ESE650 Project 1: Color Segmentation

Due Date: **1/30/2014 at 1:30pm** on Canvas, and in class

Many of today's robotics applications require a world model to be built from camera data. In this project, you will take in an image of a scene with a red barrel, and output the relative world coordinates of that barrel.

To do this, you will need to implement a learning algorithm that segments color images to detect a red barrel using a robot's camera. Given a set of training images, you will hand-label some training examples with discrete color labels. From these training examples, you will build a color classifier and red barrel detector.

You will then use your algorithms to mark the center of the detected barrel and display the distance to the barrel on new test images.

1. A set of training images taken from the robot's color camera will be available at http://bit.ly/1dfqOPB. Download this file and be sure you can load and interpret the file format in Matlab. Image files are named as x.y.png where x is the distance from the barrel to the camera (in meters) and y is meaningless.

2. Hand label appropriate regions in the training images with discrete color labels. For this project, we will be especially interested in regions containing the red barrel. In Matlab, you may find the "`roipoly`" function useful. If you are more ambitious, you could try to implement more automated ways of labeling images - perhaps by first doing an unsupervised image segmentation, or implementing an adaptive region flooding algorithm.

3. You may try different color spaces for your algorithms. For example, in Matlab, you can use the "`rgb2ycbcr`" to convert the RGB pixels in your labelled regions to the YCbCr color space. The issue of lighting condition could be resolved by choosing a good color space or run some low-level adaptation on the image.

4. Use a learning algorithm to partition the color space into appropriate class color regions. You should first to implement an approach discussed in class, but you can also try other machine learning approaches, i.e. decision trees, support vector machines, neural networks etc. Your algorithm should be able to robustly generalize to new images (hold out some of the training images to test this).

5. Once the red regions are identified, you can use shape statistics and other higher-level features to decide where the barrel is located in the images. Now use your algorithms to label and identify the coordinates of the centroid of the barrel in a new test image. You should also compute an estimate of the distance to the barrel. You'll be expected to quickly be able to classify and display your results on a new set of test images as follows:

```
dirstruct = dir('*.png');
for i = 1:length(dirstruct),
    % Current test image
    im = imread(dirstruct(i).name);
    % Your algorithm here!
    [x, y, d] = myAlgorithm(im);
    % Display results:
    hf = figure(1);
    image(im);
    hold on;
    plot(x, y, 'g+');
    title(sprintf('Barrel distance: %.1f m', d));
    % You may also want to plot and display other
    % diagnostic information such as the outlines
    % of connected regions, etc.
    hold off;
    pause;
end
```

6 You will upload to Canvas a zip file containing your code and a written description of your algorithm in PDF form (no need to include the training set in the zip file!). Use the naming convention "project1_[YourPennKey].zip".

7 For the presentation in class you are expected to bring your own laptop or use the classroom computer. The classroom computer has MATLAB installed but no compatibility is guaranteed. The projector has a VGA port and you may need a VGA adaptor for your laptop. During the presentation, you will be asked to present your algorithm and run your code on a set of test images. The test images will be released both online and on a USB flash disk prior to the presentations. Clearly presenting your approach and having good algorithm performance are equally important.