Assignment 2: Vision
CS 393R: Autonomous Robots
9/5/2018
Assignment due: 11:59:59 PM Wednesday, September 19, 2018

Your task is to write code to detect the ball, the goal, and the beacons while moving around the field. As a starting point, you may want to consider this paper from the reading list:

> Fast and Cheap Color Image Segmentation for Interactive Robots
> James Bruce, Tucker Balch, and Manuela Veloso
> `http://www.cs.cmu.edu/~mmv/papers/wirevision00.pdf`

You will use blob detection to identify objects on the field, access them through world objects, and fill in the appropriate information so that they display correctly in the Vision and World windows of the tool.

You will probably find both of these windows useful for debugging. When a beacon is detected and its appropriate world object is filled out (vision distance, image center X and Y, etc), the tool will draw the beacon over the image to indicate its perceived size and location. The world window will draw the observed beacon at the distance and bearing that you provide, assuming that you are at the center of the field.

You are encouraged to debug your initial implementation using logs using the tool's "Run Core" feature with vision processing enabled. This will allow you to run your vision code on logs through the tool, making it easy to tweak your code and test on a consistent dataset.

This assignment will involve both detecting true positives and throwing out false positives. Examples of false goals, balls, and beacons will be provided, and it will be your job to make sure that these are not detected during your evaluation.

The assignment will also require you to have a log prepared ahead of time. The log should be 100 frames long and contain observations of balls, goals, and each beacon variation. The log must contain frames in which all three of these object types are simultaneously visible. Every frame must contain some of these objects, and objects must be observed at varying distances throughout the log. It is acceptable to combine or alter logs using the Log Edit window of the tool.

After your evaluation, each student should write a short but professional memo that clearly and concisely describes what you did, what problems you encountered, how you overcame them, and how successful you were in the end. Make sure to compare your methods against what you did for Assignment 1. Also be sure to briefly describe your contribution to the final outcome.

Checklist:

[____] (1 point) Show that your code successfully detects all beacons in your premade log.

[____] (1 point) Show that your code successfully detects all goals in your premade log.

[____] (1 point) Show that your code successfully detects all balls in your premade log.

[____] (1 points) Show that your code doesn't detect false positives for beacons, goals, or balls in your premade log.

[____] (1 point) Show that your beacon distance estimates are accurate within 10% of ground truth at a range of .5 to 2 meters.

[____] (2 points) During evaluation I will arrange beacons around your robot, similar to how they are arranged in the World window. Show that your robot can turn in place and detect these beacons with reasonable distance estimates. This will be evaluated based on the streamed segmented image in the Vision window. Your robot should not detect any false positives.

[____] (1 point) Show that your robot's gaze can follow the ball as it rolls by the robot's feet. The ball will be rolled at approximately 2m/s. The robot should scan its head for the ball's initial position and watch the ball roll by until it stops or goes out of view.

[____] (2 points) Clarity and quality of your memo. Email it -- along with any code files you changed -- to Josiah and Peter by the due date.

Extra Credit:

[____] (1 point) Implement an algorithm to determine if a beacon is occluded, with only its left or right half showing. Indicate this through the Vision window by visually altering the beacon overlays that are drawn. Possible indicators include altering overlay color, adding a text to the overlay, etc.

[____] (1 point) Develop an algorithm to reliably estimate skewed goal distances. The robot should be able to estimate the goal's distance within 15% of ground truth. The goal will be rotated a maximum of 60 degrees relative to the robot, from distances of 1 to 3 meters.

Initials: ______ TA ______ Teammates ______ ______ ______