Homework 4 – out 9/24/03; due 10/1/2003

15-491 CMRoboBits

1 Introduction

This homework covers robot vision. Vision is a very important part of the robot's sensory input. It allows for the completion of a much larger variety of tasks. This assignment includes some written questions as well as some programming to help you become familiar with the AIBO's vision system. You will learn the about the robot's vision system by first calibrating it, and then using it to accomplish a visual task.

2 Background

There are many algorithms available to analyze images for feature detection. Efficiency is generally a large concern as images are inputted rather quickly. You will become familiar with one of the methods for processing such information as described in Lecture 4. It is a very good algorithm for detecting solid colored objects and will work very well for the visual servoing part of this assignment. By greatly reducing the amount of information we have about the image, we can classify the color of the object using a lookup table. We then group these colors together to partition the image into objects and are able to gather information about each one.

3 Assignment

The assignment is broken down into 3 parts:

Part 1: (20 points)
- Vision questions.

Part 2: (40 points)
- Calibration of the vision system

Part 3: (40 points)
- Visual servoing

4 Lab work

- Calibration
  - For this part of the assignment, you will be calibrating the vision. You should calibrate the vision for the following objects: the carpet, the charging station color tower, the charging station circle marker, and the orange part of the wheels on the speedboard. Label the green parts of the charging station items with the same color green as the carpet. You will be graded partially on the ability of your calibration to correctly segment images taken by other groups, so make sure that you get all the relevant lighting conditions in your example images.
  - For calibration instructions see the handout: calibration.pdf
- Visual servoing

- For this part of the assignment, you will be using your newfound knowledge of regions and runs to perform a simple control task. The task is to walk to the left or right in such a way as to position a border of the green carpet in the middle of the image. This is achieved by using a simple proportional controller using direct feedback from the image. An example behavior is provided which performs this control for centering the orange ball left to right on the image. This behavior can be found in dogs/agent/Behaviors/VisualServo.cc. You will probably not need to modify this behavior. This behavior uses the output of Vision::findVisualServoingError() to determine the error in the image. This function can be found in dogs/agent/Vision/Vision.cc around line 2690. The example provided uses the average position of the center of an orange ball to calculate the error. The example comments will guide you through the process of accessing the regions and the runs within a region. You will need to modify this function to find the error of the green border from the center of the image. The robot should end up in a position where the largest green region on the image has a boundary centered on the image (in the x coordinate). Your error function should exactly match this goal configuration. The example behavior will likely work better when using the green boundary then when using the orange ball so don't waste time trying to improve it unless you already have the vision done.

- There is a tool that may help you to debug and test your error calculation. The tool will only run under Linux because it requires X. The tool is called vis_test and can be made by running make in the util/vision_test directory. In order to run the tool you must get some example images. You can get example images uses 'Camera' like you did in the calibration part of the assignment. You can get many examples quickly by setting 'dump_vision_rle=3' in '/memstick/config/spout.cfg'. The 3 indicates that the robot should save a run length encoded, color segmented image every 3rd frame. The images will be saved to the log file like they were in the calibration part of the assignment when you press the back head button. You must have the 'LogWrite' behavior in your run.cfg in order for this to work. To extract the images, run 'log_extract' like before but use 'dogs/util/log_processing/log_extractor/extract_configs/vis_rle.cfg' instead of 'vis_raw.cfg'. Finally, to run the vision test program, copy the 'i*.ppm', 'c*.ppm', and 'i*.inf' files to the dogs/util/vision_test directory. You will have raw 'i*.ppm' files if you used 'Camera' and colorized 'c*.ppm' if you used 'LogWrite'. Run the program by 'vis_test i*.ppm' or 'vis_test c*.ppm' as appropriate. The program will run your servo error calculation code on the first image and print the result to the terminal window. To go to the next image, go to the GUI and press the space bar. To go back an image, press the backspace key while the GUI has the focus. You use q to quit or simply use Control-C to kill the program. Any pprintf calls you make in your function will also be displayed to the console. You will be using this tool again in next week's assignment.

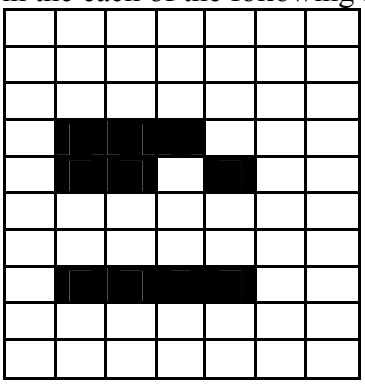
5 Questions

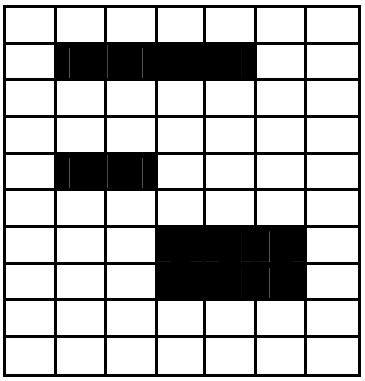
1. Color Segmentation

(14, 130,100)	(15, 128,110)	(12, 125,103)	(26, 98,102)
(15, 128,110)	(26, 98,102)	(14, 130,100)	(134, 113,111)
(26, 98,102)	(126, 123,101)	(126, 123,101)	(116, 113,121)
(124, 114,112)	(134, 113,111)	(26, 98,102)	(12, 125,103)
(12, 125, 103)	(14, 130, 100)	(12, 125, 103)	(14, 130,100)

The above table represents a black and white image in the YUV color space. Draw the image similar to the images drawn below. Provide a threshold that could be used on the Y channel to segment the image into the colors: black and white.

2. How many runs are in the each of the following images? How many regions?





6 Grading

- Part 1
  - Question 1: 10 points
  - Question 2: 10 points
- Part 2
  - Calibration performed: 30 points
  - Performance of calibration results: 10 points
- Part 3
  - Robot stops at edge of carpet: 40 points
  - Robot almost stops at edge of carpet: 35 points
  - Robot uses information from runs of green color: 20 points