achievable from the following formula.

$$m = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \quad (1)$$

$$c = y_1 - m \times x_1 \quad (2)$$

Figure 4 shows the results from the a) Bug-1 Algorithm, b) Bug-2 Algorithm and c) Dist-Bug Algorithm.



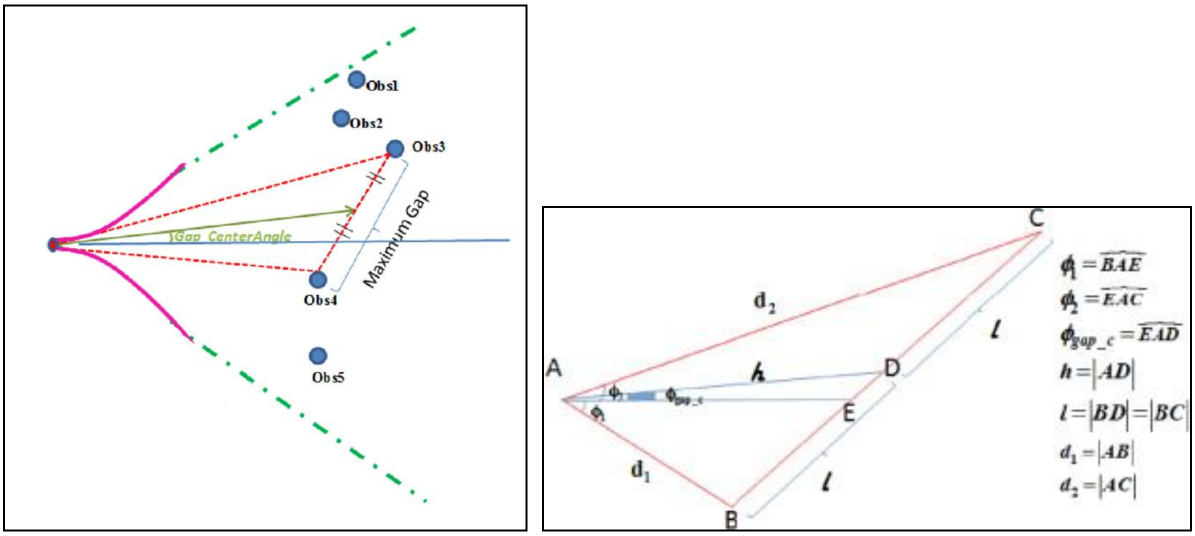
a) Bug-1 Algorithm b) Bug-2 Algorithm c) Dist-Bug Algorithm

Figure 4: Comparison between three version of Bug Algorithm (Zohaib *et al*., 2012)

The Bug Algorithm is the simplest sensor-based technique and has some advantages where it computes minimum distance to destination in a complex environment and does not suffer from local minima problem (Zohaib *et al.*, 2012). However this technique has some disadvantages where this algorithm assumes the robot as a point and the trajectories are sometimes very long which took longer time for the robot to reach goal (Zohaib *et al.*, 2014).

Follow the Gap Method (FGM)

Follow the Gap Method (FGM) (Sezer *et al.*, 2012) in one of the method for navigating the character which works by finding the widest gap among the obstacles and allows the robot to move through center of the obstacles. It also calculates the best heading vector through the gap and finally calculates the final angle (Zohaib, 2014). In FGM, the obstacles are firstly considered as circular objects. This method requires the calculation of gap center angle, ϕ *gap_c* with the parameters d_1, d_2, ϕ_1, ϕ_2 and final heading angle. Parameter ϕ_1 and ϕ_2 are the angles of obstacles to the maximum gap while d_1 are d_2 the distances from obstacles to the maximum gap. Hence, the maximum gap is determined from the generated gap array. Figure 5 (a) and (b) shows a triangle is illustrated to represent the gap center angle.



(a) (b)

Figure 5: Illustration of triangle representing Gap Center Angle (Sezer *et al.*, 2012)

The gap center angle can be obtained by firstly applying the Cosine Rule into the ABC triangle.

$$(2l)^2 = d_1^2 + d_2^2 - 2d_1d_2\cos(\phi_1 + \phi_2)$$
$$l^2 = \frac{d_1^2 + d_2^2 - 2d_1d_2\cos(\phi_1 + \phi_2)}{4}. \qquad (3)$$

Then, the Appollonius theorem is applied to the ABC triangle.

$$d_1^2 + d_2^2 = 2l^2 + 2h^2. \tag{4}$$

Replacing l^2 with Equation (3)

$$h^2 = \frac{d_1^2 + d_2^2 + 2d_1 d_2 \cos(\phi_1 + \phi_2)}{4}. \tag{5}$$

Finally the Cosine Rule is applied once again to the ABD triangle

$$l^2 = d_1^2 + h^2 - 2d_1 h \cos(\phi_1 + \phi_{gap\_c}). \tag{6}$$

Replacing l^2 and h^2 with equation (3) and (5);

$$\frac{d_1^2 + d_2^2 - 2d_1 d_2 \cos(\phi_1 + \phi_2)}{4} = d_1^2 + \frac{d_1^2 + d_2^2 + 2d_1 d_2 \cos(\phi_1 + \phi_2)}{4} - 2d_1 \frac{\sqrt{d_1^2 + d_2^2 + 2d_1 d_2 \cos(\phi_1 + \phi_2)}}{2} \cos(\phi_1 + \phi_{gap\_c})$$

$$\phi_{gap\_c} = \arccos\left(\frac{d_1 + d_2 \cos(\phi_1 + \phi_2)}{\sqrt{d_1^2 + d_2^2 + 2d_1 d_2 \cos(\phi_1 + \phi_2)}}\right) - \phi_1. \tag{7}$$

Finally, the gap center angle is achieved and this is where the robot moves between two obstacles. By moving towards the biggest gap, safe trajectory is ensured (Sezer *et al.*, 2012). Another main advantage is that FGM does not suffer from local minima problem (Zohaib *et al.*, 2012).This technique is one of the new approach in navigating autonomous robot in complex environment, however this method gives a drawback which also cannot avoid U and H-shaped obstacles (Zohaib *et al.*, 2014).

New Hybrid Navigation Algorithm (NHNA)

The New Hybrid Navigation Algorithm (NHNA) (Zhu *et al.*, 2012) is the improved method of Distance-Bug Algorithm. This improved technique consists of two layers, deliberative layer and reactive layer. In the deliberative layer, the A* search is being used to find the path to goal. On the other hand, the reactive layer steers the robot on the path planned by the deliberative layer, ensuring obstacle avoidance by using distance-histogram bug algorithm (Zhu *et al.*, 2012). As a result, the combination of these two layers has successfully control the robot's behavior in less computational time. However, some drawbacks occur such as it is unable to avoid obstacles with U and H shape, requires prior information of the environment and in some cases it may carry the robot away from its trajectory position (Zohaib *et al.*, 2014).

Intelligent Bug Algorithm (IBA)

Intelligent Bug Algorithm (IBA) (Zohaib *et al.*, 2012) is a technique improved from previous Bug Algorithm which offers relatively lesser time for the robot to reach goal (Zohaib *et al.*, 2014). It is specifically built to solve navigation of autonomous robot in maze-like environment. In this method, the robot simply moves from start to goal by following a reference path generated. When it faces an obstacle, it will move along the edge of the obstacle until it reaches a leaving point. The leaving point decision is based on the point where obstacle-free path is sensed, then it continues to move towards goal (Zohaib *et al.*, 2013). IBA is a goal-oriented algorithm and offers smoother trajectory in contrast with other Bug Algorithm.

Intelligent Follow the Gap Method

Newest technique, Intelligent Follow the Gap Method (IFGM) is developed in 2014 by Zohaib *et al.*, works also by finding the gap between obstacles. Unlike previous techniques, it is specifically develop to avoid obstacles with U and H shape. It use the same method in Follow the Gap method, where center angle and maximum gap need to be computed. The main difference is that the movement of the autonomous robot consists of two cases. The calculated distance of stored obstacles from the robot *(obs.dist)* and the maximum gap (max. gap) are compared to d_{th} and d_{obs}. The d_{th} represent the value greater than the width of a squared shaped robot or radius of the robot, and d_{obs} is set to be the value higher than the speed of the robot (Zohaib *et al.*, 2014). This method need to check whether,

$$obs.\ dist > d_{obs}$$
$$max.\ gap > d_{th}$$

If the condition is true, then the autonomous robot will move using Follow the Gap Method and if otherwise it will move using Intelligent Bug Algorithm (IBA). The IBA is used as a solution in cases where local minima will occur (Zohaib *et al.*, 2014). The example of results are shown in Figure 6.

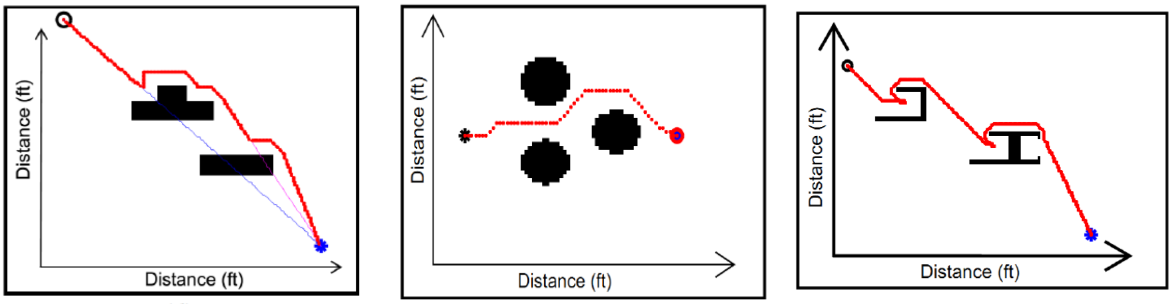


Figure 6: The results of Intelligent Follow the Gap method (Zohaib *et al.*, 2014)

This method can solve the previous problem which the difficulty in avoiding obstacles with U and H shape. Also, this technique do not suffer from local minima problem and no prior information is of the environment is required (Zohaib *et al.*, 2014).

Application in 10 years

This path planning with collision avoidance techniques mentioned in this paper are focussing on its implementation in virtual world, so it can be tested in many times before implementing them in mobile robots from the real world. These real world application areas include manufacturing industries, assembling industries, exploration, spying and monitoring (Zeyad *et al.*, 2015). It can also be implemented in transportation system especially in aircraft traffic control, autonomous cars and underwater vehicles (Abrar *et al.*, 2015). Other than that, the concept of path planning with collision avoidance in virtual worlds can also be used in computer games, simulations, city models and on-line games.

Conclusion

These techniques can be used for avoiding obstacles in complex environment, each of them has their own advantages and disadvantages. Intelligent Follow the Gap Method is the best method so far which can solve the local minima problem and can avoid obstacles with U and H shape. However, these techniques mainly did not focus on other issues like ensuring natural motion of the autonomous robot. Therefore, this can be the future research study in this field.

References

[1] Abrar, M., Marwah, M. & Khaled, M. (2015). Multi-Sensor Based Collision Avoidance Algorithm for Mobile Robot. *IEEE Long Island Systems, Applications and Technology LISAT2015*. 1 May, 2015. Long Island, New York.

[2] Amir, H. & Habib, I. (2010). Evolutionary Approach for Mobile Robot Path Planning in Complex Environment. *IJCSI International Journal of Computer Science Issues*, **7**(4), 1694-0784

[3] Anthony, S. (1994). Optimal and Efficient Path Planning for Partially-Known Environments. *IEEE International Conference on Robotics and Automation (ICRA '94).* 8-13May, 1994. San Diego, California.

[4] Borenstein, J. & Koren Y. (1991). The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots. *IEEE Transactions on Robotics and Automation*, **7**(3), 278-288.

[5] Carsten, J., Rankin, A., Ferguson, D., & Stentz, A. (2007). Global Path Planning on Board the Mars Exploration Rovers. *Journal of Field Robotics*, **26**(4), 337-357

[6] Ferguson, D. & Stents, A. (2006). Field D*: An Interpolation-based Path Planner and Replanner. *Proceedings of the International Symposium on Robotics Research (ISRR)*. 12-15 October 2005. San Francisco.

[7] Geraerts, R. & Overmars, M. (2007). The corridor map method : A general framework for real-time high-quality path planning. *Computer Animation and Virtual Worlds*, **18**, 107-119

[8] Geraerts, R. & Overmars, M. (2008). Flexible Path Planning Using Corridor Maps. *Algorithms - ESA 2008: 16th Annual European Symposium.* 15-17 September 2008. Karlsruhe, Germany.

[9] Gireesh, K. T., Poornaselvan, K. J. & Sethumadhavan, M. (2010). Fuzzy Support Vector Machine-based Multi-agent Optimal Path Planning Approach to Robotics Environment. *Defence Science Journal*, **60**(4), 387-391

[10] Hart, P., Nilsson, N. J. & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans Syst. Sci. Cybernetics*, **4**, 100-107.

[11] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* MIT Press Cambridge, MA, USA.

[12] Kalarani, G. (2014). A Survey of the Various Path Planning Techniques Used in the Navigation of Autonomous Mobile Robot. *Indian Journal of Applied Research*, **4**(10), 442-444.

[13] Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, **5**(1), 90-98

[14] Leena, N. & Saju, K. K. (2014). A Survey on Path Planning Techniques for Autonomous Mobilerobots. *IOSR Journal of Mechanical and Civil Engineering.* 76-79

[15] Nash, A., Daniel, K. & Koenig, S. (2010). Theta* : Any-Angle Path Planning on Grids. *Journal of Artificial Intelligence Research*, **39**, 533-579

[16] Ouanezar, A., He X. & Gang Zhao. (2008). *Survey and the Relative Issues on the Path Planning of Mobile Robot In Rough Terrain* (www.paper.edu.cn). Accessed on March 2015.

[17] Payeur, P., Le-Huy, H. & Gosselin, C. (1994). Robot Path Planning using Neural Networks and Fuzzy Logic. *Proceedings of IECON'94 - 20th Annual Conference of IEEE Industrial Electronics*, **2**, 800-805.

[18] Pratihar, K., Deb, K. & Ghosh, A. (1999). A Genetic-Fuzzy Approach for Mobile Robot Navigation Among Moving Obstacles. *International Journal of Approximate Reasoning*, **20,** 145-172.

[19] Sezer, V. & Gokasan, M. (2012). A Novel Obstacle Avoidance Algorithm: Follow the Gap Method. *Robotics and Autonomous Systems*, **60**(9), 1123-1134.

[20] Shyba, Z. & Tauseef, G. (2015). A Path Planning Technique for Autonomous Mobile Robot using Free-Configuration Eigenspaces. *International Journal of Robotics and Automation*, **6**(1), 14 - 28

[21] Stentz, A. (1994). *Optimal and Efficient Path Planning for Partially-Known Environments, Intelligent Unmanned Ground Vehicles*. Springer US

[22] Zacksenhouse, M., DeFigueiredo, R. J. P. & Johnson, D. H. (1988). A Neural Network Architecture for cue-based motion planning. *Proceedings IEEE International Conference on Decision and Control.* 7-9 December, Austin.

[23] Zelinsky, A. (1994). Using Path Transforms to guide the search for find path in 2D. *International Journal Rob. Res*., **13**(4), 315-325.

[24] Zeyad, A. A., Shahrizal, S. & Hoshang, K. (2015). A Comprehensive Study on Pathfinding Techniques for Robotics and Video Games. *International Journal of Computer Games Technology*, **2015,** Article ID 736138

[25] Zhu, Y., Zhang, T., Song, J. & Li, X. (2012). A New Hybrid Navigation Algorithm for Mobile Robots in Environments with Incomplete Knowledge. *Knowledge- Based Systems*, **27,** 302-313.

[26] Zhu, Y., Zhang, T., Song, J. & Li, X. (2012). A New Bug-type Navigation Algorithm Considering Practical Implementation Issues for Mobile Robots. *IEEE International Conference on Robotics and Biomimetics.* 531-536.

[27] Zohaib, M., Pasha, M., Nadeem, J. & Jamshed, I. (2012). Intelligent Bug Algorithm (IBA): A Novel Strategy to Navigate Mobile Robots Autonomously. *Third International Multi-topic Conference.* 18-20 December, Jamshoro, Pakistan.

[28] Zohaib, M., Pasha, M., Riaz, A., Javaid, N., Ilahi, M. & Khan, R. D. (2013). Control Strategies for Mobile Robot with Obstacle Avoidance. *Journal of Basic and Applied Scientific Research*, **3**(4), 1027-1036.

[29] Zohaib, M., Pasha, M., Javaid, N., Salaam, A. & Iqbal, J. (2014). An Improved Algorithm for Collision Avoidance in Environments Having U and H Shaped Obstacles. *Studies in Informatics and Control Journal*, **23**(1), 97-106.