COMS 4733 Lab 3, Due 10/25/16 11 AM electronically

In this assignment, we will explore vision processing with a mobile robot. We have some tasks below, ordered by difficulty.

1. (100 points) Color marker tracking. In this task, we will explore tracking a distinguished color marker. Create a target with a prominent unique color that the robot can image. Hold the target on a stick or other method and have the robot image it and track/follow the target as you move it in the robot's workspace. Your robot should maintain a constant distance from the target, moving toward it if the target starts to go farther away, and backing up if the target is moving towards the robot. The robot will also need to rotate and translate if the target moves at an angle – the idea is to keep the target centered in the image. Methods:
    a. You will initiate the color tracker manually. Take an image and have the user click on the image and draw a rectangle around the region you want to track. By computing statistics on the pixel colors in this region, you can create color thresholds to segment the target. Which color space you use may improve your system's performance. The standard Python OpenCV BGR color space (Blue, Green, Red) is more susceptible to illumination changes. You can try another color space called HSV (Hue, Saturation, Value)
    b. Using the thresholds above, create a binary image: every pixel is either white (target regions to track, color 255) or black (non-target regions, color 0. Then find the largest target blob in the image, and calculate its centroid and area. To help you get rid of noise and small artifacts, you can use a series of image filters to a) Gaussian Blur the image –smooth it, b) Erode the image (morphological operation) to get rid of small blobs and c) Dilate the image to amplify and connect smaller regions into a larger blobs.
    c. Now, take a new image and again threshold and binarize the image. Calculate the centroid and area of the largest blob in the new image, and compare it to the previous centroid and area of this blob. If they don't change, the target hasn't moved. If they do, you need to move your robot to either increase or decrease the blob area (move forward and back) and rotate to keep the blob centered (centroid location).
    d. You will have to play a bit with the **gains** on your robot's movement, i.e. how fast and far to move to re-adjust the image. Keep in mind you are doing this continuously as each image frame is read in real-time. Given the web link for the images, you probably will only get 2 or 3 frames per second which will help determine how fast to move the robot. You can also reduce the camera resolution to allow faster processing and a possibly higher frame rate.
    e. Move the marker to make the robot follow you. Show that it will stop when you stop, and turn when you turn, etc.

Note: a good strategy is to do the image processing on your laptop first to test it. When it is working you can then port the code to the Raspberry Pi and integrate the motion control.

Extra Credit: (10 points): Have your robot track a flesh colored object (i.e. your hand!) and determine from user specified hand gestures if the robot should move forward, backward, turn left or turn right.