# Code for Robot Path Planning using Artificial Potential Fields

Rahul Kala
Robotics and Artificial Intelligence Laboratory,
Indian Institute of Information Technology, Allahabad
Devghat, Jhalwa, Allahabad, INDIA

Email: rkala001@gmail.com
Ph: +91-8174967783
Web: http://rkala.in/

*Version 1, Released: 7th June, 2014*

## 1. Background

The code provided with this document uses Artificial Potential Fields for robot motion planning. Assume that you have a robot arena with an overhead camera as shown in Figure 1. The camera can be easily calibrated and the image coming from the camera can be used to create a robot map, as shown in the same figure. This is a simplistic implementation of the real life scenarios where multiple cameras are used to capture different parts of the entire workspace, and their outputs are fused to create an overall map used by the motion planning algorithms. This tutorial would assume that such a map already exists and is given as an input to the map.

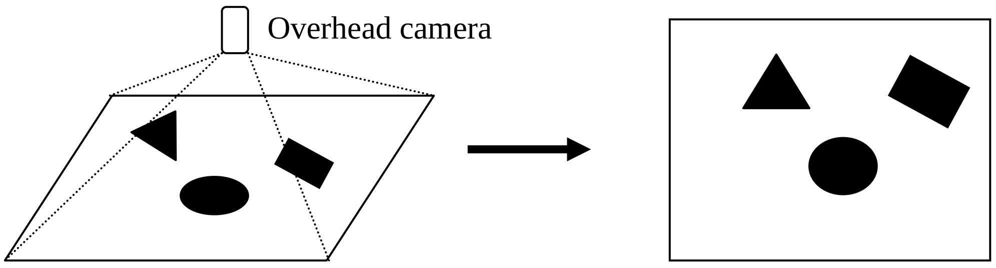


**Figure 1: Overhead camera system and creation of robot map**

The same camera can also be used to capture the location of the robot at the start of the planning and also as the robot moves. This solves the problem of localization. An interesting looking region of interest becomes the goal of the robot to be used in the motion planning algorithms. The robot is not shown in the map in Figure 1. This tutorial assumes that the source and goal of the robot is explicitly supplied.

## 2. Problem Solving using Artificial Potential Fields

The readers should please familiarize themselves with Artificial Potential Fields before reading this tutorial. Artificial Potential Field based navigation is a reactive planning technique, where the immediate distances from obstacles are considered to compute the immediate move, without much bothering about the future. In such a manner immediate actions lead to motion of the robot, ultimately leading to the goal. All obstacles repel the robot with a magnitude inversely proportional to the distance. The goal attracts the robot. The resultant potential, accounting for the attractive and repulsive components is measured and used to move the robot. The potential field for a sample scenario is shown in Figure 2. Directions indicate the direction of the potential vector.

The distance of the obstacles at all angles from the robot is measured. In this tutorial only 5 distances at specific angles are measured to compute the repulsive potential. These are forward, left side, right side, forward left diagonal and forward right diagonal. The different inputs are summarized in Figure 3.

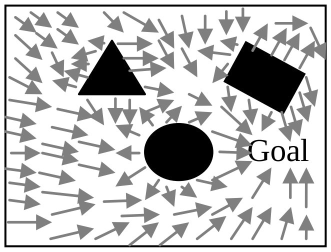


**Figure 2: Artificial Potential Fields**

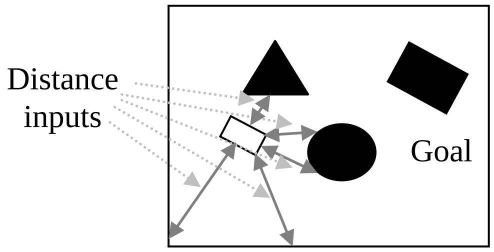


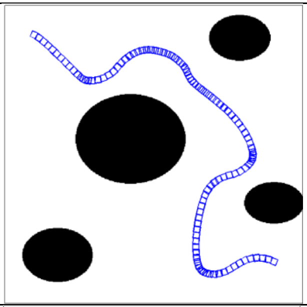
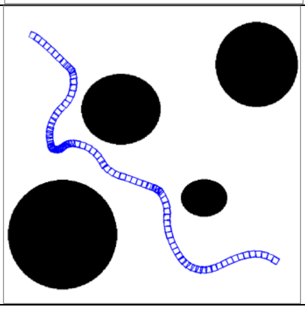
**Figure 3: Measurement of repulsive potential**

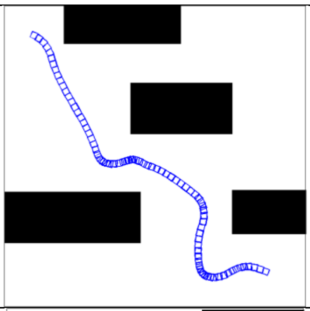
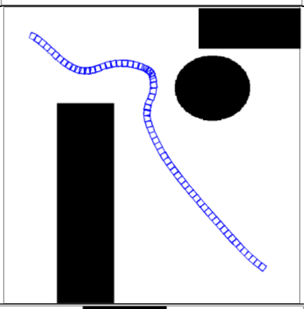
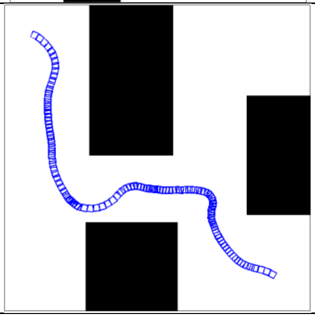
## 3. How to Execute and Parameters

In order to execute the code you need to execute the file 'astart.m'. First make a new bitmap file and draw any map on it, over which you need to execute the algorithm. Paint or any other simple drawing tool can be used. Make sure you save the file as a BMP. Place the file in the project folder. You may prefer to open any of the supplied maps and re-draw them. Change the name of the map file in the code to point out to the map that you created. Supply the source and the goal positions and the initial heading direction (in radians). You can use paint or any other drawing utility which displays the pixel positions of the points, to locate the source and goal on your drawn map. Robot static and kinematic parameters may be changed to suit the requirements. However this will change the ideal fuzzy inference system. Changes to the rules and the membership functions can be carried at the fuzzy editor.

Execute the algorithm. You will need to take screenshots of the display, the tool does not do that for you. The simulation will stop when the robot reaches its goal, collides or is forced to stop as per the safety precautions. The path length and execution time are given at the console.

## 3. Sample Results

| S. No. | Path | Path Length | Execution Time (sec) |
|---|---|---|---|
| 1. |  | 839 | 2.84 |
| 2. |  | 785 | 2.68 |

| | | | |
|---|---|---|---|
| 3. |  | 775 | 2.15 |
| 4. |  | 775 | 1.85 |
| 5. |  | 779 | 2.94 |

All results on Intel i7, 3.4 GHz 3.4 GHz with 12 GB RAM.
For all results:
source=[50 50]
goal=[450 450]
resolution of original map: 500×500

## 4. Disclaimer

Please feel free to ask questions and extended explanations by contacting the author at the address above. Please also report any errors, corrections or improvements.

These codes do not necessarily map to any paper by the author or its part. The codes are usually only for reading and preliminary understanding of the topics. Neither do these represent any state-of-the-art research nor any sophisticated research. Neither the author, nor the publisher or any other affiliated bodies guarantee the correctness or validity of the codes.