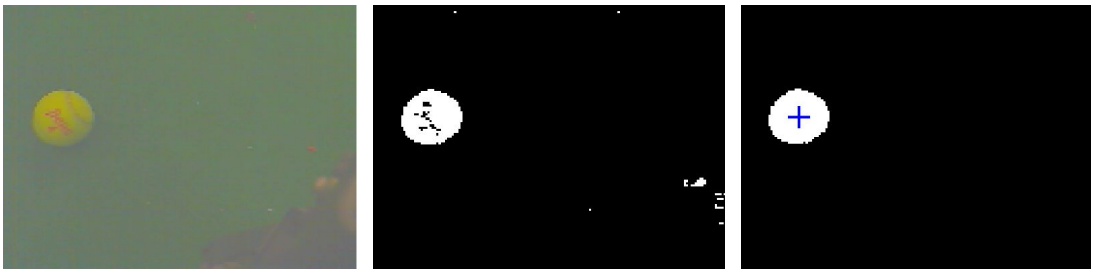
Programming Assignment Week I

Color Learning and Target Detection

1. Introduction

In this assignment you will be estimating a Gaussian model for ball color and developing a ball detection function based on the color model. The figure below depicts an example of an input image, a binary image based on the color model, and a segmented binary image after filtering.



Input RGB Image | Color-Based Binary Image | Segmented Ball Cluster

2. Instructions

Phase 1. Color Model Learning

1. Collect color samples of the YELLOW ball from the training images. You will create your own samples using the MATLAB function roipoly. (See *example_train.m*)
2. Visualize and observe the color distribution of the collected samples. You may convert the color space using functions such as *rgb2hsv*, *rgb2lab*, etc.
3. Determine the number of dimensions D for your model, and the number of Gaussians if you want to try GMMs.
4. Write a function to estimate your model parameters, that is, the mean and the (co)variance, from the sample data.
5. You may want to save the parameters for later use in Phase 2.

Phase 2. Target Color Detection

1. You will complete a function that takes a new image and your model parameters as input, and returns the color-segmented binary image and the center location of the ball. The signature of the function is given below.

```
function [segI, loc] = detectBall(I)
```

2. Load or hard code the estimated model parameters inside the function.

3. In order to segment the target color out from an input image, you will need to set a threshold probability. This is equivalent to having a uniform model of background colors.
4. Your function should return a binary image as a result of the segmentation.
5. In order to compute the location of the ball, you can consider applying some filtering on the binary image. Useful MATLAB Image Processing Toolbox functions include *bwconncomp*, and *regionprops* .
6. *example_test.m* is provided to help visualize your result.

Phase 3. Evaluation and submission

To submit your result to our server, you need to run the script *runeval* in your MATLAB command window. Please specify the path where the encrypted test data are located. A script will then evaluate your *detectBall* and generate output files (SubmissionBallSegmentation.mat and SubmissionBallLocation.mat) to be uploaded to the Coursera web UI. You may submit your result multiple times, and we will count only the highest score towards your grade.

(Last update 2016-05-07 15:02pm)