COMP.4510 and COMP.5490
Fall 2017

PS6
Mapping with Known Poses

Out: Tuesday, October 24
Due: Thursday, November 2 before the start of class

Objective: Given perfect knowledge of a robot's pose in a hallway (*x, y, theta*), build a map of its surroundings, using the log odds ration derivation.

Reading

The following material is relevant to this problem set:

- 9.1 Introduction to Occupancy Grid Mapping
- 9.2 The Occupancy Grid Mapping Algorithm

Files Needed

Use the uml_mcl code to provide a launch file with the hallway world and the */robot/base_scan* and */robot/cmd_vel* topics.

You must use the two topics to move around and sense the world. However, you may also subscribe to the */stage/base_pose_ground_truth* topic to learn where you really are. So, you'll still be using the sloppy motors to move, and the bad laser to sense the world, but you have your "known pose." You may *not* use the ray-tracing code as part of the solution.

Get the code:

```
$ git clone https://github.com/DeepBlue14/mcl_ws.git
```

After cloning, do the following:

```
$ cd mcl_ws
$ rm –r devel build
$ catkin_make
$ source devel/setup.bash
$ cd src/no_weights/src/raycaster
$ make clean
$ make
$ cd ../../../with_weights/src/raycaster
$ make clean
$ make
```

You should now be able to:

```
$ roslaunch uml_mcl mcl.launch
```

Followed by:

```
$ rosrun no_weights no_weights.py
```

Or:

```
$ rosrun with_weights with_weights.py
```

Setup ****Only if you are NOT using VLabs****

These things might be needed:

```
$ sudo apt-get install libcv-dev
$ sudo apt-get install libcvaux-dev libhighgui-dev
$ sudo apt-get install python-numpy python-opengl
$ sudo apt-get install swig
```

The Assignment

Implement the log odds ratio algorithm as described in the text and lecture:

1: **Algorithm occupancy_grid_mapping**($\{l_{t-1,i}\}, x_t, z_t$):
2: for all cells $\mathbf{m}_i$ do
3: if $\mathbf{m}_i$ in perceptual field of z_t then
4: $l_{t,i} = l_{t-1,i} + \textbf{inverse\_sensor\_model}(\mathbf{m}_i, x_t, z_t) - l_0$
5: else
6: $l_{t,i} = l_{t-1,i}$
7: endif
8: endfor
9: return $\{l_{t,i}\}$

Build a map of the world where each cell is represented by the probability of its being occupied.

Drive the robot around (probably with a simple wall-avoidance reactive algorithm, similar to the one you implemented in PS3) and have the map accumulate knowledge.

Implement some fashion of displaying the map, most likely as an image with white to black pixel where white is *p(occupancy)* = 0.0, and black is *p(occupancy)* = 1.0.

To Turn In

Turn in the following:

1. Your code to implement the “mapping with known poses” algorithm. Submit your entire ROS package.

2. A 300 to 500 word discussion about what was interesting about the assignment. Include as part of the discussion:
    - Challenges you encountered (and how you solved them)
    - Representation of the world map
    - Performance of your implementation
    - Code excerpt accomplishing the probabilistic updates of the map

This write-up file should contain your name and email address.

Submit

If you are a CS major, submit material you wish to deliver electronically via:

$ submit jkuczyns ps6 zip-file

Remember that the "submit" command only works on the cs and mercury servers. Use "scp" to transport your code from VLabs to cs.uml.edu.

If you are NOT a CS major, email your zip file to jkuczyns@sc.uml.edu.

Credits

This assignment was jointly developed by Fred Martin, James Dalphond, and Nat Tuck. Thanks to Eric McCann for updating it for 2015, and James Kuczynski for updating it for 2017.