Lab #2 – Localization using Particle Filters

Before we get starter, here's a brief reminder of the rules in effect during the lab session.

In-lab rules:

- ***No food or drinks are allowed inside the lab.*** Be sure to eat properly before your session,
- you can step out for water if you need to. Notify your TA since he/she is the only person that can grant you access to the lab.
- ***You can work on your laptop during the session if you wish,*** but please do not use any
- other electronic devices during the session.
- ***Keep the hallway free of bags and backpacks.***
- ***While in the lab, you're responsible for taking good care of all equipment***.
- ***Treat everyone else in the lab with respect and consideration***.
- ***You must listen and follow all instructions provided by your instructor and TA***.

How your work will be marked:

Your TA will observe your work during the work period.

- 15% for attending this lab and working hard.
- 85% for your solution to the programming exercise (due no later than the deadline)

You need to submit ***only one file per team***. We will thoroughly test your code and grade based on how well your solution works. We will also look at your coding style, and at whether your code has been produced with optimization in mind. Comment your code thoroughly and document what it does!

Note that any assigned marks for the programming part are contingent on each team member being able to explain how the code works and what it's doing.

The quiz will take place in a tutorial or lecture after the lab session has passed and will contain questions related to the concepts and material covered by this lab.

The point is – the goal of the lab is to help you fully understand the concepts covered in class. Teams should make sure everyone has a solid grasp of the ideas and principles involved in robot localization.

The goal of this lab session is to give you practical experience in implementing ***particle filters*** for robot localization. Particle filters are a common and very useful technique that allows a robot to determine a small number of unknown parameters from noisy sensor readings. In particular, position, heading direction, and velocity are typical quantities that can be estimated with reasonable accuracy via particle filters.

Learning Objectives – after completing this lab you should be able to:

Explain the concept of ***particle*** and how it it used within the context of robot localization.

Explain and implement the two steps involved in robot localization: sensing and acting.

Explain under what conditions we can expect particle filters to be efficient and effective at localizing a robot.

Understand a sensor noise model, and use it to properly evaluate the probability that a particle provides a good guess as to where the robot is.

Understand the trade-off between localization robustness, number of particles, and computational expense of the localization task.

Implement a fully working particle filter for localization, including every major step and component.

Skills Developed:

Working with noisy systems and randomized algorithms. Determining acceptable levels of sampling.

Ability to implement robot localization in a simple setting that nevertheless applies almost as-is to real-world robot-localization problems.

Thinking in terms of dynamics, prediction, and consistency of prediction with sensor readings as a source of information.

Reference material:

Your lecture notes on particle filters.

Sebastian Thrun's on-line short lectures on particle filtering (thanks Sebastian!). Start here:

http://www.youtube.com/watch?v=4S-sx5_cmLU

Then follow the trail to where it leads. Mind that Sebastian's robot setup and ours are somewhat different, so do not get confused with Python details!

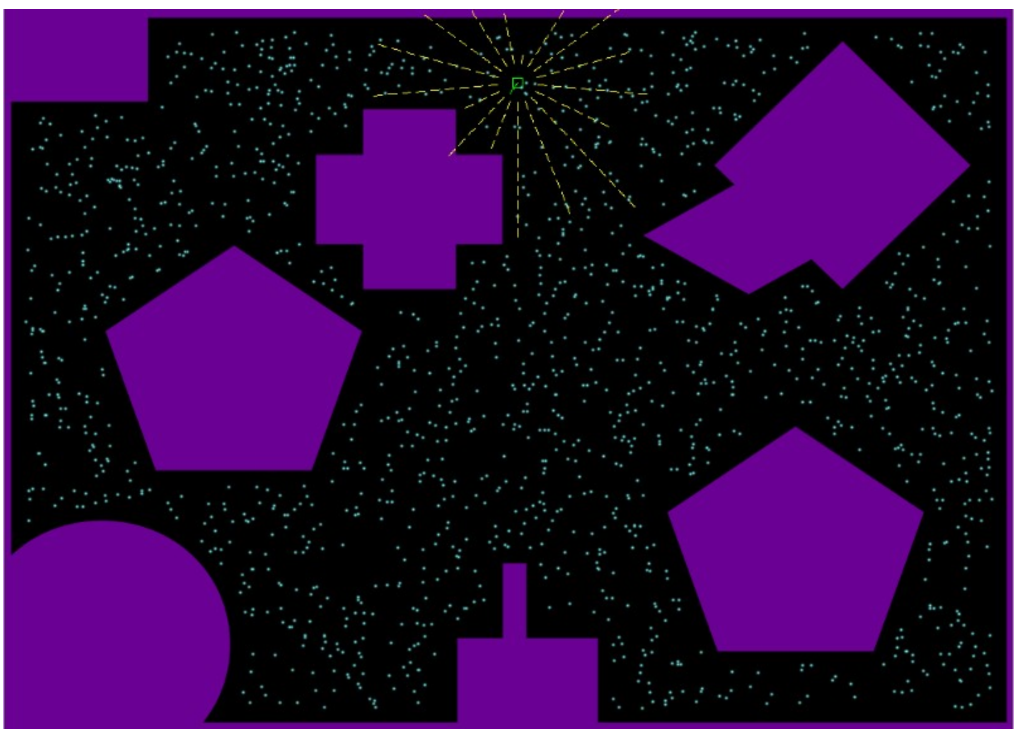
Programming Exercise.

This problem is designed for Linux. Therefore, you must work at the Linux lab IC 406. It is possible to work remotely via ssh but note that the display portion of the program will run very slowly over ssh. Your final solution ***must compile and run on mathlab***.

Download and unpack the starter code from the course website into an empty directory. Unpack the starter code.

Your task:

To implement a particle filter for robot localization. The starter code provides a set of ***maps*** which represent rooms a robot may find itself in. You can design your own if you so wish, all you need to note is that ***perfect black*** is interpreted as empty space. ***Anything not black*** is a solid obstacle. ***Make sure to include a thick border at the edge, otherwise you will crash the program.*** The figure below shows a sample map, the robot, and the set of initial particles.



Your task is to implement the particle filtering loop that will allow the robot to determine its location in the map. You will implement the following components:

- ***- A sensor likelihood estimation function***
- ***- A motion update***
- ***- A sensor update***
- ***- An importance sampling function***

Programming Exercise.

Read and understand all the comments in the starter code. This means both the .c and .h files.
your TA and I will deduct marks for questions whose answer is clearly in the comments.

The comments in the code provide a complete description of your task.

Once you have a working particle filter going discuss within your team the following issues:

- What is the minimum number of particles needed for successful localization In each of the maps given?

- What makes a map difficult from the point of view of localization?

- What happens if instead of a Gaussian noise model, you use a different model? (e.g. instead of using a Gaussian function in the computeLikelihood() function make the likelihood inversely proportional to error).

- Can you think of any way to improve the noise model?

- The motion model for the robot and particles is noisy. Is this a bug or a feature? what happens if there is no noise in the motion? Here you should think about the re-sampling step and what it means to choose a high-probability particle multiple times.

** ***Don't forget to complete the REPORT.TXT file in the starter code! *****

Have fun robot-localizing!

Send questions by email, and ask your TA during tutorial. Drop by during office hours!

Compiling your code:

Use the included ***compile.sh*** script. Whatever you do, your solution must compile with this same script.

General advice:

Your particle filter should be able to localize the robot in ***all*** the maps provided by us given enough particles.

Document any design decisions you make, and explain how your code works.
We will look at the comments in your code as well as to your answers on the REPORT file.

Be mindful of the random seed you are using, and in general, be careful with the importance sampling step. If you are not careful you'll run into trouble quickly.

Remember that all team members ***must understand thoroughly*** the submitted solution and the process that led to it.

Submitting your solution:

Create a single compressed file (.tgz, tar-gzip format) named ***ParticleFilter_teamname.tgz***

This file ***MUST*** contain:

Your code (including all files needed to compile and run it)
Your REPORT.TXT file

This file ***SHOULD NOT CONTAIN***:

Any map images
Any screen-shots
Random junk

Submit your file electronically on mathlab using the command:

```
submit -c cscc85f18 -a Lab2 -f ParticleFilter_teamname.tgz
```

Only one file should be submitted per team. We will deduct marks for duplicates.

We will not accept solutions received after the deadline.