Occupancy Grid Mapping

MEE 4411/5411 Project 1, Phase 2

"Due:" Thursday, September 28th

For this assignment, you will write code to read in a text file describing a map and then use this information to create a 2D occupancy grid representation of the environment. This will be used in the next Phase to generate plans for the robot to drive through the environment.

1 Reading in the Map File [MEE 5411 Only]

Complete the implementation of `myReadEnv.m`. `myReadEnv()` takes the filepath to an environment file (e.x. `'sample_maps/map0.txt'` and returns the bounding box of the environment and of the obstacles. For simplicity, we will assume that all environments are rectangular and that all obstacles are axis-aligned rectangles (meaning that they are not rotated with respect to the environment rectangle).

The environment is specified by a text file. The format is straightforward. Here's an example:

```
# An example environment
# boundary xmin ymin xmax ymax
boundary   0    0    45   35

# block xmin ymin xmax ymax
block   1.0  1.0  3.0  3.0
block   20.0 10.0 21.0 20.0
```

The format uses # for comments. The rectangular boundary extents for this environment are $(x_0, y_0) = (0, 0)$ (lower left) and $(x_1, y_1) = (45, 35)$ (upper right). A `block` line also specifies the rectangular boundary. All values are whitespace delimited (meaning they are separated by whitespace, as opposed to commas, semicolons, etc.) and all numerical values are represented as a float. All distances are in meters. Blocks can overlap and the file can have blank lines.

You will need to write code to read in the text file. Hint: `fopen` will open a text file. The Matlab documentation page for `fopen` also contains references to other functions that let you read the file line by line at the bottom of the page in the "See Also" section. `myReadEnv` should output two matrices, one called `boundary` that is a 1×4 matrix representing the boundary and one called `blocks` that is a $n \times 4$ matrix representing the blocks, where each row is a single block.

1.1 MEE 4411

I have distributed another file, `readEnv.p` that already does this task. (A `.p` file is a protected file, meaning you cannot see the code but you can run it exactly like a regular `.m` file. Matlab uses `.p` files in a lot of the code they implement and distribute.) Those in MEE 4411 can use this file for Phase 2 while those in MEE 5411 can use this to debug their own code or to begin working on the next pieces of Phase 2. The input and output structure is exactly the same as `myReadEnv`. Those in MEE 4411 can implement `myReadEnv` for extra credit.

2 Creating the Occupancy Grid

Once you have successfully read in the text file, the next step is to take advantage of tools that Matlab provides for you. Being able to pick up and use code that somebody else has written is an important skill for any (aspiring) roboticist. Specifically, we will be using the `BinaryOccupancyGrid` provided by the Robotics Systems Toolbox. This was introduced in Matlab 2015a, so if you have an older version then you will have to update. If for some reason the version of Matlab that you have does not include this toolbox, then I would encourage you to update to the latest version of Matlab (2017a). The online documentation provides plenty of information, including examples of how to use this code. This page talks about occupancy grid more generally and how Matlab represents them.

Your task is to implement the function `envToOccGrid.m`, which takes as its input the output matrices of `readEnv` as well as `sz` (indicating the size of a grid cell in meters per cell) and `margin` (indicating the amount to inflate each obstacle by to create the configuration space). The function then outputs a `BinaryOccupancyGrid` object call `map`. You need to use the information contained in the inputs to fill in the correct fields of the `map` objects.

To be able to more easily debug this, I have provided a function called `plotEnv`, which takes in the `map` object and the `blocks` matrix and draws the environment (placing the blocks on top of the occupancy grid map). To be sure that your function works, make sure to test out a lot of different `res` and `margin` values. Pay close attention to cases where the grid cells line up exactly with the obstacles! One example is using `sz = 1` and `margin = 0` in `map2`.

I strongly encourage you to create your own map files. I will give extra credit to anyone who comes up with a map that I really like and want to use for planning in the next phase of the project, or next year when I teach this again.

3 Submission

When you are finished you may submit your code by putting it all into a single `.zip` file and emailing it to pdames@temple.edu with subject line “MEE4411 Project 1.2” (clicking the previous link will automatically start a correctly formatted email for you). Prof. Dames will test your submission and send you a list of items to fix. You are encouraged to update your code to fix any bugs, however each student is limited to 3 submissions per phase of the project.