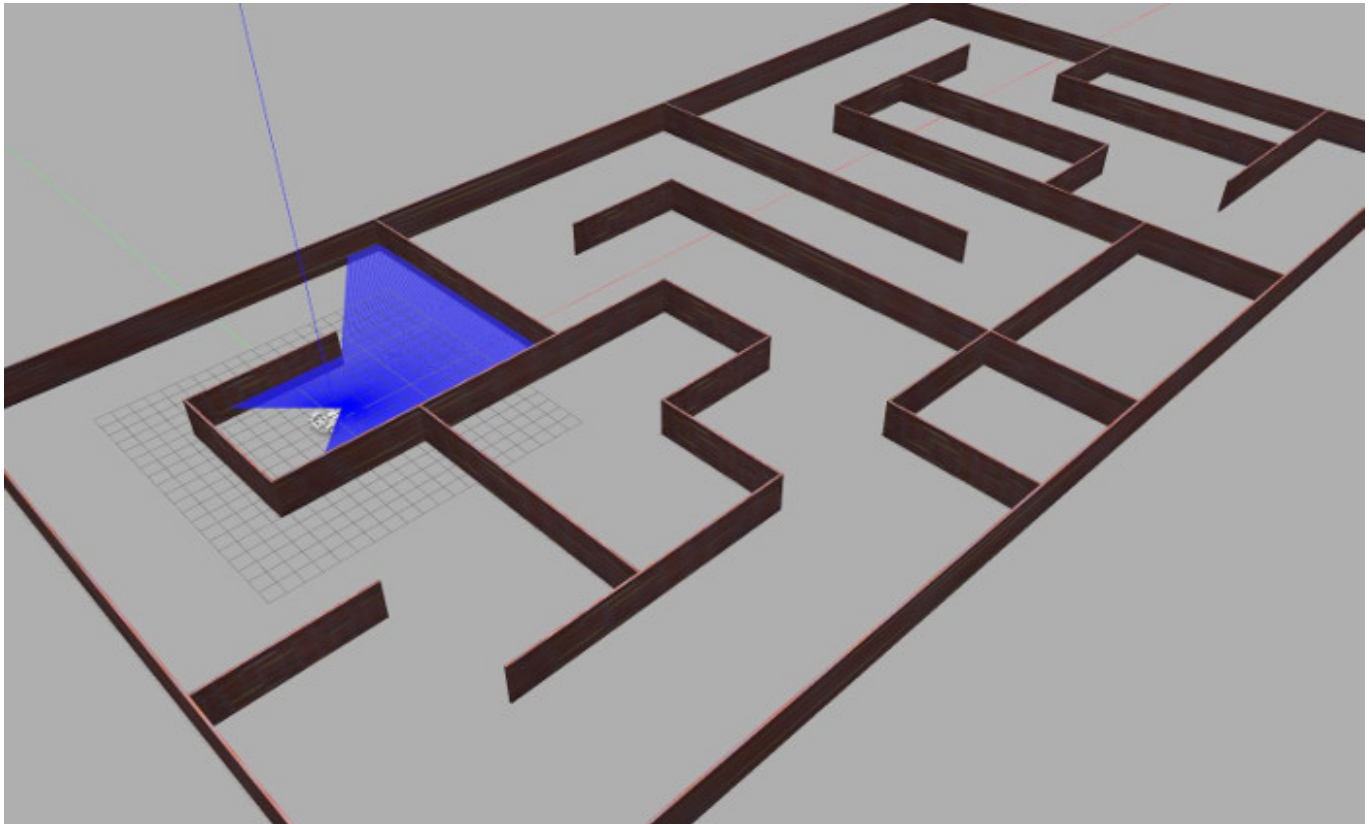


# Path Planning Using Potential Field Algorithm



Rymsha Siddiqui [Follow](#)

Jul 28, 2018 · 9 min read



## Abstract

Unmanned aerial vehicles (UAVs) have recently attracted the attention of researchers due to their various potential civilian applications. However, current robot navigation technologies need further development for efficient application in various scenarios. One key issue is the **Sense & Avoid** capability which is currently of immense interest to researchers. For autonomous decision making and control of UAVs, several path-planning and navigation algorithms have been proposed. The aim of this paper is to carry out a comprehensive study on UAV path-planning algorithms and how potential field algorithm works. The test case scenario presented in the experiment helps in

understanding how the time complexity varies with the dimensions and provided space of the system.

## Key Words

Path-planning, Robotics, Obstacle Detection, Potential Algorithm, A\* Algorithm, Algorithm Efficiency

## Introduction

Using robots to minimize human work has become a rising field of research in the past few years. From using mobile robots for military purposes on land as well as underwater to using them in restaurants for cooking food for customers, robotics has become an emerging field for human ease. However, one major problem while dealing with unmanned vehicles is path planning and obstacle detection while minimizing the input energy and getting the optimum results. Some recent literary works have been addressing this issue regarding how to use different techniques to create an obstacle distribution model and how we can get the desired results while making a few assumptions in a real-world system.

Potential Field Algorithm is one of the most fundamental and traditional algorithms used for such a problem. This paper will give a brief overview of how different algorithms can be used for path planning purposes and how the potential field algorithm helps us in Obstacle Detection in accordance to the literature sources used and the results achieved by the experiment conducted.

## A Comparative Study of Different Path-Planning Algorithms

Several scientists have been dealing with path planning optimization and obstacle detection problems in the recent past. A number of algorithms can be used and manipulated in several ways in order to use them for path-planning of UAVs. The following is an overview of the family of algorithms and their features:

### 1. Potential Field

- Difficult to implement for a real-world application

- Poor performance in narrow passages
- Poor performance in a dynamic environment
- Prone to get stuck in local minima situations
- Problems in dealing with the symmetrical obstacles

## **2. Floyd–Warshall**

- It is a subset of Dynamic Programming with optimal substructures
- It can lead to an optimal answer in a grid-based map
- Better performance compared to Dijkstra and Bellman-Ford algorithms[i]
- Capable of solving hard path-planning problem
- Due to high computational complexity, not a suggested method for on board implementation

## **3.Genetic Algorithm**

- Highly capable heuristic method.
- Time of computation and algorithm complexity is highly dependent on the algorithm implementation.
- The recent combination of this algorithm with the learning process resulted in a very promising online application[ii]

## **4. A\* Algorithm**

- It can provide a general heuristic approach for the search of an optimal path
- It can potentially search a huge area of the map
- The worst case for this algorithm results in considering all the nodes and edges in the graph

- The solution time would benefit from a static environment
- Based on the scenarios, may lead to optimal or near-optimal solutions

## 5. Dynamic Programming

- It solves a complex problem by breaking into a collection of sub problems
- It requires memories to save all the solutions generated from iterations
- Time of computation rises as the size of the problem grows
- The solution guarantees the optimal answer based on the implementation

## Terminologies to Be Used

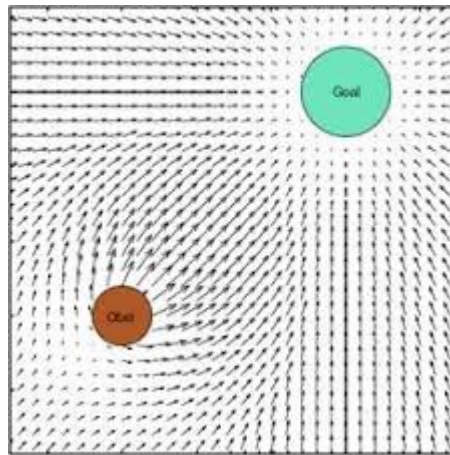
Potential field algorithm has been used for obstacle detection and path planning in robots traditionally. While studying the algorithm, a few terminologies must be defined to make the explanation straightforward. The following is a brief summary of the terms being used in the paper in order to explain the algorithm efficiently:

- **World space** *is a space where the vehicle exists.*
- **Configuration** *is a vector containing the parameters required for defining the positional state of the vehicle which usually consists of three position coordinates and three orientation coordinates of an UAV.*
- **Obstacle space** *is the physical space occupied by obstacles.*
- **Free space** *is the physical space which is not occupied by obstacles.*
- **Path** *is the curve, in a physical space, traced by the vehicle.*
- **Path planning** *comprises determination of a path from the present state called the initial state to the final state called the goal state.*
- **Local goal** *refers to the intermediate goal that the UAV tries to achieve in order to transition from the initial state to the goal state. This involves solving a number of sub-*

*problems to obtain the solution of the general overall problem. The overall objective is finding the path or trajectory for navigating the UAV to the global goal.*

## What is Potential Field and How Does it Work?

A potential field is any physical field that obeys Laplace's equation. Some common examples of potential fields include electrical, magnetic, and gravitational fields. A potential field algorithm uses the artificial potential field to regulate a robot around in a certain space. For our ease, we consider a space to be divided into a grid of cells with obstacles and a goal node.

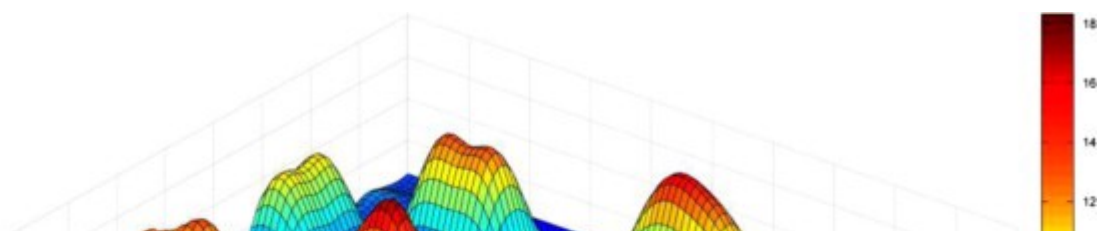


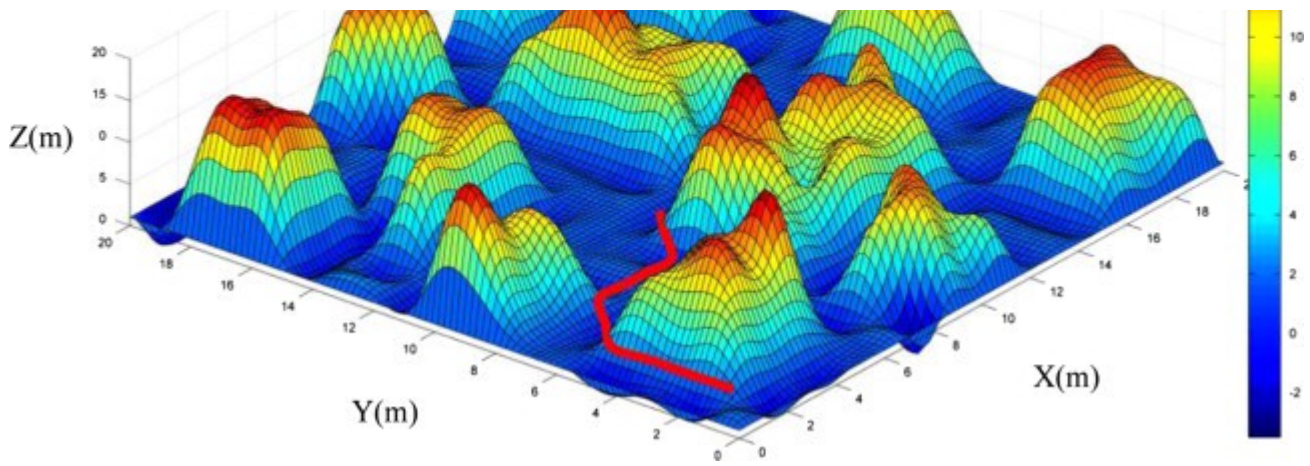
Potential Fields generated in a 2D Space

The algorithm assigns an artificial potential field to every point in the world using the potential field functions which will be described further in the explanation. The robot simulates from the highest potential to the lowest potential. Here, the goal node has the lowest potential while the starting node will have the maximum potential. Hence, we can say that the UAV moves from lowest to the highest potential.

## Potential Fields Generated:

Two kinds of artificial potential fields are generated within the system: Attractive field and Repulsive fields.





Obstacle Detection in a 3D Space

The goal node exhibits an attractive field while the obstacles in the system produce repulsive fields.

**Force of repulsion  $\propto 1/\text{Distance from the robot to obstacles}$ .**

Hence, the generated total force at each point of the graph is:

$$U_{\text{total}} = U_{\text{attraction}} + U_{\text{repulsion}}$$

## Force of Attraction:

The following function can be used for generating the force generated by the goal node. Here,  $x$  and  $y$  are the coordinates of the starting node,  $(x_{\text{goal}}, y_{\text{goal}})$  are the coordinates of the goal node and  $C$  is a constant.

$$P_g = C \sqrt{|x - x_{\text{goal}}|^2 + |y - y_{\text{goal}}|^2}$$

## Force of Repulsion

Two kinds of repulsive forces are produced within the system:

### 1. Repulsive forces by the boundaries:

This force remains uniform throughout the system and hence, doesn't affect the calculations; this force is useful in keeping the robot away from the boundaries. The following equation can be used to find the repulsive forces exhibited by the boundaries:

$$P_{HA} = \frac{1}{\delta + \sum_{i=1}^s (g_i + |g_i|)}$$

where  $g_i$  is a linear function that represents the boundary of the convex region,  $\delta$  is a constant number with a small value and  $s$  is the number of boundary face segments.

## 2. Repulsive forces by the obstacles:

The force of repulsion from obstacles can be calculated through the formula given below. Here,  $p_{max}$  is the highest potential,  $(x_0, y_0)$  are the coordinates of the center of an obstacle and  $l$  is the side length of the obstacle:

$$p_{ij} = \frac{p_{max}}{1 + g}$$

where

$$\begin{aligned} g(x, y) = & (x_0 - l/2 - x) + |x_0 - l/2 - x| \\ & + (x - x_0 - l/2 + 1) + |x - x_0 - l/2 + 1| \\ & + (y_0 - l/2 - y) + |y_0 - l/2 - y| \\ & + (y - y_0 - l/2 + 1) + |y - y_0 - l/2 + 1|. \end{aligned}$$

Hence, the resultant force on the environment is:

$$P = P_0 + P_{goal}$$

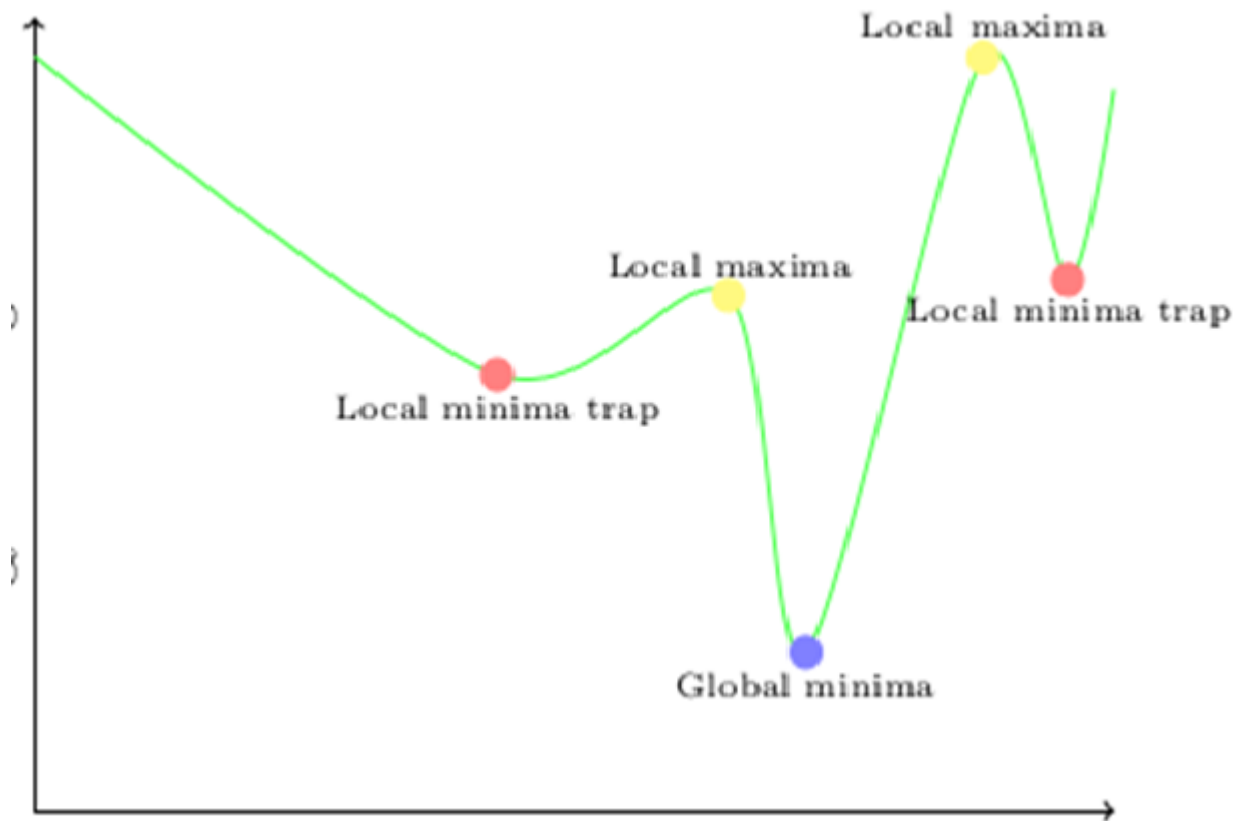
Where,

$$P_0 = \max\{p_i\}$$

## Local Minima Trap

A challenge for potential field algorithm is the local minima trap issue. This occurs when all artificial forces (attractive and repelling) cancel each other out such as in situations when an obstacle is located exactly between the UAV and the goal or when obstacles are closely spaced. Several approaches can be used to overcome this issue[iii]. The solution

approach that we use here adds an imaginary obstacle in the local minima region to repel the traveling object.



Global and Local Minima in a system

## Pseudo Code for the Algorithm:

```

 $t = 0, x_c(0) = x_{\text{start}}, \text{Flag} = 0$ , Calculate the potential function
while Next decision is not goal do
  if  $\text{Flag} = 0$  then
    Go to the next position with lowest potential,
  else if  $\text{Flag} = 1$  then
    Change the cell weight and treat the cells as occupied by an obstacle
    Update the potential of each cell,
  end
end
if the UAV is trapped in a cell or visit the same cells multiple times then
   $\text{Flag} = 1$ ,
  Search for the trapping point/points,
end
if the UAV flees from the local minima then
   $\text{Flag} = 0$ ,
end

```



```

|  $x_{t+1} \leftarrow$  cell nearby with lowest potential function.
|  $t \leftarrow t + 1$ 
end

```

Potential field algorithm for UAV-path planning in tessellated area.

## Time Complexity of the Algorithm

Potential field algorithms require evaluating forces in the configuration space and the complexity of these algorithms can often be  $O(M^D)$  where  $M$  is the total number of nodes in the space of computation and  $D$  is the dimension of the space. Successful application of this algorithm can be found in robot path-planning.

## Results & Analysis of Algorithm

Consider the grid below with obstacles(yellow colored cells) and the goal node(blue colored cell):

5		1		Goal
4				
3		2		3
2				
1				
0				
0	1	2	3	4

Grid with obstacles and empty spaces

Calculating the forces for repulsion by boundaries using the given formula:

Repulsive Forces by Boundaries	
$g_i = x - v$	$P_{HA} = 1/\delta + \sum (g_i +  g_i )$

Convex Region	P(0,5)	P(4,5)	P(4,0)
P <sub>HA</sub>	0.25	1.25	0.3125

Where:

$\delta$	$l$	$s$
0.2	3	4

The calculated forces of repulsion by the 3 obstacles are:

	Repulsive Forces from Obstacles		
	Pi=Pmax /1+g		
pmax is the goal potential			
	Obstacle 1	Obstacle 2	Obstacle 3
(x0,y0)	(1.5,4.5)	(1.5,2.5)	(3.5,2.5)
g(0.5,0.5)	5	1	3
g(1.5,1.5)	3	0	1
g(2.5,2.5)	1	1	0
g(3.5,3.5)	3	3	1
g(0.5,1.5)	3	0	3
g(0.5,2.5)	0	0	0
g(0.5,3.5)	0	1	3
g(0.5,4.5)	0	3	3
g(1.5,1.5)	3	0	1
g(1.5,2.5)	1	0	1
g(1.5,3.5)	0	1	1
g(1.5,4.5)	5	3	3

$g(1.5,0.5)$	0	1	1
$g(2.5,0.5)$	1	1	1
$g(2.5,1.5)$	1	1	0
$g(2.5,3.5)$	1	0	1
$g(3.5,0.5)$	1	0	0
$g(3.5,1.5)$	3	1	0
$g(3.5,2.5)$	1	0	0
$g(3.5,4.5)$	1	1	1
$g(2.5,4.5)$	0	1	1

As shown in the picture above, we calculate  $g(x,y)$  for the center co-ordinates of each cell in the world.

Hence the potential generated by each obstacle  $P_i$  is given by:

Obstacle 1	Obstacle 2	Obstacle 3	
$P_i$	$P_i$	$P_i$	$P_0$
5.64	1.64	3.64	5.64
3.64	0.64	1.64	3.64
1.64	1.64	0.64	1.64
3.64	3.64	1.64	3.64
3.64	0.64	3.64	3.64
0.64	0.64	0.64	0.64
0.64	1.64	3.64	3.64
0.64	3.64	3.64	3.64
3.64	0.64	1.64	3.64
1.64	0.64	1.64	1.64
0.64	1.64	1.64	1.64
5.64	3.64	3.64	5.64
0.64	1.64	1.64	1.64
1.64	1.64	1.64	1.64
1.64	1.64	0.64	1.64
1.64	0.64	1.64	1.64
1.64	0.64	0.64	1.64
3.64	1.64	0.64	3.64
1.64	0.64	0.64	1.64

1.64	1.64	1.64	1.64
0.64	1.64	1.64	1.64

The resultant force calculated by adding the attractive and repulsive forces is:

Resultant Force on environment: $P = P_0 + P_g$ $P_0 = \max\{p_i\}$			
x	y	$P_g$	$P = P_0 + P_g$
0.5	0.5	5	10.64
1.5	1.5	3.605551	7.245551
2.5	2.5	2.236068	3.876068
3.5	3.5	1	4.64
0.5	1.5	4.242641	7.882641
0.5	2.5	3.605551	4.245551
0.5	3.5	3.162278	6.802278
0.5	4.5	3	6.64
1.5	1.5	3.605551	7.245551
1.5	2.5	2.828427	4.468427
1.5	3.5	2.236068	3.876068
1.5	4.5	2	7.64
1.5	0.5	4.472136	6.112136
2.5	0.5	4.123106	5.763106
2.5	1.5	3.162278	4.802278
2.5	3.5	1.414214	3.054214
3.5	0.5	4	5.64
3.5	1.5	3	6.64
3.5	2.5	2	3.64
3.5	4.5	0	1.64
2.5	4.5	1	2.64

Hence the potential at each cell in the world is:

Potentials at each cell			
6.6	7.6	2.6	1.64

6.8	3.87	3.05	4.6
4.2	4.4	3.8	3.64
7.8	7.2	4.8	6.64
10.6	6.11	5.7	5.64

The arrows in the picture above represent the path followed by the robot. Here, we notice that the resultant path followed is the shortest possible path while ignoring the obstacles in the path. The time taken for solving the grid above is  $O(M^2)$ .

Through the process of tessellation, the entire area is divided into cells. The only points considered in path planning calculations are the centers of each cell. The UAV is constrained to travel only from the center of one cell to the center of cells connected to the UAV's currently occupied cell. Virtual objects are used to avoid local minima traps in the domain.

## Conclusion

From the simulation study of the potential field algorithm, we found that this algorithm has a reasonable time solution, but it shows poor ability to overcome the local minima trap and provides non-optimal results. Different scenarios may yield different results as the algorithm is difficult to implement in real world scenarios. It is clear from the results that there is a trade-off between the optimality and computational time requirements. Choice of the algorithm in practice needs to be based on operational requirements in terms of computational time and optimality.

## References:

[i] B. M. Sathyaraj, L. C. Jain, A. Finn and S. Drake, Multiple UAVs pathplanning algorithms: A comparative study, Fuzzy Optimization and Decision Making 7(3) (2008) 257–267.

[ii] N. Ernest, D. Carroll, C. Schumacher, M. Clark, K. Cohen and G. Lee, Genetic fuzzy based artificial intelligence for unmanned combat aerial vehicle control in simulated air

combat missions, J. Def. Manag. 6(144) (2016) 2167–0374

[iii]N. Ahuja and J.-H. Chuang, Shape representation using a generalized potential field model, IEEE Trans. Pattern Anal. Mach. Intelli. 19(2) (1997) 169–176

## Bibliography

Y. K. Hwang and N. Ahuja, A potential field approach to path planning, IEEE Trans. Robot. Autom. 8(1) (1992) 23–32.

Y. Koren and J. Borenstein, Potential field methods and their inherent limitations for mobile robot navigation, in Robotics and Automation, 1991. Proc, 1991 IEEE Int. Conf. (IEEE, 1991), pp. 1398–1404.

Radmanesh, Mohammadreza & Kumar, Manish & H. Guentert, Paul & Sarim, Mohammad. (2018). Overview of Path Planning and Obstacle Avoidance Algorithms for UAVs: A Comparative Study. Unmanned Systems. 6. 1–24

Y. Koren and J. Borenstein, Potential field methods and their inherent limitations for mobile robot navigation, in Robotics and Automation, 1991. Proc, 1991 IEEE Int. Conf. (IEEE, 1991), pp. 1398– 1404. M. G. Park and M. C. Lee, A new technique to escape local minimum in artificial potential field based path planning, KSME Int. J. 17(12) (2003) 1876–1885.

G. Faria, R. A. F. Romero, E. Prestes and M. A. P. Idiart, Comparing harmonic functions and potential fields in the trajectory control of mobile robots, in Robotics, Automation and Mechatronics, 2004 IEEE Conf., Vol. 2 (IEEE, 2004), pp. 762–767

[Programming](#) [Path Planning](#) [Potential Field](#) [Robotics](#) [Obstacle Avoidance](#)

[About](#) [Help](#) [Legal](#)