Project #4

Robot Vision-Based Localization

Introduction

- NASA hired your robot explorer to travel to a distant 2-dimensional planet
- Unfortunately, you have no idea how to figure out where your robot is
- Your job: To design a robot vision algorithm that determines position and orientation of your robot.

Your Task

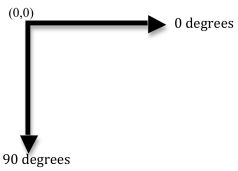
- Design a vision algorithm based on a series of images provided as input (input.dat)

TestImage0.pgm
TestImage1.pgm
TestImage2.pgm
TestImage3.pgm
TestImage4.pgm
ENDOFINPUT

TestImage0.pgm TestImage1.pgm TestImage2.pgm TestImage3.pgm TestImage4.pgm

Output.dat

- Given the map environment, your program must output:
  - For each image file, your program must print the (x,y) location associated with the center and the orientation of the robot, each on a separate line in the file "output.dat" (Note: All numbers must be positive).
  - Robot location must be provided as pixel coordinates and orientation in degrees. Assume that your coordinate system is based on the representation below.



Grading Criteria

- Optimal credit (20 points for each case): Program outputs a correct orientation within 10 degrees and pixel coordinates within a total error of 10 pixels. The program must run within 1.5 minutes.
- Suboptimal credit (16 points for each case): Program outputs a correct orientation within 20 degrees and pixel coordinates within a total error of 30 pixels. The program must run within 1.5 minutes.
- Partial credit (10 points for each case): Program outputs a correct orientation but not the correct pixel coordinates (or vice-versa). If you believe your program will take longer than 1.5 minutes to run through an image, you must indicate that via a message to the console.
- Note: You are allowed to use the code provided in ThresholdProgram.zip (under Resources) without penalty (but this does not count as your separate class file).
- Deductions
  - Program will not compile (60 pts)
  - Program crashes or does not terminate (20 pts)
  - Program does not contain at least one separate class file (20 pts)
  - Program does not comply with good design constraints or requirements (e.g. non-working makefile, no zip file, etc.) (2 - 10 pts)

Submission

- A zip file containing your code is to be submitted via T-Square (ECE3090-Assignments-Project4) by the DUE date of November 2nd
- We will test your code on the Jinx cluster so make sure your program correctly compiles and runs on that system (development can be done on your machine - e.g by using GNU ARM). Information on the cluster is located at: http://support.cc.gatech.edu/facilities/instructional-labs/jinx-cluster
- Do your own work and abide by Georgia Tech's Honor Code!!