ESE650 Project 1: Color Segmentation

Due Date: **1/29/2015 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 Files > Project1 (Canvas) and also at https://upenn.box.com/s/pko7gokjk6nm6cqgbtdv1ilobfg9qnnu. These images will be available as an archived set of PNG color images. Download this file and be sure you and load and interpret the file format in Matlab.

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 want to use a different color space 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. Lighting invariance will be an issue, so you should think carefully about the best color space to use, and perhaps 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, etc. You need to make your algorithm so that it is 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 computations 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 written description of your algorithm in PDF form and a zip file of your code. Use the naming convention "project1_[pennkey].pdf" and "project1_[pennkey].zip" where [pennkey] is replaced by your pennkey. Steve would submit as project1_smcgill3.pdf, project1_smcgill3.zip

7 During the presentation in class you are expected to bring your own laptop or use the classroom computer. The projector has a VGA port and you may need a VGA adaptor for your laptop. You will be asked to run your code on the test set images which will be released both online and on a USB flash disk prior to the presentations