*Research Journal of Recent Sciences* ISSN 2277-2502
Vol. **3(5),** 110-115, May **(2014)** *Res.J.Recent Sci.*

***Review Paper***

# Two New Methods for Path Planning of Autonomous Mobile Robot

**Mohsen AhmadiMousavi[1], BehzadMoshiri[2], Mohammad Dehghani[3] and HabibYajam[4]**
[1]Department of New Sciences and Technologies, University of Tehran, Tehran, IRAN
[2]Department of Computer and Electrical Engineering, University of Tehran, Tehran, IRAN
[3]Department of Electrical Engineering, Amirkabir University of Technology, Tehran, IRAN
[4]Department of Electrical Engineering, Sharif University of Technology, Tehran, IRAN

**Available online at: www.isca.in, www.isca.me**
Received 30th August 2013, revised 17th October 2013, accepted 16th December 2013

## Abstract

*This paper investigates four methods for finding shortest path between source and destination in a specific environment. For the known gradient method, we have propped a way to reduce the computational complexity of gradient field. Besides, the proposed method attempts to find the optimal path starting from a suboptimal path with the lowest computations. The considered robot is a mobile robot with three freedom degrees in two-dimensional environment. This will cause the isolation of the angle of trajectory path. The result of the simulations of the methods shows that the new approach provides an appropriate method for mobile robot routing in comparison to other methods.*

**Keywords:** Path planning in image space, mobile robot, computational complexity.

## Introduction

Path Planning is finding an appropriate path for robot movement in an obstacle-free space from the source to the destination. Obstacle-free space is the space of the robot movement which is not located in the configuration space. In this paper, the optimal path is the path which is between the source and destination and has the smallest distance among all possible paths[1]. In some cases having a smoother path with less acute angles is one of affecting parameters in path planning[2]. In this paper, we consider path planning for a robot with three freedom degrees in a two dimensional space. Our main issue is to find the movement path for a robot in a known environment. A machine vision system is assumed over the head of the robot which overlooks the fixed and moving obstacles. This machine vision system detects fixed and moving obstacles in the environment and provides the information in form of a sequence of images. So far, several methods have been proposed for routing in an unknown environment. The key techniques in this field work based on the visibility graph which is obtained from output of robot's range finder sensor[3]. Generalized A* algorithm in order to path plan based on distance map[4,5], and dynamic mapping of the environment and path planning on the map[6] are examples of this methods.

Optimal path planning is an important issue in mobile robot automatic navigation that leads to find the optimal path from the starting point to the destination without any collisions in the given environment with the aim of optimizing some certain criteria[7]. To overcome this problem, a number of path planning strategies proposed, some by using concepts of potential field[8], some based on visibility graphs[9] and some are image-based methods that are also known as grid methods[10]. The potential field method is simple and well-structured to avoid colliding with obstacles. But such methods have inherent limitations including trapping in local minima and among adjacent obstacles, swinging in the presence of adjacent obstacles and narrow passages, and high dependency to parameters[11]. Barraqu and used bit arrays to generate potential field in numerical form for path planning[12] which is a category of artificial potential field method. In this paper, a method for calculating gradient which results in appropriate artificial potential field is provided. The main disadvantage of the methods based on the visibility graph, is low efficiency[13]. To overcome the problem of complexity in these methods, Jiang used a visibility graph method without producing a complex space for robot movement, to find the shortest path[14]. The main problem in image-based methods is how to determine the image size. The larger the image size, the more accurate will be the environment's representation. However, using more pixels will exponentially increase the memory space and the search range[15]. Graph-based path planning techniques have been proposed by Latom be for solving this problem. Road map, cell decomposition, and potential field are examples of such methods. Planning problem is fully formulated and based on the formulation a robust technique for finding optimal path is proposed[16]. An efficient path planning approach which is for car-like robots and based on a recursive segmentation of free-collision path has been provided[17-19]. In Section 2 we will overview the path planning in the image space and the input image to the path planning section. Section 3 will introduce several methods for path planning. Section 4 presents a new approach to find the optimal path and Section 6 provides and compares the results of simulations.

## Path Planning in Image Space

In image-based path planning, the path consists of a sequence of neighboring pixels locations which its beginning is the robot's current location and its end is the robot’s final position. In addition, the sequence of robot’s angles in each pixel is one of the results of path planning. In path planning in image space, distance of a path can be regarded as number of pixels which are included in the path.

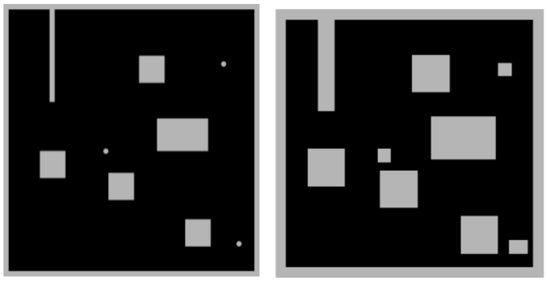
In this paper, it is assumed that a machine vision unit segregate fixed and mobile obstacles by using different colours. In each picture frame, a blue colour is used for fixed obstacles and moving obstacles are considered red.

Input of path planning unit is animage which is based on the current position and the final position of the robot and such a sequence of images should contain moving obstacles in their current position at the moment. First, it is necessary to integrate the image to reduce the complexity of the problem.

Given a sequence of images, using timing information we can calculate speed of the moving obstacles in the past moments, accordingly we can guess the next most probable places for moving obstacles. So the whole trajectory of the obstacles in a time interval can be obtained and we can replace the position of the moving obstacle with an appropriate fixed obstacle in each frame. This makes our path planning involved in a space containing only fixed obstacles. Another strategy for dealing with moving obstacles would be path planning considering only fixed obstacles while moving obstacles would be considered whenever they are approaching the robots’ location, and so the path planning should be done in a safe margin to avoid collision with moving obstacles. The recent method has advantages over previous methods, but the complexity and the workload would be increased.

The size of the robot is another issue that should be considered. Our aim is to avoid collisions for any parts of the robot and the obstacles.

Since any robot has certain size and its size of the robot makes the path planning more difficult. In a solution, the robot can be considered only a point, but proportional to the distance of the external surfaces of the robot from the centre of it, we can consider a margin around the obstacles. If the robot has a circular shape, we consider the distance equal to the radius of the robot, but in case of a different shape the farthest point of robot’s external surface to the centre of it should be considered, or it will be proportional to the angle of the robot with respect to the obstacle. And the later approach makes a lot of complexity. Whereas the path planning is often being done in real time and low computation cost is required, this method has not been used. Figure 1 is an example of an input image and a binary image with a safe margin.



(a) (b)

**Figure-1**
**(a) an example of obstacles in an Image, (b) Obstacles in binary shape**

## Path Planning Methods

Different methods can be used for path planning such as A* algorithm and heuristic algorithms. Following will be a brief reference to some of the methods and also explanations of the proposed methods. Since the robot has a circular shape and has four omni-directional drive wheels, the robot’s path will be completely independent of its angle. Under such circumstances, our main issue is the calculating the position of path. And the angle of the robot along the path will be calculated after the position of the path being obtained.

## Path Planning by Using Potential Field

One of the commonly used path planning methods is the gradient field method which uses gradient field as a guide for the robot to he path planning. In this method, the robot path would be in a direction which has the least repulsive field gradient and the highest attractive gradient field. Repulsive gradient should be inversely proportional to the distance to obstacles. In fact, this part of the field gradient, will keep away the path from the space around the obstacles. To guide the robot to the destination attractive gradient should be proportional to the distance to the target. There are different equations for calculating such fields. An equation for calculating square repulsion of each block is as follows[20]:

$$U_{rep,i}(q) = \begin{cases} \frac{1}{2}\eta_i\left(\frac{1}{\rho(o_i(q))} - \frac{1}{\rho_0}\right)^2 & ; \rho(o_i(q)) \le \rho_0 \\ 0 & ; \rho(o_i(q)) > \rho_0 \end{cases} \quad (1)$$

Where $\eta_i$ is a fixed amount for every obstacle and the distance to $i$ th obstacle is $\rho(o_i(q))$. In this equation, the field obeys the radius of influence showed by $\rho_0$ and the field can be a suitable value for a gradient field. On the other hand, different methods have been used for calculation of the attractive field. One of these methods uses an equation as follows:

$$U_{att,i}(q) = \begin{cases} \frac{1}{2}\zeta_i\left\|o_i(q) - o_i(q_f)\right\|^2 & : \left\|o_i(q) - o_i(q_f)\right\| \le d \\ d\zeta_i\left\|o_i(q) - o_i(q_f)\right\| - \frac{1}{2}\zeta_i d^2 & ; \left\|o_i(q) - o_i(q_f)\right\| > d \end{cases} \quad (2)$$

These definitions have useful properties and their performance has been investigated in different conditions[20].As noted in an image-based path planning the goal is to find a sequence of pixels which starts from the origin and reaches the destination. Accordingly, for each pixel of obstacle-free space the distance to each obstacle and the distance to the destination point can be calculated, and based on the obtained values repulsion field of each obstacle and the gravitational field can be calculated in any pixel. However, the calculations on this method can provide good answers, but the high amount of computational complexity of the method is not desirable. For example, for finding the repulsive field of a certain obstacle in a certain pixel first nearest point of the obstacle to the pixel should be determined, there for it requires detecting the edge of the obstacle then nearest point of each obstacle to the object should be determined. Subsequently based on the detected edges the repulsive field of that obstacle would be acquired, and these calculations should be repeated for each obstacle. Apparently, the calculations has high amount of computational complexity.

Due to this issue, a new method for calculating the gradient field will be provided in the following.

**The proposed image-based method for calculating gradient fields of obstacles:** Each binary image is a matrix with values 0 and 1.We assume that the areas of obstacles have been defined by the value“1”.We define the magnifier operator which changes each pixel’s with value “0” in the neighbourhood of a pixel with a value of “1”.This means that the outer edge of each obstacle should be added to the obstacle’s space or in other words, images with 1 pixel shifted in four directions (in case of quaternary neighbourhood) or eight directions (in case of octamerous neighbourhood), should combined into one image. This combination should be expressed by logical "or". In figure 2 the effect of the operator on an example image in cases of quaternary neighbourhood and octamerous neighbourhood is shown.

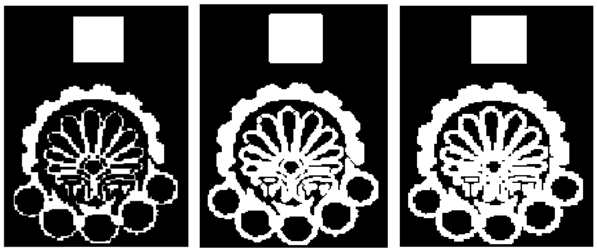

(a) (b) (c)

**Figure-2**
**The magnifier operator: (a) Original image, (b) Magnified with quaternary neighbourhood, (c) Magnified with octamerous neighbourhood**

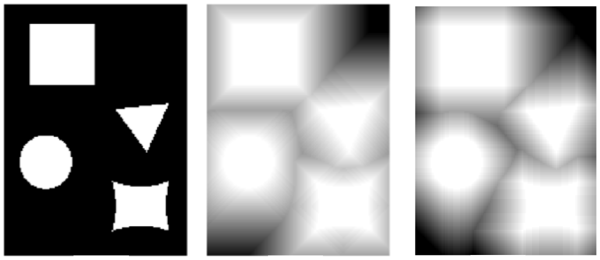

(a) (b) (c)

**Figure-3**
**Gradient field: (a) Binary image of obstacles, (b) Gradient field using magnifying quaternary neighbourhood operator, (c) Gradient field using magnifying with octamerous neighbourhood operator**

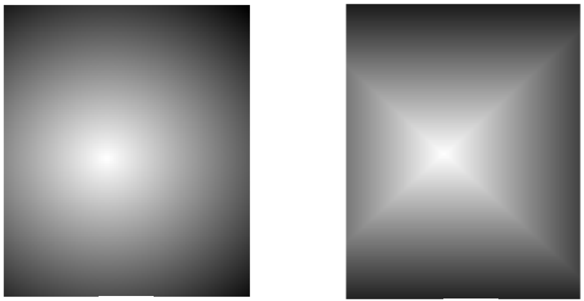

(a) (b)

**Figure-4**
**Gradient of Attractive fields: (a) Calculated inversely proportional to the distance, (b) Calculated by the proposed algorithm**

In the proposed method for calculating the gradient field of a binary shape a parameter ρ should be defined. This parameter acts the same as parameter $\rho_0$ in equation. (1) and makes any obstacle effective in a certain distance (ρ pixels) and ineffective beyond. Gradient field is being calculated by applying the magnifier operator ρ times on the image in both cases of quaternary and octamerous neighbourhoods. And each time, the resulting image should be combined by a factor of $1/i$ with the previous image. In each step, the factor makes the values of augmenting neighbours reduce as the distance to the obstacle increases. In figure 3the gradient field calculated using the proposed method is shown.

However, the attractive gradient field can be calculated in the same way but as the target location is known for many sequences of images and also the distance of each pixel to target location should be measured once, accurate calculation of the attractive gradient field would not have a high computational complexity. Subsequently, the gradient field can be easily calculated based on the distance and field. In figure 4, the field gradient calculated based on the distance and using the proposed method are compared with each other.

For path planning using the gradient field an image should be created which is based on the combination of repulsive and attractive fields according to the equation (3).

$$\hat{G} = \alpha \overline{G_{atr}} - \beta \sum_i \overline{G_{rep,i}} \qquad (3)$$

In the above equation, $\overline{G_{atr}}$ is the normalized attractive field and $\overline{G_{rep,i}}$ is the normalized repulsive field for the $i$th obstacle. Although in the above equation repulsive fields are combined together with an addition operator this increases the field unreasonably in the places where the obstacles' repulsive field overlap. To solve this problem we propose averaging the fields in places the overlapping of obstacles occurs. It should be noted that if the distances between the obstacles are more than a limit the overlapping would be negligible. In this case the proposed algorithm can be applied to all obstacles once and together.

In this method the coefficients α and β are supersensitive. G is a suitable combination of attractive but it can have value even in obstacle space. This might lead to malfunctions in path planning. To address this problem, first the attractive field should be normalized and then it should be set to zero on each obstacle's space.

For specifying the path we start from the robot's current position then in each pixel all eight neighbouring pixels should be examined and the pixel with highest combined gradient field value should be chosen for next pixel in path's sequence. This process should be repeated until reaching the end point. Figure 5 shows the combined gradient field and resulted path using this operator with coefficients α = 1 and β = 0.5.

## Corner Passing Method

This method is also based on a binary image which is created from obstacles. This method can easily be expressed as an algorithm. In first stage, from the starting point a straight line should be drawn toward the target. If there is an obstacle in the path two paths from the starting point would be considered to the outermost corner of the object and the same process for each path should be repeated until reaching the destination without any collisions with obstacles. The distance of each path would be computed and the path which has the shortest distance is the optimal path. This method requires a lot of computing because at each stage the outermost corners of the object should be calculated, but it gives the most optimal path. In figure 6, the obtained optimal path by this method is shown.

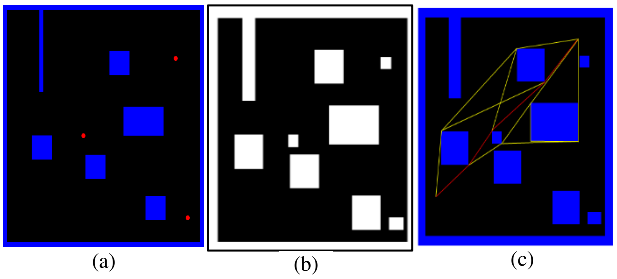


**Figure-6**
**Functioning of Corner Passing Method: (a) Image of Obstacles, (b) Binary shape of obstacles with safe margin, (c) Planned path using the method (the red path is the most optimal one)**

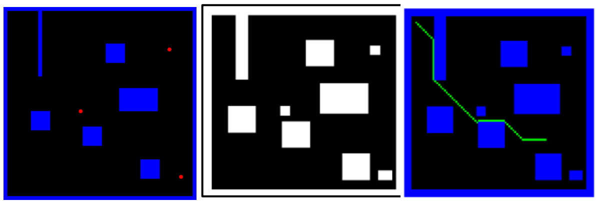


**Figure-7**
**Functioning of nearest neighbour method: (a) Image of Obstacles, (b) Binary shape of obstacles, (c) Planned path using the nearest neighbour method**

## Nearest Free Neighbour Path Planning

This method is also based on a binary image created from obstacles. For this method same as "corner passing method", we start with a binary image based on obstacle positions and sizes. And for keeping the robot away from obstacles a safe margin would be considered. Then for each pixel, the next place of robot is a neighbouring pixel with shortest distance to the target which is neither repeated in path sequence nor positioned in obstacle space. This method is very simple to implement and has very low computation complexity. However, the planned path my not be optimal and in some cases a path to the distension would not be found. In figure 7 an example of this path planning method is shown and compared.

A* algorithm can be used as an optimising algorithm for such a path planning method.

## Another Approach in Path Planning

Although, in most cases the main issue in a path planning approach is finding a suitable path to the destination. But another approach that could require less computational cost is optimizing a suboptimum path after finding one. A proposed path planning method will be explained in the following which is based on the later approach and has low computational complexity.

## Another Proposed Method for Path Planning

First, a low computational method such as nearest free neighbour method is applied for achieving a suboptimal path. In this method a constant $n$ would be used that should be specified before running the algorithm. In the first stage of algorithm (s = 1) the first pixel of the path would be considered as the beginning of a line segment and the next $n^{th}$ pixel of the path would be considered as the end of the segment line, if there is no collisions with any obstacle the line segment would be the replaced with the $n^{th}$ pixels of the path, and if there is at least a collision with obstacles the end of the segment line would be ($n$ – 1)th pixel of the path and if there is no collisions the segment path replaces the first n – 1th pixels of the path and so on till there is no collisions. In next stage we would start from the end of the last line segment and repeat the same calculations for finding a shorter path. This method has less computation than "Passing Corners" method while providing appropriate results. However, because of disability of the "nearest neighbour method" in the presence of concave obstacles, this method also does not work in such situations. In figure 8 an optimized path by the proposed method is shown.

"Via points" are the points which determine the end of each segment line. These points are not neighbours but they fully describe the whole path because they are the only points in which the path is not straight. These points can also be defined in other methods. In particular corners of the crossing point method and pixel coordinates of the other two methodsare other expressions ofvia points. Via points can fully expressthe output of the path planningunit.

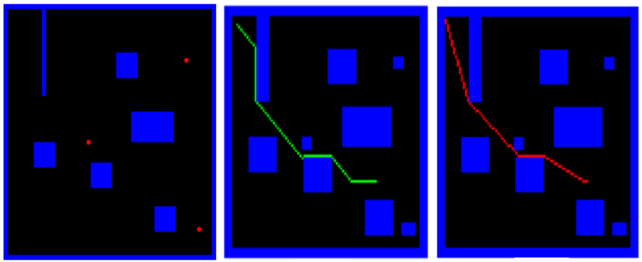

(a) (b) (c)

**Figure-8**
**Functioning of proposed path optimising: (a) Image of Obstacles, (b) Binary shape of obstacles and achieved path with nearest neighbour method, (c) Optimised path with proposed method**

## Robot's Angle of Trajectory

As mentioned, the robot's path is assumed to be completely independent of the angle of the robot because of circular shape of it and its four omni-directional drive wheels. Consequently, the angle of the robot can be calculated using equation (4).

$$\varphi_i = \varphi_{Str} + i\,\frac{\varphi_{End} - \varphi_{Str}}{N} \quad (4)$$

Where $N$ is the total number of crossing points and $\varphi_{str}$ and $\varphi_{End}$ are, respectively, the target angle and the angle of the robot's current position. And $\varphi_i$ is the angle of the robot in the $i$th via point.

It should be noted that for a robot without the appropriate degrees of freedom (three degrees of freedom) assumption of independency of position and angle of the robot would be completely unacceptable. Also, if the shape of robot is not circular, it is necessary to add the angle of the robot as a parameter in the path planning process for achieving the optimal path.

## Comparison of Described and Proposed Methods

In this paper 4 methods of path planning have been described but it can be shown that crossing point method always outputs the optimum path. On this basis, we define the following parameters for comparison of methods:

$$\text{Path Ratio} = \frac{\text{Path Distance}}{\text{Angel Method Distance}} \quad (5)$$

Where "Angel Method Distance" is the distance of found path by the angle method, and "Path Distance" is the distance of found path using the compared method. The less the parameter is the more optimised will be the method's found path. In Table 1 the results of 30 independent runs of the algorithm with different start and end points in a simulated environment with no concave obstacles are shown. Four compared methods simulated on a 3GHzdual-core processor using MATLAB® on Windows™ operating system.

**Table-1**
**Implementations results**

| Method | Path Ratio | Average Path Distance | CPU time (sec) |
|---|---|---|---|
| Proposed Method for Path Optimization | 1.830 | 42.444 | 1.601 |
| Nearest Free Neighbour | 3.921 | 90.932 | **1.043** |
| Corner Passing | **1** | **23.191** | 8.872 |
| Proposed Gradient Field | 8.653 | 200.678 | 1.272 |

As it can be seen in the table 1, the most optimal path can be found using passing corners method and the fastest method among all is the nearest free neighbour method. So in the case that online path planning is not considered the corner passing method would be the most desirable method. However, due to presence of moving obstacles, in most cases, online calculation of path planning is required and so the time of calculation is asubstantial parameter. The results of the gradient field method are not desirable. It should be mentioned that the adjustment of the parameters α and β in the gradient field method have been done using trial and error. Although it could be due to not correctly tuned parameters α and β, in some cases the final results obtained by the gradient field method are not by any means acceptable due to stocking in local optimums. In the proposed method, the distance of optimised path is favourable and the calculation shaved one in a reasonable time.

To have a fair comparison between our proposed algorithm and previous proposed algorithms in the literature, we should set all parameter equivalently. In this paper we have compared our proposed method with the Pamosoaji and Hong's method[21].

Table 2 shows a comparison between our proposed method and Pamosoaji and Hong's algorithm.

**Table-2**
**Implementations Results**

| Method | CPU time (sec) |
|---|---|
| Proposed Method for Path Optimization | 1.601 |
| Pamosoaji and Hong' s Method | **3.351** |

As it can be seen, one of the obvious advantage and superiority of our method is its speed. Although we should confess that in some cases this speed causes to less accuracy due to its low computations. But it is important to set a trade-off between the computation complexity and the algorithm's speed that we considered it in our proposed method.

## Conclusion

According to the results presented in table 1, it can be clearly conclude that the new proposed approach has better efficiency than other methods. Although the shortest path is obtained by corner crossing method, it has a high computation complexity. On the other hand, the computation cost of nearest free neighbour method is the lowest among all four methods, but the found path is not optimal. Gradient field-based method requires a lot of computations which can be reduced using the first proposed method for calculating the gradient field. However, due to the high dependency on the parameters α and β gradient field method's outputs are not acceptable in some cases.

## References

1. Jan K., Chang G.Y. and Parberry I., Optimal path planning for mobile robot navigation, *IEEE/ASME Trans. Mechatronics*, **(13)**, 451 **( 2008)**
2. Tae-Kyeong Lee, Sang-HoonBaeka, Young-Ho Choi and Se-Young Oh,Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation, *Robotics and Autonomous Systems*, **(59)**, 801–812 **(2011)**
3. Boult T.E., Dynamic digital distance maps in two dimensions, *IEEE Trans. Robot,* **(6)**, 590–597 **(1990)**
4. Subbarao K. and Larry S.D., Multiresolution path planning for mobile robots, *IEEE J. Robot*, **(2)**, 135–145 **(1986)**
5. Alexopoulos C. and Griffin P.M., Path planning for a mobile robot, *IEEE Trans. Syst., Man, Cybern,* (**22**), 318–322 **(1992)**
6. Zelinsky, A mobile robot exploration algorithm, *IEEE Trans. Robot*, **(8)**, 707–717 **(1992)**
7. Zheng T.G., Huan H, Aaron S, Ant Colony System ALgorithm for Real Time Globally Optimal Path Planing of Mobile Robots, *ACTA Automatic Sinica,* **(33)**, 279-285 **(2007)**
8. S.SGe, Y.J Cui, New potential functions for mobile robot path planning, *IEEE Transactions on Robotics and Automation*, **(16)**, 615-620 **(2000)**
9. Li L, Ye T, Tan M. Present state and future development of mobile robot technology research, *Conf.Robot*, 475-480, **(2002)**
10. V. Boschian, A. Pruski, Grid modeling of robot cells: a memory-effcient approach, *Journal of Intelligent and Robotic Systems*, **(8)**, 201-223 **(1993)**
11. Murphy R.R., Hughes K., Marzilli A. and Noll E., Integrating explicit path planning with reactive control of mobile robots using trulla, *Robotics and Autonomous Systems*, **(27)**, 225-245 **(1999)**
12. Barraquand J., Langlois B. and Latombe J.C., Numerical potential field techniques for robot path planning, *IEEE Trans. Syst., Man, Cybern.*, **(22)**, 224–241**(1992)**
13. Yang X., Moallem M. and Patel R.V., A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation,*IEEE Trans. Syst., Man, Cybern. B, Cybern.,* **(35)**, 1214–1224 **(2005)**
14. Jiang K., Seneviratne L.D. and Earles S.W. E., Shortest path based path planning algorithm for nonholonomic mobile robots, *J. Intell. Robot. Syst., Theory Appl.*, **(24)**, 347–366 **(1999)**
15. Gabriely Y. and Rimon E., Competitive On-line Coverage of Grid Environments by a Mobile Robot*, Computational Geometry*, **(24)**, 197-224 **(2003)**
16. Hu T.C., Kahng A.B. and Robins G., Optimal robust path planning in general environments, *IEEE Trans. Robot. Autom.*, **(9)**, 775–784 **(1993)**
17. Laumond J.P., Jacobs P.E., Taix M. and Murray R.M., A motion planner for nonholonomic mobile robots, *IEEE Trans. Robot. Autom.*, **(10)**, 577–593 **(1994)**
18. Sharifi M. and Shahriari B., Pareto Optimization of Vehicle Suspension Vibration for a Nonlinear Half-car Model Using a Multi-objective Genetic Algorithm, *Research Journal of Recent Sciences*, **1(8)**, 17-22 **(2012)**
19. Panah Amir, Enhanced SLAM for a Mobile Robot using Unscented Kalman Filter and Radial Basis Function Neural Network, *Research Journal of Recent Sciences*, **2(2)**, 69-75 **(2013)**
20. Mark W. Spong, S. Hutchinson, M. Vidyasagar, Robot Modeling and Control, *John Wiley and Sons*, Inc., **(2005)**
21. Pamosoaji A.K. and Hong K., A Path-Planning Algorithm Using Vector Potential Functions in Triangular Regions, *IEEE Trans. Systems, Man, and Cybernetics: Systems,* **(43)**, 832-842 **(2013)**