University of Girona

Spain

Probabilistic Robotics

Lab 3 - Particle Filter

Submitted by :
Mr. Nayee Muddin Khan
DOUSAI

1 Introduction

The Lab 3 is based on particle filter, this is basically used for localization of robotics, which is also known as Monte Carlo localization. As we know, it is successfully applied to low-dimensional estimation problems, for example in environments with known maps. This report focuses on the implementation methods for particle filter. In this lab work, we will describe about the way of implementing methods,observations,results and the problems faced.

2 Implementation

The implementation is divided into three parts: Prediction, Weighting and resampling. Prediction was the section which has implemented as pre-lab exercise. Weighting and resampling was implemented in the labs. The algorithm uses a bunch of particles to represent the distribution of likely states, with each particle representing a possible state, i.e. a hypothesis of the location of the robot. The steps are followed as:

2.1 Prediction

For every iteration, we first predict the position p_xy and angle p_ang of every particle which are located in world frame. What we do is to move particles based on an odometry difference between two consecutive measurements in the vehicle frame. For the angle, we can simply add angle to the current p_ang because the change of angles is the same in both vehicle and world frame.
However, as for the position, if we just directly add x and y directions to p_xy, all the particle will just move to the right. Because the directions is the movement under the vehicle frame and the robot only moves in its x direction in the frame, all the particle will only move to the right, which is the x direction in the world coordinate. We should transform the directions to the world coordinate according to the p_ang before we add it to p_xy. This was the first problem we faced. Meanwhile, in order to simulate the uncertainty of the prediction, some Gaussian noise should be added to p_xy and p_ang. The final correct prediction result is shown below figure:

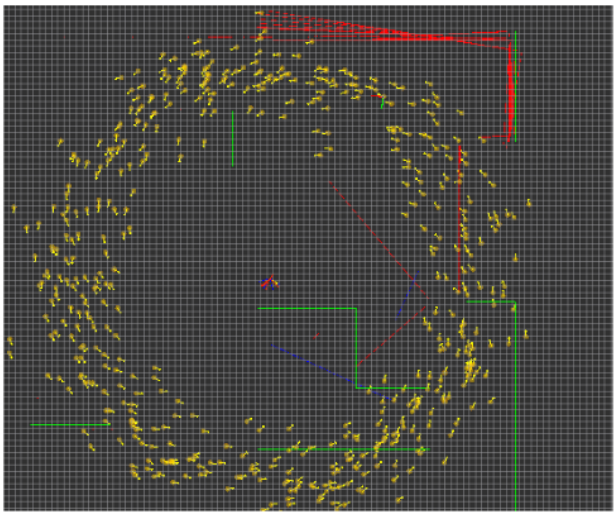

Figure 1: Prediction of Particles

2.2 Weighting

After predicting the position and angle of every particle, we need to calculate the weights for them. If the particle is close to the real odometry of the robot, high weight should be assigned and vice versa. The first step of our implementation is using function get_polar_line to acquire the range and angle of lines obtained from the measurements and all the lines in given map. It should be noted that, for the sake of comparison, we should transform the given map lines from world frame to the vehicle frame because the measured lines are already in the vehicle frame.

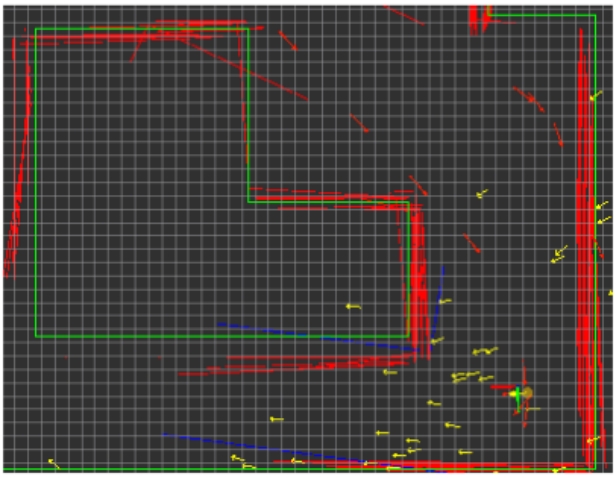

Figure 2: Weighting of Particles

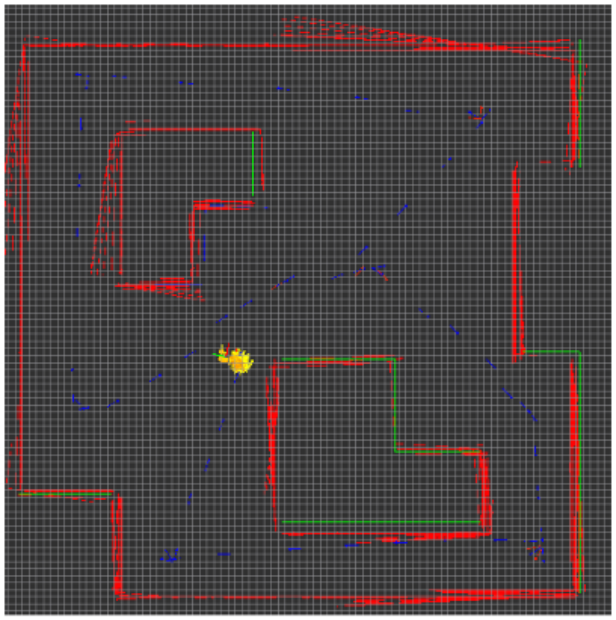

Figure 3: Resampling of Particles

2.3 Resampling

In weighting part, we found out that only one particle has a large weight. How-ever, what we want is to refocus the particle set to regions in state space with high weights. The more particles in a position, the more likely the robot is there. This is the resampling.