Lab 7 – SLAM

Overview

Aside from onboard sensors, we can control and program robots with remote sensors like cameras. This process usually involves mapping the area around the robot and locate objects of interests. With this mapping and localization information, we can then direct the movement of the robot.

Objectives

- Continued practice with OpenCV
- Implementation of A*
- Experience with a simple mapping and localization task

Procedure

1. Create A*
    a. Download and run the test scripts provided to make sure they work.
        i. The astar method is blank so it just returns an empty list and doesn't draw a path.
    b. Create the astar method using the pseudocode provided in Figure 1.
        i. Note, there were a few elements missing from the pseudocode shown in lecture so use the provided Figure.
    c. Run astar on the three test images to verify it works.
    d. Take screenshots of the test images with the paths drawn in for the Results.

```
visited = []
p = c(n_s → n_s) + h(n_s)
F=[([n_s], p)]
while F != [ ]:
    testPath = pop(F) =[n_s, … , n_k]
    if n_k not in visited:
        visited.append(n_k)
        if goal(n_k ):
            return testPath
        newPaths=[ ]
        for n in neighbors(n_k):
            if n not in testPath:
                p = c(n_s → n_k) + h(n_k)
                newPaths.append((testPath+n,p))
        F=F+newPaths
return []
```

Figure 1: A algorithm. Fixed from lecture slides.*

2. Task 2 Image
    a. For the task 2 image, extract the start and end location by finding the centroid of the blue and green circles respectively.
    b. Threshold the task 2 image so that the pixels for the obstacles (red rectangles) are white and everything else is black.
    c. Run astar on the threshold image with the centroids of the blue and green circles as the start and end locations.
    d. Take screenshots of the task images with the paths drawn in for the Results.
3. Task 3 Image
    a. For the task 3 image, extract the start and end location by finding the centroid of the blue and green circles respectively.
    b. Threshold the task 3 image so that the pixels for the line are black and everything else is white.

c. Run astar on the threshold image with the centroids of the blue and green circles as the start and end locations.
    i. **Note, if the centroids of the green or blue circles don't fall exactly on the line, you can move them slightly to get on the line.**
d. Take screenshots of the task images with the paths drawn in for the Results.

Questions

1. Aside from A*, we could have also used a Greedy search where the cost of a path is just its distance to the goal. Would a Greedy search have worked better here than A*? Why or why not?

2. Why did you have to rescale the images for task 2 and 3?

3. A* produced a path, which is a list of points. If we were looking at real top down view of the robot, how could we translate those points to actual robot movements?

Write up reminders

1. Methods
    a. Screenshot of your A* algorithm
    b. Describe Breadth First Search and A*
        i. Provide pseudocode and/or diagrams
    c. Briefly review the OpenCV operations you used
2. Results (Note, b/c there are only 3 tasks, the weight of the Results section is lower)
    a. Screenshots of the images with and without the paths drawn in. Use the scaled up versions of the test images so that they are bigger.
    b. Remember to describe the data you are presenting.
3. Supplementary Material and Submission
    a. Submission – One zipped folder named after the team members usernames. In the folder should be the report, a folder for your code, and a folder for all the videos.
    b. Supplementary Material – Should contain the following sub-sections
        i. Section 1
            1. List of python files you submitted and a 1-2 line description of each
        ii. Section 2
            1. Screenshots/pdfs of your code

Rubric – Lab 7

General formatting (10)

- Title, date, name, section headings
- Miscellaneous style and formatting

Abstract (15)

- Gives a brief intro or overview of the Lab and/or topics covered.
- Gives a brief summary of the Results and main points from the Discussion.
- Well written and concise without typos or grammar issues.

Methods (30)

- Possess all relevant equations and algorithm explanations
- Tasks and procedures are described.
- Looks nice and is easy to follow.

Results/Supplementary (30)

- Figures and text for all experiments.
- Data and analysis for all experiments. Includes any required statistical summaries or tests.
- Figures are properly formatted (labels, legends, captions, etc.).
- Required videos and code.

Discussion (15)

- All questions are answered.
- Well written and concise without typos or grammar issues.