Introduction to Game Programming and Robotics

Unit # 17

Acknowledgement

- Most of the examples/material presented in these slides are taken from an online course on "Programming a Robotic Car" hosted at UdaCity.com and taught by Sebastian Thrun and Gundega Dekena.

Landmark Positions for RoboCup Competition

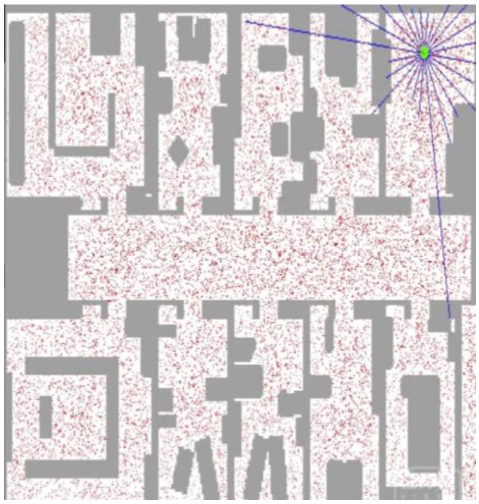
- GoalLeftTop: (-5, 1.05)
- GoalLeftBottom: (-5, -1.05)
- FlagLeftTop: (-5, 3)
- FlagLeftBottom: (-5, -3)
- GoalRightTop: (5, 1.05)
- GoalRightBottom: (5, -1.05)
- FlagRightTop: (5, 3)
- FlagRightBottom: (5, -3)

Particle Filters

- Particle filters comprise a broad family of Monte Carlo algorithms for approximate inference.
- In robotics, early successes of particle filter implementations can be found in the area of robot localization, in which a robot's pose has to be recovered from sensor data.
- The key advantage of particle filters is that they are really easy to program.

Localization via Particle Filter

- Here is a floor plan of an environment where a robot is located and the robot has to perform global localization.
- Global localization is when an object has no idea where it is in space and has to find out where it is just based on sensory measurements.



Localization via Particle Filter (Cont'd)

- The robot, which is located in the upper right hand corner of the environment, has range sensors that are represented by the blue stripes.
- The sensors use sonar sensors, which means sound, to range the distance of nearby obstacles.
- These sensors help the robot determine a good posterior distribution as to where it is.
- What the robot doesn't know is that it is starting in the middle of a corridor and completely uncertain as to where it is.

Localization via Particle Filter (Cont'd)

- In this environment the red dots are particles.
- They are a discrete guess as to where the robot might be.
- These particles are structured as an x coordinate, a y coordinate and also a heading direction -- three values to comprise a single guess.
- However, a single guess is not a filter, but rather it is the set of several thousands of guesses that together generate an approximate representation for the posterior of the robot.

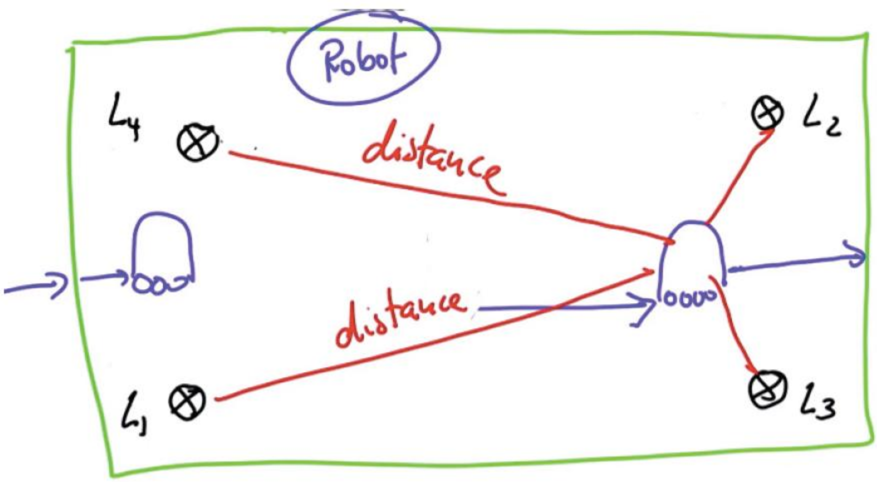
Survival of the Fittest

- The essence of particle filters is to have the particles guess where the robot might be moving, but also have them survive, a kind of "survival of the fittest," so that particles that are more consistent with the measurements, are more likely to survive.
- As a result, places of high probability will collect more particles, and therefore be more representative of the robot's posterior belief.
- The particle together, make up the approximate belief of the robot as it localizes itself.

Field with Four Landmarks

- Suppose you can program a move and it can sense the distance to four designated landmarks (L1, L2, L3, L4) .
- The distances from the landmarks to the robot make up the measurement vector of the robot.
- Since the robot lives in a world of 100x100, if it falls on one end, then it appears on the other -- it is a cyclic world.

Field with Four Landmarks (Cont'd)



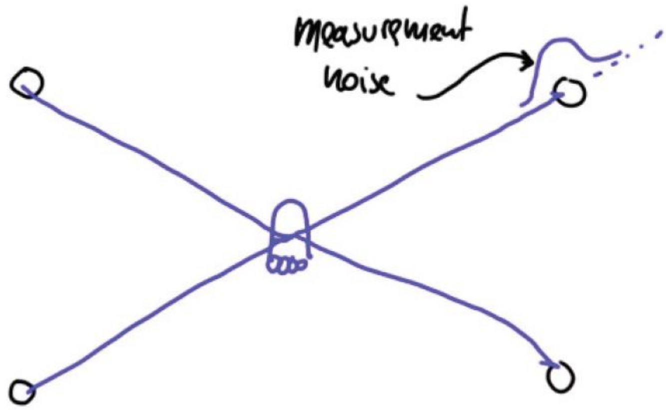
Initialization Particle Filters

- The particle filter you are going to program maintains a set of 1,000 random guesses as to where the robot might be, represented by a dot.
- Each dot is a vector that contains an x-coordinate
- (38.2), a y-coordinate (12.4), and heading direction (0.18).
- The heading direction is the angle the robot points relative to the x-axis; so as this robot moves forward it will move slightly upwards.

Second Half (Importance Weights)

- The second half of particle filters works like this, suppose you have a robot that sits amid four landmarks and can measure the exact distances to the landmarks.
- The image on the next slide shows the robot's location and the distances it measures, as well as "measurement noise," which is modeled as a Gaussian with a zero mean.
- This means there is a chance of the measurement being too short or too long, and that probability is governed by a Gaussian.

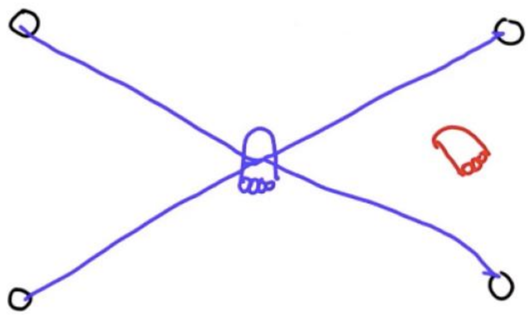
Measurement Noise



Assessing a Particle's Belief

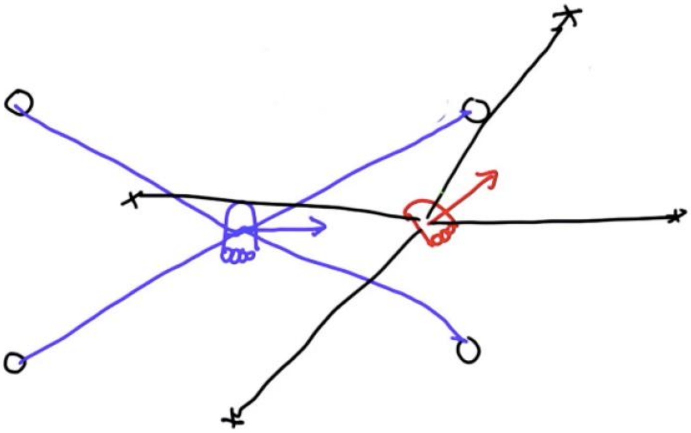
- Now we have a measurement vector that consists of the four values of the four distances from L1 to L4.
- If a particle hypothesises that its coordinates are somewhere other than where the robot actually is (the red robot indicates the particle hypothesis), we have the situation shown in the next slide.

Assessing a Particle's Belief (Cont'd)



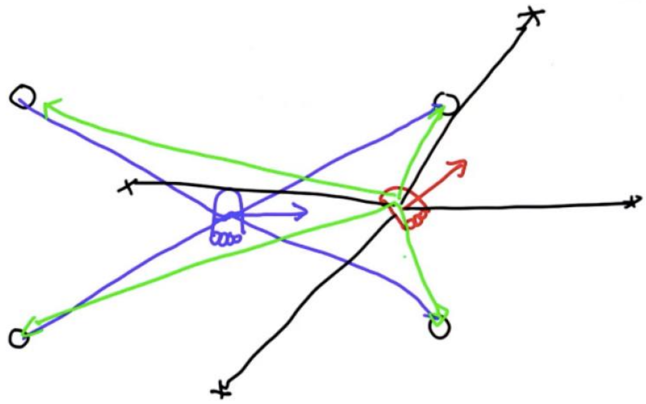
Combining Prior and Measurements

- The particle also hypothesises a different heading direction. You can then take the measurement vector from our robot and apply it to the particle.



Computing Error

- However, this ends up being a very poor measurement vector for the particle. The green indicates the measurement vector we would have predicted if the red particle actually were a good match for the robot's actual location.

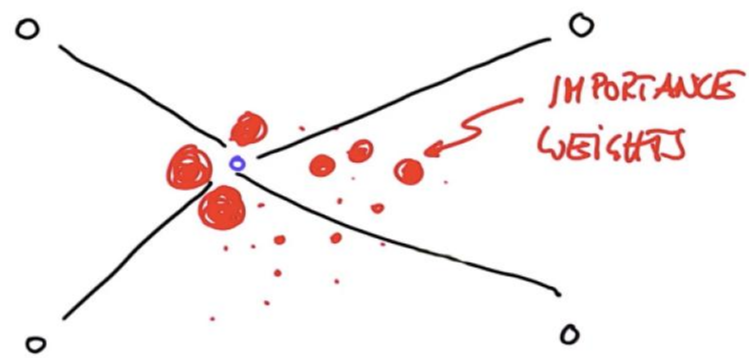


Computing Importance Weight

- The closer your particle is to the correct position, the more likely will be the set of measurements given that position.
- Here is the trick to particle filters; the mismatch of the actual measurement and the predicted measurement leads to an importance weight that tells you how important that specific particle is.
- The larger the weight the more important it is.

Importance Weights (Cont'd)

- When you have a bunch of particles, each has its own weight; some are very plausible, while others
- look very implausible as indicated by the size of the particle.



Survival of the Particles

- Next we allow the particles to survive at random, but the probability of survival will be proportional to the weights.
- That is, a particle with a larger weight will survive at a higher proportion than a particle with a small weight.
- This means that after resampling, which is randomly drawing new particles from the old ones with replacement in proportion to the importance weight, the particles with a higher importance weight will live on, while the smaller ones will die out.
- The "with replacement" aspect of this selection method is important because it allows us to choose the high probability particles multiple times.
- This causes the particles to cluster around regions with high posterior probability.

Resampling

- Resampling is when you are given N particles, each of which has three values and each one has a weight, w.
- The weights are continuous values and the sum of all of them is W .

$$W = \sum_i w_i$$

Normalize the weights:

$$\alpha_1 = \frac{w_1}{W}$$
$$\alpha_2 = ...$$
$$\alpha_N = ...$$

The sum of all alphas is:

$$\sum_i \alpha_i = 1$$

Resampling (Cont'd)

- Resampling puts all the particles and their normalized weights into a big bag.
- If there are N particles to begin with, you draw N times.
- In the end, those particles that have a high normalized weight α, will occur more frequently in the new set.
- This is resampling.

Quiz

- During the process of resampling, if you randomly draw a particle in accordance to the normalized
- importance weights, what is the probability of drawing p1 – p5.