# ESE650 Learning in Robotics, 2018 Spring
# Project 1: Color Segmentation

Wudao Ling

## Introduction

In this project, I trained a hybrid GMM model for red color classification in RGB color space. Then I used connected component (shape) analysis to detect red barrels in images. At last, I applied linear regression to predict barrel distance from the barrel area.
We are only provided with training set, which contains 50 images and 51 barrels. The model performance can be validated by testing set, and the testing result will be attached at the end.

## 1 Hand Labeling

I used roipoly to hand-label 2 color classes. Besides RedBarrel, I also labeled a class called TrickyCases. During the implementation, I found several cases hard for my GMM model, for example, light yellow ceiling and dark red floor. Thus I address these cases by labeling them and emphasizing in Bayesian method. Hand-labeling on these 2 classes output boolean masks each training images, and pixels that don't belong to them are class Others (NOR operation).

## 2 Bayesian Method

Our aim is to learn the distribution:

$$\mathbf{P}(\mathbf{C}|\mathbf{X}) = \frac{\mathbf{P}(\mathbf{X}|\mathbf{C})\mathbf{P}(\mathbf{C})}{\mathbf{P}(\mathbf{X})}$$

Where $C = \{"RedBarrel", "TrickyCases", "Others"\}$. $X$ are 3 dimensional RGB color pixels.
Given an input color pixel vector $X$, we can find a class $c \in C$ that maximizes the posterior probability:

$$\mathbf{c} = \mathbf{argmax_c P(c|X)}$$

$P(X)$ will be the same for different classes as it is the sum of posterior, the prior $P(C)$ can be obtained from statistics in training set, and the likelihood $P(X|C)$ need GMM model to calculate.

# 3 Gaussian Mixtures Model

A few assumption is made for my GMM model for regularity and training speed concern. First, the weight of each cluster is fixed as 1/K. This means each cluster in GMM has equal weight on pixels likelihood. Second, the covariance is fixed as diagonal matrix, which means red, green and blue channel are independent of each other.
I use Expectation Maximization (EM) to optimize the model. EM algorithm is guaranteed to converge to a local minimum.
First randomly initialize $\mu, \Sigma$
E-Step: Estimate $r$

$$\mathbf{r}(k|x) = \frac{N(\mathbf{x}; \mu_k, \Sigma_k)}{\sum_{m=1}^{K} N(\mathbf{x}; \mu_m, \Sigma_m)}$$

M-Step: Estimate new $\mu_k$ and $\Sigma_k$

$$\mu_k = \frac{\sum_x \mathbf{r}(k|x)\mathbf{x}}{\sum_x \mathbf{r}(k|x)}$$

$$\Sigma_k = \frac{\sum_x \mathbf{r}(k|x) diag(\mathbf{x} - \mu_k) diag(\mathbf{x} - \mu_k)}{\sum_x \mathbf{r}(k|x)}$$

My model is called hybrid GMM model, because K for RedBarrel GMM is 3 and K for TrickyCases GMM is 10, while Others is a unimodel Gaussian. This design come from the 10 images cross-validation as well as inspection on color vector after EM algorithm. The testing result indicates, my color model works well in most cases and has a great performance on avoiding false positive, however it may fail to classify red color in dark environment. If given longer time, I can manage to train a GMM on Others class, and it will definitely improve the performance.

Figure 1: Remove False Negative

Figure 2: Detect red in dark environment

## 4 Object Detection

After pixels are classified, I conducted morphological transformation on segmented images. In specific, I did **morphologyEx** opening, which is erosion followed by dilation, to remove noises. For red objects such as bicycle and PR2 that has nearly the same color as red barrel but has large hollow in segmented images, the opening operation with 15*15 kernel effectively remove them.

Figure 3: Opening effect 1

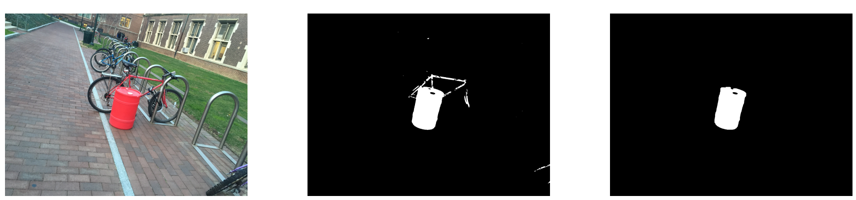

Figure 4: Opening effect 2

Then, I used OpenCV's **findContours** and **boundingRect** to to detect and bound red barrels. A qualified barrel need to pass area check and ratio check. The contour area should not be less than 700 or less than 1/5 of the largest contour area. Also, the bounding box of contour should have a height/width ratio within [1.15, 2.6].

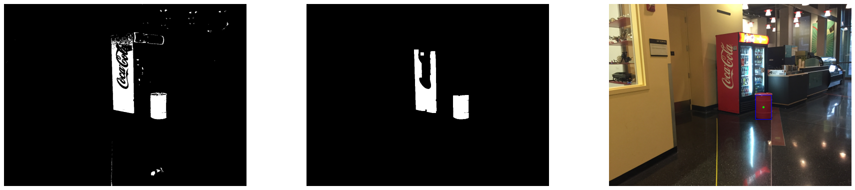

Figure 5: Successful Area/Ratio Check

However, testing result indicates that my shape analysis generally fail in 2 cases. First, when the barrel is very tilted, the vertical bounding box can fail the ratio check. A simple improvement is using **minAreaRect** (green box in Fig.6) to obtain a rotated bounding box. Second, when the barrel is occluded, I need an approach to integrate parts of barrel while not affect multiple barrel detection.

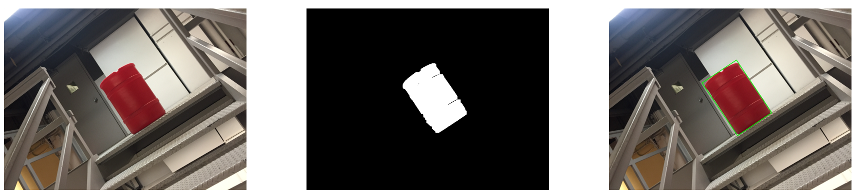

Figure 6: Failed Area/Ratio Check for vertical bounding box

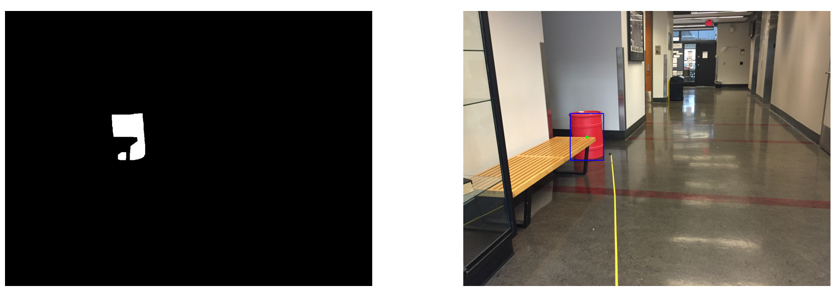

Figure 7: Successful Occlusion

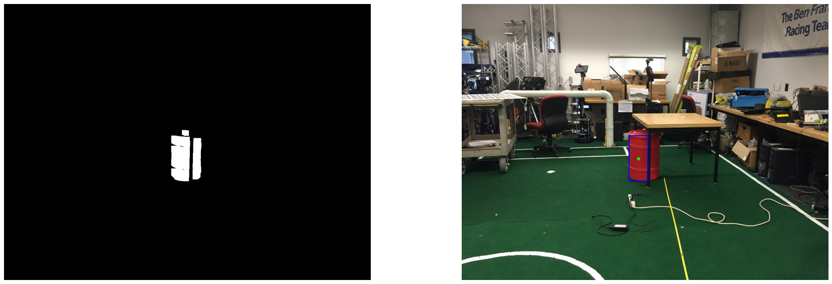

Figure 8: Partial Failed Occlusion

Figure 9: Failed Occlusion

## 5 Distance Estimation

I model the distance and reciprocal of area in training data using linear regression, and this model can be validated by pinhole camera model. For testing images, the barrel area will be the bounding box's area in Object Detection step, and distance will be predicted by linear model.

## 6 Test Results

ImageNo = [1], CentroidX = [660.], CentroidY = [503.5], Distance = [2.23921652], found 1 red barrel.
ImageNo = [2], find no red barrel.
ImageNo = [3], CentroidX = [698.5 552.5], CentroidY = [451.5 388.5], Distance = [2.79269867 4.48003384], found 2 red barrel.
ImageNo = [4], find no red barrel.
ImageNo = [5], CentroidX = [645.5], CentroidY = [467.], Distance = [6.76932323], found 1 red barrel.
ImageNo = [6], CentroidX = [632.5], CentroidY = [429.], Distance = [11.41489627], found 1 red barrel.
ImageNo = [7], find no red barrel.

ImageNo = [8], CentroidX = [740.], CentroidY = [482.], Distance = [12.7962294], found 1 red barrel.
ImageNo = [9], CentroidX = [673.], CentroidY = [428.5], Distance = [12.3322234], found 1 red barrel.
ImageNo = [10], CentroidX = [665.], CentroidY = [396.], Distance = [7.68696754], found 1 red barrel.