Assignment 4: Vision
CS 393R: Robotics

Due Date: Thursday, October 20, 2011

Your task: Write code to have the Nao walk and scan its head. Write code to detect each of the six beacons. Learn to take and use logs.

This assignment is to be done **individually**. You will share a robot with your assignment 1-3 partner.

In this assignment, we will be using the humanoid Nao robots and the UTNaoTool (a tool developed by UT Austin Villa to develop and debug code on the Nao robots). You will be using UT Austin Villa's Nao codebase, and filling in missing parts of the vision and behavior modules. See http://www.cs.utexas.edu/~pstone/Courses/393Rfall11/resources/nao.html for information about getting the UT Austin Villa codebase set up on the lab machines, taking care of the Naos, taking and using logs, and other information you will want to know about using our codebase. There are three main components to this assignment.

First, walking and head panning using the **Nao**:
In $NAO_HOME/core/behavior/BehaviorModule.cpp, you should create a behavior that has the robot walk and scan its head side to side or in a figure 8 (or something similar - watch Robocup videos for examples). You will be using this walk with head scan when taking logs - both for this assignment where you will be detecting beacons and in the next assignment where you will be localizing off of beacons.

Second, taking and using **logs**:
See the webpage mentioned above to learn how to take and use logs.

Third, detecting **beacons** (substantially harder than the previous two):
You will implement code in $NAO_HOME/core/vision/VisionModule.cpp (and add headers in $NAO_HOME/core/vision/VisionModule.h if you create additional functions) to detect beacons. You must implement the detectBeacons and getDistanceToBeacon functions. You may find it useful to add additional functions to help you implement these functions. Be sure that you write strong enough code such that **all** beacons are detected, but no non-beacons are detected as beacons. For example, your code should not pick up beacons with three colors (ie, one with pink, then blue, then pink) or mis-shappen beacons (for example, one with a top colored section that is thinner than the bottom section). You will have to handle other non-beacons that are not mentioned here, so be thoughtful and thorough when writing your code. You should also be able to recognize beacons at distances both near and far.

As you edit the two files in $NAO_HOME/core/vision, be sure to use the compile script in $NAO_HOME/build to compile swig if you change the .h file. As you make changes in the .cpp file, call make in $NAO_HOME/tools/UTNaoTool for your changes to be reflected when you open the tool and view a log using run core.

You will find the vision window of the tool useful for debugging. It has one large image on the left and 6 smaller ones on the right (not all are used). Clicking on any of the small images make it the big image. The first image is the raw camera image; the second is the segmented image. The fifth image shows detected beacons.

Note that you can do much of your beacon detection work before even turning a robot on. You should start by testing your code on the logs I provide (vision1.log and vision2.log) in $NAO_HOME/logs, which

were taken in the lab. Only once your code is doing well on these logs should you take new logs on the robot.

Checklist: (Scores will be out of 10)

[____] (1 point) Walk and scan head (both pan and tilt should change such that maximal area is covered by the scan).

[____] (1 point) Show that your detectDistanceToBeacon function provides a relatively good estimate of the distance to the beacon. How you show me is up to you.

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

[____] (1 point) Show that your code doesn't detecting anything that is a non-beacon as a beacon in provided logs.

[____] (2 points) Show that your code successfully detects all beacons without detecting anything else as a beacon in a log you took sometime before the demo. The robot should have been walking and scanning its head when the log was taken, and at least two non-beacons must be present and correctly handled.

[____] (2 points) Take a log (or logs) during your demo. Then show me in view log that your code successfully detects all beacons without detecting anything else as a beacon. The robot will need to walk and scan its head while you take the log. I will bring some non-beacons to test with (in addition to the real beacons).

[____] (2 points) Clarity and quality of your memo. Email it - along with any code files you changed - to Peter and Katie by class time on October 20.

Extra Credit:

[____] (1 point) Implement an algorithm to determine if a beacon is partially observable, and correctly set the partiallyVisible flag. Display a message in the Log window of the UTNaoTool noting whether each beacon in each frame is partially observable.