CSC C85 – F.R.A.S.
Project # 2 – Robot Localization

This project will have you implementing a probabilistic localization algorithm on a Lego EV3 Robot. Your task is to make sure the robot can navigate around a color-coded map that represents a city with streets, intersections, and buildings. The project is a good approximation of real-world probabilistic localization scenarios, so the experience you gain here will translate to more complex robots and more difficult tasks.

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

Implement a histogram-based, Markov localization algorithm that iteratively updates the robot's belief about its location on the map given sensor measurements and the robot's actions.

Deal with real-world sensor and signal processing issues – your robot's color sensor will return noisy, unreliable readings. You will learn to handle these to build a robust localization algorithm.

Learn to interact with a mobile robot remotely – you will become familiar with the EV3's robot control API and use it to control the robot and read sensor data in real time from a laptop computer.

Design and build a mobile robot for a specialized task – this means you will consider what the robot needs to do, and use this knowledge to determine the best configuration of the robot's components that will make your design optimal for this task.

Implement software that can handle environmental variation – changes in the map, Illumination conditions (which affect sensor measurements), battery level, etc.

Skills Developed:

Real-time control and sensor reading from a mobile robot from a laptop computer.

Sensor and signal management – including denoising, and management of erroneous measurements.

Writing software that can robustly handle environmental changes and uncertainty.

Reference material:

Your lecture notes on robot localization and sensors.

The EV3 Robot Control API documentation (see the source code for this)

Acknowledgements:

Project testing and implementation by L. Tishkina, Summer 2018.

The Little Lost Robot:

You have taken on an important assignment to build a fully automated robot to locate and dispose of hazardous material transportation. After a long and arduous design and verification process, your robot is ready for roll-out.

On the test day, the robot will be dropped onto a realistic simulation of an accident involving the release of a hazardous substance. The robot's task is to locate a pile of dangerous muck and transport it to a safe location. However, on the test day you find that GPS localization is off-line and the robot's receiver seems to be inoperative. Your little robot is now lost.

Being a careful designer, you have installed a backup system on the robot that will allow it to use landmarks in the environment to locate itself, navigate toward the hazardous muck, and proceed to the safe disposal location. Will your robot find its way? or is it destined to become a pile of rusting junk in some lost corner of the test area?

Project Overview:

You will be in charge of:

* Designing and building a mobile robot that will successfully navigate the map we provide, reaching a specified location, for the hazardous material or the safe storage facility.

* Making appropriate use of the EV3’s sensors to gather information about the environment as the robot travels.

* Implementing a histogram-based localization algorithm just as the one described in lecture to enable your robot to determine its location. This is a probabilistic algorithm, so it will be expected to deal with uncertainty, noise, and a changing environment and still localize the robot.

* Testing and validating that your robot works.

* Completing the questions in this handout given the work you carried out and your observations on how the robot behaves on the map.

Setup

Your robot will be operating within a world consisting of colored regions. The ***map*** colors are meaningful:

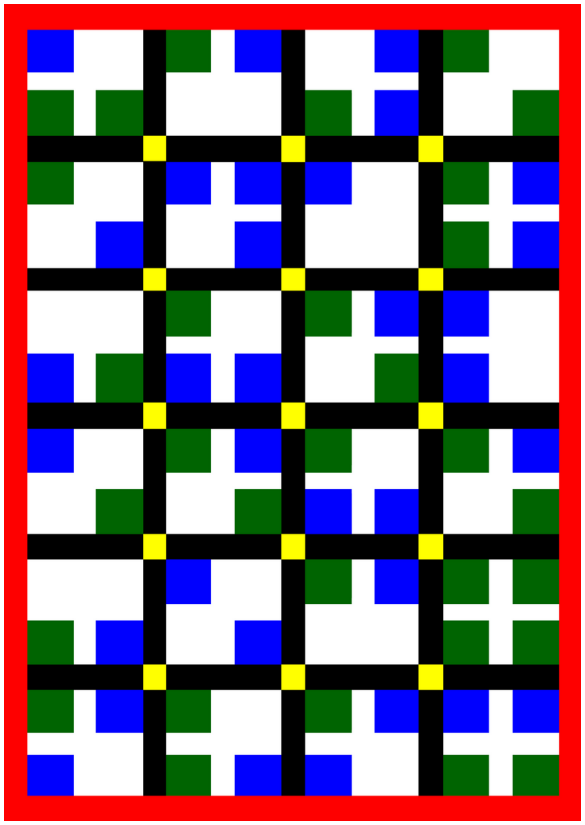
* Yellow - corresponds to street intersections.

* Black – indicates streets. Your robot ***can only travel along streets***.

* Green, Blue – Indicate buildings (the landmarks in our case)

* Red – Map boundaries. Your robot must not wander outside the map.

Setup (continued)



The image at the left shows one map your robot will be operating in.

You will be provided with a printout of the map on which to do testing.

The starter code reads the map from an image file and sets up an array with the building colours at each intersection for you to use.

Your robot will be placed at ***an arbitrary location, facing a random direction.*** It must find and follow streets, stopping at intersections to scan building colours.

Your task is to use the EV3's sensors and your knowledge of robot localization to write a probabilistic localization routine that will enable the robot to determine its location.

Once your robot has determined its location, It must proceed to the specified target and destination location. The destination is specified on the command line when your code is run.

Upon successful completion, your robot may play a tune, do a happy dance, or otherwise signal its success.

Starter kit

Each team will be provided with:

* A printout of the map at a scale suitable for testing. Take good care of it and bring it for testing on project review day. Note that we ***may test on a different map during the project review.*** The map is read from an image file, so you can create and print your own maps for testing.

* A complete EV3 set. You will take this set with you, and be responsible for keeping it in good working condition and safe as long as it is in your care.

* Starter code that provides a framework on which to build your algorithm. Read this code carefully as it describes your task, what is implemented, and what you must complete.

Your Tasks

1) Design and build the mobile robot

* You are free to design and build the robot as you see fit. But you should carefully consider the following:

  - It should be able to navigate the streets easily (i.e. should be small enough to comfortably fit in the map).

  - It will have to make sharp (90 degree) turns at intersections

  - It will have to either navigate backwards easily, or perform 180 degree turns If it reaches the map's boundary.

  - It will need to have the colour sensor placed in such a way that it can scan the map for streets, intersections, building colours, and the boundary.

* Beyond the above, use your imagination. ***We will evaluate your robot in terms of design, mobility, and appropriate use of sensors.*** Particularly cool designs will earn extra credit.

2) Implementation and testing of the localization algorithm

For simplicity, you will only need to determine ***which intersection*** and ***facing direction*** the robot is at.

This means that while the robot is moving along a street, you do not need to update the robot's position.

A) Once a robot hits an intersection, it will do the following:

* Scan the intersection to determine the building colours at the 4 corners of the street. Buildings will be green or blue (or may not be there) and you must design a motion pattern that results in the robot scanning these colours in a ***predictable***, and ***reliable*** order. This is essential so you can check your readings against map data.

  The map data gives you building colours around each of the intersections in the map. See the starter code for details.

* Test your intersection scanning process thoroughly – success in this project depends to a large degree in how good your robot is at scanning colours at intersections. However, keep in mind that since you're implementing a probabilistic method, the method itself should be resistant to noise and the occasional erroneous reading.

Your Tasks

2) Implementation and testing of the localization algorithm (cont...)

(Once a robot hits an intersection, it will do the following: cont...)

* Update the robot's belief about its potential location based on the readings you acquired.

We will use a probabilistic algorithm to determine robot location, this means you have to keep track of the probability the robot is located at a given Intersection.

However, this is not all. You also need to know ***which direction*** the robot is moving. The starter code provides you with an array in which to keep track of the belief the robot has of being at each intersection, and for each possible orientation it may be traveling along.

Q1) What is the total number of probabilities your robot has to keep track of for a map of size m x n?

Initially the robot has no idea where it is, or which direction it is facing.

Q2) What should be the initial value of the probabilities your robot is keeping track of?

You now have to deal with the following problems:

* Noise – sensor readings are noisy and can be unreliable. You need to implement some strategy to deal with this and ensure you get accurate colour readings most of the time.

Q3) Describe your strategy for dealing with sensor noise, and explain why it should work and how you tested it.

* Determine agreement - Given the sensor readings, you have to determine whether there is agreement between what the robot sensed and what the map says the robot should see at a given intersection. ***Here you need to account for the different possible directions the robot may be facing.***

* Update probabilities – Based on the amount of agreement (you need to define what that means) you have to update the probabilities for each possible intersection and facing direction.

Your Tasks

2) Implementation and testing of the localization algorithm (cont...)

(Once a robot hits an intersection, it will do the following: cont...)

(You now have to deal with the following problems: cont...)

***** Update probabilities – cont...

You have to decide ***how*** and by ***how much*** to change the probabilities for each intersection and direction.
You must have a good reason for the particular update strategy you choose. The updates should account for factors such as your estimate of how reliable the sensor readings are.

The update should depend ***exclusively*** on the current measurements. You should not use historical data of previous intersection readings.

After the update, you ***must*** ensure the sum of probabilities over the beliefs array is ***1.0***.

Q4) Describe your probability update function in detail and explain how you defined 'agreement' as well as how you Determined the amount by which to update probabilities.

At this point, the robot has finished processing this intersection.

Two things could happen:

* A specific intersection, and a specific traveling direction has emerged as a ***clear winner*** over the rest and you can conclude the robot has found itself.

  If this is the case – Make your robot emit a distinctive sound, or emit a clever remark.
  Go to step ***C)*** 'completing the mission'

  Otherwise:

B) Move

* The robot is still not sure of its location. It must keep exploring the environment.

* Have your robot move to a different intersection. The exploration pattern is up to you, but should not be always a straight line.

* If the robot hits the boundary of the map it must turn back. It will end up at the same intersection it just scanned.

Your Tasks

2) Implementation and testing of the localization algorithm (cont...)

(Move ... cont...)

Q5) Describe in detail your robot's exploration strategy – it should be more complex than simply moving in the same direction until hitting the boundary of the map. How did you choose this Strategy?

At this point you need to shift the probabilities in the beliefs array to reflect the fact that the robot has moved.

How the probabilities are shifted will depend on what specific action your robot has carried out, and different things need to happen if the robot moves forward, moves back, turns in some direction, or reaches a boundary and turns back.

As a simple example, suppose your robot moved forward to the next Intersection.

Consider what needs to happen to the current beliefs for the intersection at ***(x,y).*** We have four beliefs, one for each possible facing direction. Since the robot moved forward by one intersection, then

- The current belief for the 'up' direction is shifted to (x, y-1)
- The current belief for the 'right' direction is shifted to (x+1, y)
- The current belief for the 'down' direction is shifted to (x,y+1)
- And the current belief for the 'left' direction goes to (x-1,y)

You need to shift all the current beliefs in the correct direction, taking care to handle properly what happens when a belief value shifts out of the map, as well as what happens to interesections that received no shifted belief values.

Q6) Describe in detail how beliefs are shifted when the robot gets to a boundary. Consider this problem carefully, the correct way to handle this is not obvious. Explain clearly what assumptions regarding the robot's motion support your strategy for shifting belief values in this case.

Q7) Describe in detail how beliefs are shifted when the robot turns 90 degrees (either left or right) without going to a different Intersection. Describe your assumptions about robot motion, and how these support your strategy for handling this case.

Spend some time carefully thinking out how to correctly shift the belief values to account for robot actions – if your update is incorrect, localization will very likely fail.

Your Tasks

2) Implementation and testing of the localization algorithm (cont...)

(Move ... cont...)

Note – whether the robot turns out not, it must eventually explore a new intersection, This may require multiple actions (e.g. turning + moving forward). You need to shift beliefs appropriately for ***each*** of these actions.

* Once you updated the probabilities to account for motion, go back to step ***A)*** and process the new intersection.

C) Completing the mission

* Your robot now knows where it is with some certainty.

Q8) Explain your method for deciding when the robot has achieved correct localization.

* Your robot must proceed to the designated location.

The robot must do this without wandering around the map (it already knows where it is, so no more intersection scanning and probability updates should happen). The robot should not attempt to go outside the map or hit a boundary at this point.

* However – if your robot becomes lost, it should be able to determine it's lost and return to the localization loop.

* Once you have reached the specified target the robot's mission is complete.

Have your robot do a happy dance, play a song, or emit a facetious remark. ***Bonus marks for cool behaviour upon completion of the mission.***

3) General comments

As stated above, your robot's localization loop will consist of

1) Scanning an intersection and updating beliefs based on agreement between the sensed building colours and the map.

2) Moving to a new intersection, shifting the beliefs accordingly to account for the robot's actions

The loop ends when localization has been achieved.

Your Tasks

(3) General comments... cont.)

Be sure to read and understand all the comments in the starter code before you start tweaking it.

Comment on the following:

Q9) Briefly explain your robot design choices, including sensor placement and robot mobility considerations.

Q10) What parameters did you find affect the most the ability of the robot to determine its location?

Q11) Can external factors (illumination in the room, for example) affect the ability of the robot to find itself?

Q12) On average, how many intersections does your robot need to visit to achieve successful localization?

Q13) Are there locations in the map that make localization harder? (i.e. does it matter where your robot starts its journey? Why?)

You will submit your team's answers to all the questions in this handout along with your code, electronically, by the deadline. No hand-written answers will be accepted. Please make sure to identify all team members including full name and student number.

How to fail in this project...

1) Ignore sensor noise and sensor behaviour

Simply assume that the colour sensor is perfect regardless of how you placed it on your bot. Assume that your program can just read the colour values and use them 'as is'. Also, assume your bot's sensor doesn't need shielding for stray light.

2) Ignore noise and inaccuracies in robot motion

Assume that a command to drive forward at certain speed for a specified amount of time always produces the exact same amount of displacement in exactly the same direction. Similarly assume turning always works perfectly.

Assume battery charge doesn't matter. Assume wheels don't slip on the surface and threads don't get stuck.

Assume the map is always perfectly flat and always exactly the same size.

3) Use open-loop control

Assume you can use timed motion commands such as *turn left for 2 sec.* to position the sensor and robot where you want them to be. Don't use sensor readings, after all, the colours in the map are just decorative.

Also assume the map is always exactly flat and the same size.

4) Form over substance

Design the best-looking bot ever that, sadly, can not move because you used the motors to drive awesome-looking crushing-grips-of-doom.

5) Start as late as possible

You can design and build your bot, write the localization software, test and debug, and make it work in just one afternoon... right?

6) Ignore sensor calibration

The code says it’s optional – as in, wearing a helmet while riding a bike is optional…

Evaluation

Project evaluation will take place at the Hardware Lab

* You will do a brief presentation for everyone, explain your design, and demonstrate its operation '***live***', on the same map we gave you for testing.

* Marking will be based on how well your robot manages to solve the task we gave you.

  However:

  - We do not expect perfection! Things happen in practice with robots, they don't always work as they should.

  - We will pay attention to your design, your answers to questions we will ask while you present, and whether or not different components of the solution are present.

  - The answers to questions in the report are also worth a certain amount of marks toward your project grade.

  - We will ask questions to both team members and expect both to be able to answer any of our questions.

* ***We will make a short video of your presentation/robot demo*** for reference as we review the marks for the entire class. These videos ***will not be posted***. They are exclusively to be used for marking purposes.

* ***Compress your solution (only the code please, no map images), along With the answers for this handout's questions into a .zip file. Submit this file electronically on Quercus with the name:***

  `EV3_Localization_Teamname.zip`

  Only one file should be submitted per team.

Finding Help

For issues related to the EV3 and robot control API, starter code, map data format, and so on, please contact your TA during tutorial or in office hours.

For questions regarding the localization algorithm, probabilities, sensors, or your own ideas about how to solve the problem. Contact your course instructor anytime by email, or drop by at my office.

Your TA can also help you with the localization algorithm design and implementation.

We are happy to help, so please do not hesitate to come to us with questions.

Project Timeline

Phase 1 – Oct. 3 – Oct. 14 (includes reading week!):

1) – ***Design your bot and implement the robot design that will be used for the localization task***

Pick up your EV3 robotic kit at the lab if you haven't.

- Consider at least 2 different designs for the bot. Consider carefully the task it needs to solve, how it will drive/turn/move over the map, and that it has to be able to read the colour of the road, intersections, and at each intersection it has to read the colours of the buildings around it in a repeatable and fixed order.

2) – ***Work on implementing routines to have your bot***
**** Detect intersections*** **** Orient itself to follow a street***
**** Do right-angle turns from one street onto a perpendicular one at the intersection***
**** Perform an intersection scan and read all 4 colours around it (make a different sound for each colour to show it reads the correct values)***

First progress check: Make a video!! - 25% of project grade

It's reading week, so we won't see you. Instead, make a video that shows your bot:
- Moving around the map (we need to see you built it and can control is with the Laptop).
- Demonstrate your bot's ability to ***follow streets***
- Demonstrate your bot's ability to ***detect and scan an intersection***
- Demonstrate your bot's ability to ***turn from one street onto another***
- Send us ***a link to your video, doesn't matter where you upload it as long as we can access it***. ***The video must be in by Oct 14, 9pm.***

There will be office hours for mentoring and to help you with these tasks including during reading week, I'll post the times on Piazza.

Phase 2 – Oct. 17 – Nov. 3:

During this phase – ***Implement the histogram method for robot localization as described in the handout***
**** Your robot should be able to find itself and get to the specified destination having started at any of the locations in the map***

Project Review – full team attendance is <u>mandatory</u> - 65% of the project grade Thu. Nov 3, during the Lab (practical) time slot

*** ***Code review and handout: 10% of the project mark*** ***

Team schedules for the project reviews and progress checks will be arranged by Doodle poll

Feedback (for yourselves and us) – complete this part for bonus marks.

Do you feel this project helped you to understand the following ideas:
(please answer yes, no, somewhat, and comment on what if anything could have improved your learning experience).

1) The main steps of a robot localization algorithm

2) Dealing with real-world sensors

3) Dealing with real-world motors and robot motions

4) Using probabilities to keep track of ***belief*** about a robot's position

5) Managing uncertainty (e.g. in the quality of sensor readings, or accuracy of robot motions)

6) Do you feel confident you could implement a probabilistic localization algorithm for a more complex robot, working in a real world environment? (please explain your answer)

7) Any other comments regarding your experience with this project?