Comp 417 Assignment2 Report

Zhaoqi Xu

For this assignment, I am using particle filter to implement the robot localization.
I used an array of structure to represent particles, and I saved the x, y coordinates as well as the weight of each particle in a single structure(code line: 39). I set up this array as a global variable so that I can initialize it(line:45) and update it(line:58) through the entire process.
For initialization of these particles, since it is assumed that the starting point is known, I simply initialize the particles with a normal distribution in a small region(line: 50).

Motion model:
Since I am using particles to represent the belief, I have to update particles simultaneously as the robot moving. So the motion model should be in motionCommandCallback function(line: 183) and the motion of all particles are updated at line: 198.

Sensor model:
After we updated the motion model of particles, robot is getting new information from environment through camera, and we need to update the weight of each particle according to the difference of the pixel value(line: 98) and normalize the weight according to the value(line: 159). Line 158 is getting pixel value from particles. Then re-sampled particles(line: 161). The generate_particles() function actually re-sampled the particles and drew them out. The algorithm that I am using for re-sampling is from here (https://www.udacity.com/course/intro-to-artificial-intelligence—cs271). Firstly, randomly choose a index of particles and a weight from 0 to 2*max_weight(line: 66). Then just go over the whole particle list and decide which one to be deleted and which one to be copied more(line: 72) according to the weight you randomly got and from particle in index position. (Details are at 1. Normalizing weights: https://www.youtube.com/watch?v=zlCJQmxvrkE 2. Algorithm for re-sampling: https://www.youtube.com/watch?v=wNQVo6uOgYA) After this, I just draw out the re-sampled particles.

Note that, to make my life easier, I just compare only one pixel from the robot image to other particles. Therefore, my code works well(always converge to the actual position) for the non-kidnapped case but may not converge in the kidnapped case.