ECE5242 Project1: Color Segmentation

Sijia Gao

February 2019

Abstract

The aim of this project is to detect red barrels on RGB images under various backgrounds and illumination conditions. To achieve this, we first establish a Gaussian Mixture Model based color classifier to segment red pixels. Then we facilitate contour shape constraints to extract barrels from other red objects. Finally, we utilize a linear regression model to estimate the barrel's centroid and distance to the camera from barrel's width and height.

1 Introduction

Given a well defined target, how can we capture and locate it within huge amount of images and videos? This is a basic task if we want to reconstruct object's trajectories or build a large scale searching engine. In this project, we propose methods to detect specific red barrels on RGB images. We are given 50 training images to specify the targeted red barrel we are to detect. Over all 50 training images, the red barrels are placed under different backgrounds (e.g., hallway, lab, outdoor) and various lighting conditions. From these training images, we build a red barrel detector and implement the proposed algorithm on new test images.

The pipeline of our proposed method is illustrated as follows:

1. We first manually crop out red barrel pixels using MATLAB built in function *roipoly*.

2. Using all the cropped pixels, we build a Gaussian mixture model (GMM) based color classifier to select pixels that have similar color as the targeted barrel.

3. We then find contours from the segmented images and apply contour shape constraints to highlight barrels out of other segmented red pieces.

4. Finally, we implement linear regression to estimate barrel's centroid and distance to the camera.

The organization of this report is as follows: Sec.2 describes the design of the color classifier. A Gaussian mixture model (GMM) is utilized and expectation maximization algorithm is implemented for parameter estimation. Sec.3 discusses finding contours on the segmented image and applying contour shape constraints to extract barrels from other segmented objects. Sec.4 illustrates the implementation of linear regression to estimate centroid and distance to the camera of the detected barrel. Sec.5 gives performance of our proposed method on new test images and Sec.6 concludes the whole report.

2 Color classifier

Instead of conventional RGB, we transform pixel to LAB color space for enhanced conceptual purpose: uniform changes in LAB correspond to uniform changes in perceived color. In this section, we discuss how to build the color classifier using a Gaussian mixture model. From Bayes rule, the probability of a pixel being barrel red is written as

$$p(\text{barrel red=1}|\text{pixel}) = \frac{p(\text{barrel red} = 1)p(\text{pixel}|\text{barrel red} = 1)}{p(\text{pixel})} \tag{1}$$

Here, $p(\text{pixel}|\text{barrel red} = 1)$ is modeled via a Gaussian mixture model described later in this section. We assume even distribution on prior $p(\text{barrel red} = 1) = p(\text{barrel red} = 0)$ and $p(\text{pixel})$. As a result, the posterior probability is proportional to the likelihood.

The likelihood $p(\text{pixel}|\text{barrel red=1})$ in (1) is characterized by a Gaussian mixture model learned from the red barrel pixels cropped from the training images. The model is written as:

$$p(\text{pixel}|\text{barrel red} = 1) = \sum_{j=1}^{K} \mathcal{N}(\text{pixel}; \mu_j, \sigma_j)\pi_j \tag{2}$$

Here, K is the number of clusters and $\pi_j, j = 1, 2, \ldots, K$ with $\sum_{j=1}^{N} \pi_j = 1$ denotes the weight of the jth cluster. μ_j and σ_j are the mean and covariance of the jth cluster, respectively. In this report, we choose number of clusters $K = 3$ after cross validation on training images.

We use expectation maximization (EM) to estimate π_j, μ_j and σ_j in (2). Parameter update at each iteration is written as:
E step:

$$\gamma_j(\text{pixel}) = \frac{\mathcal{N}(\text{pixel}|\mu_j, \sigma_j)\pi_j}{\sum_{j=1}^{K} \mathcal{N}(\text{pixel}|\mu_j, \sigma_j)\pi_j}$$

M step:

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(\text{pixel}_n)\text{pixel}_n}{\sum_{n=1}^{N} \gamma_j(\text{pixel}_n)}$$

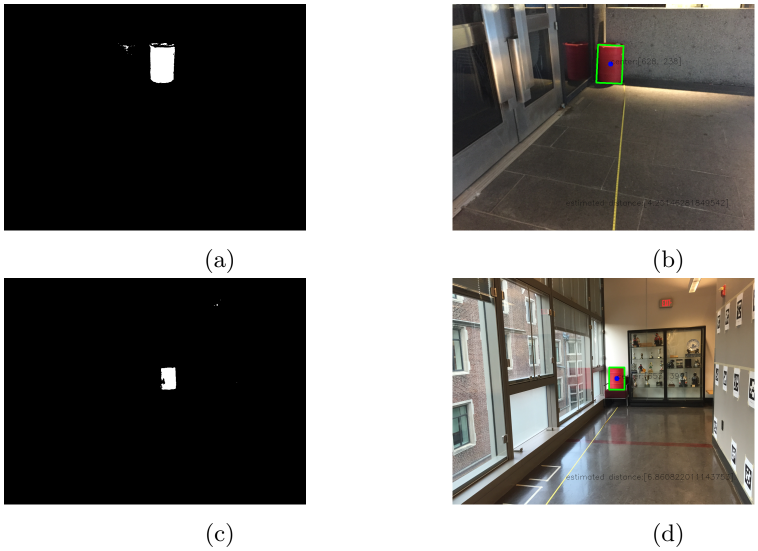


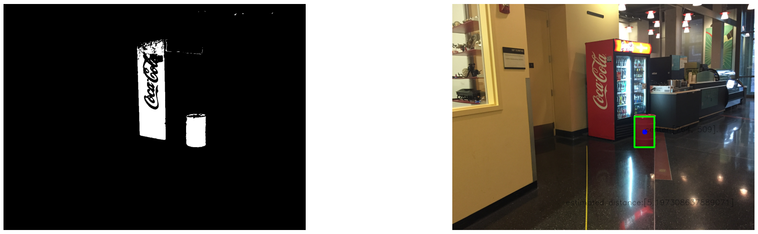
Figure 1: Combination of soft and hard threshold contribute to red barrel detection under dim light (a)(b) and differentiating red barrels from other red objects in (c)(d) where the barrel is placed on a dark red sofa. The first column shows the segmentation output of our color classifier and the second column illustrates the final output of our red barrel detector.

$$\sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(\text{pixel}_n)(\text{pixel}_n - \mu_j)(\text{pixel}_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(\text{pixel}_n)}$$

$$\pi_j = \frac{\sum_{n=1}^{N} \gamma_j(\text{pixel}_n)}{\sum_{j=1}^{K} \sum_{n=1}^{N} \gamma_j(\text{pixel}_n)}$$

EM algorithm is initialized via K-means from sklearn. We run EM for 20 iterations to get GMM parameters: $\mu_j, \sigma_j, \pi_j, j = 1, 2, \ldots, K$. The threshold to label a pixel as barrel red is computed as the maximum between average top 10% posteriors over all pixels and a hard threshold. Pixel with posterior above the threshold is transformed to white for contour searching in Sec.3. The combination of this top 10% posteriors (soft threshold) and a hard threshold contributes to differentiating barrels from other red objects and detecting barrels under dim lights; see Fig.1.

As another potential way to refine the color classifier specific to barrel red rather than the chair red, cone red and floor red appeared in training images, one may establish each red class a GMM model. However, this will make our color classifier sensitive to various illuminations. Therefore, we give up this approach and detect all objects with general "barrel red" and rely on contour shape constraints in Sec.3 to extract barrels from other red objects.



(a) (b)

Figure 2: Apply shape constraints to segmented objects. (a) Objects with barrel red color are segmented. A large red vendor machine is also segmented. (b) Shape constraints are utilized to extract barrel from the segmented red vendor machine.

3 Contours and shape constraints to extract barrels

After segmenting barrel red pixels in Sec.2, we use cv2.findContours and cv2.boundingRect to find contours and bounding box in rotated rectangles. Contours with area less than 1500 are removed. In addition, we notice that the ratio of width and height of the red barrel is between 0.5 and 0.9. We apply this width-height ratio constraint to differentiate barrels from other segmented red objects with substantial occupation; see example in Fig.2.

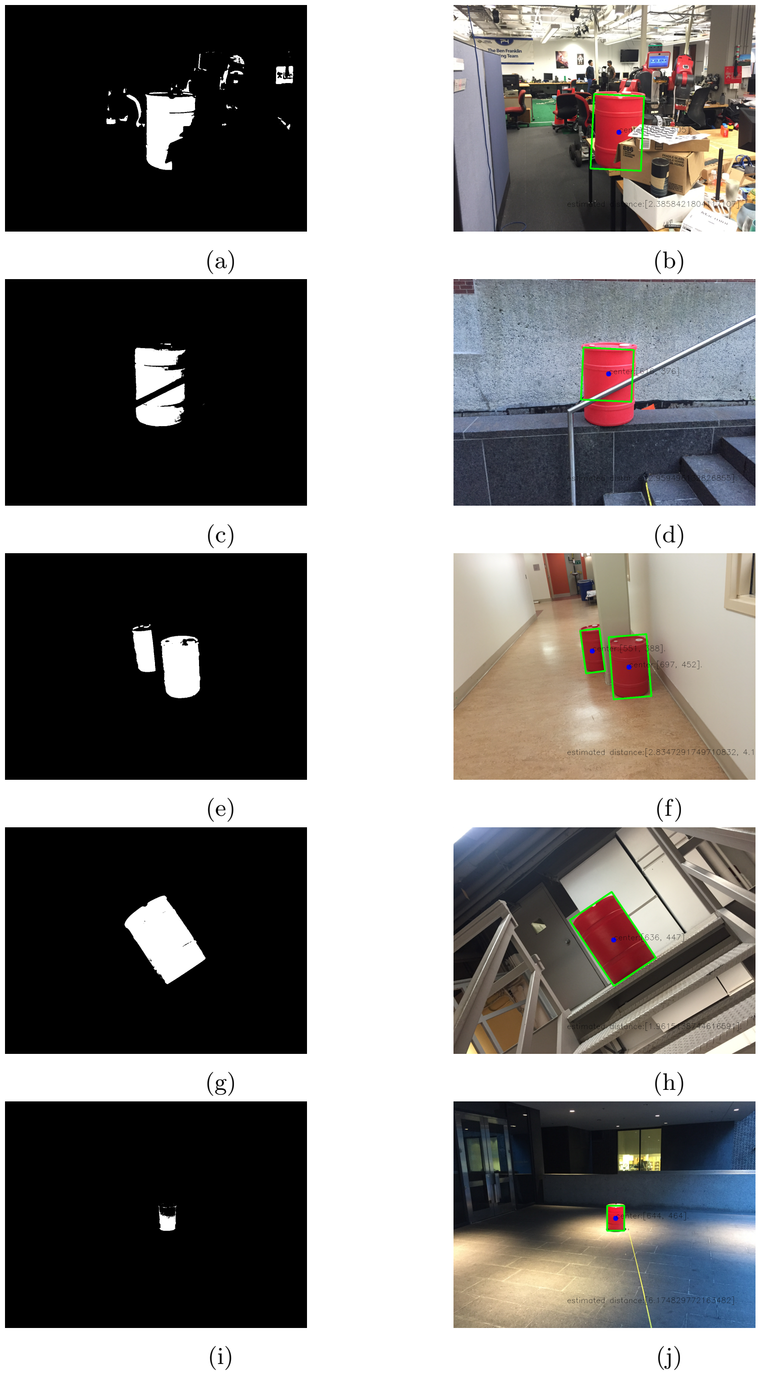
4 Centroid and distance estimator using linear regression

After finding the potential contour of red barrels in Sec.3, we use cv2.boundingRect to identify centroid, width (w) and height (h) of the rectangle structured contour and define them as the corresponding parameters of the detected red barrel.

Similar triangles in camera shooting implies that reciprocal of width ($1/w$) and height ($1/h$) of the barrel on the image is linearly proportional to its distance (d) to the camera. Therefore, we train a linear regression model that maps $[1/w, 1/h]$ to d. Therefore, given the estimated width (w) and height (h) of the barrel, we are capable to compute its distance to the camera.

5 Performance on test images

The performance of the proposed method on test images are illustrated in Fig.3 and Fig.4. First column is the segmented images and second column is the final output of the proposed red barrel detector with centroid and distance estimation. The results show that our method successfully detects all red barrels. In addition, for all images except the one in the second row in Fig.3, the contour of the barrel is nicely bounded by our algorithm without incorporating red objects close to it. However, when the barrel is partially blocked, such as the one in the second row in Fig.3, the method fails to outline the complete contour of the barrel.

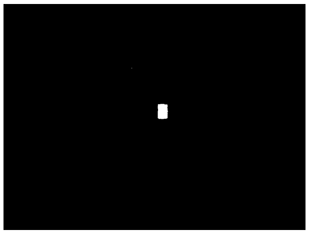


(a) (b) (c) (d) (e) (f) (g) (h) (i) (j)

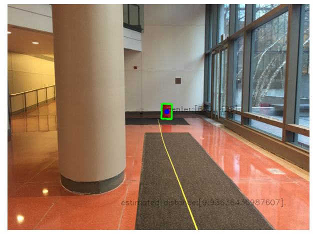
Figure 3: Performance of the proposed method on test images. First column is the segmented barrel red objects. Second column is the final output of the proposed red barrel detector. Centroids of the barrel and distance to the camera are also shown on the image.

6 Conclusion

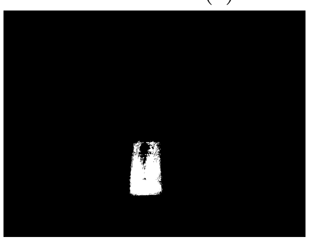
In this report, we established a detector for red barrels on RGB images under various backgrounds and lighting conditions. The pipeline of our method was first built a color classifier for barrel red objects, then found contours and applied shape constraints to extract barrels out of other segmented red objects. Finally, a linear regression was implemented to estimate distance of the barrel to the camera based on its width and height on image. The performance on new test images showed that our method successfully detected all red barrels and nicely outlined the barrel contours. Further improvement should consider selectively combining several contours to formulate a new one for the case when the barrel is partially blocked.



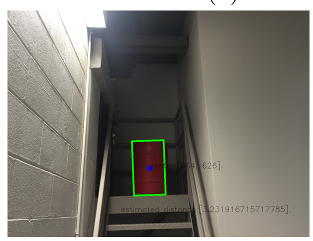
(a)



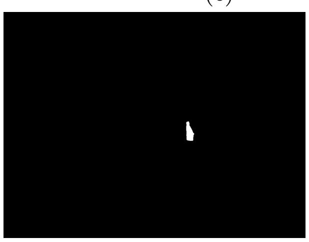
(b)



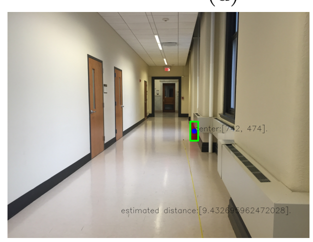
(c)



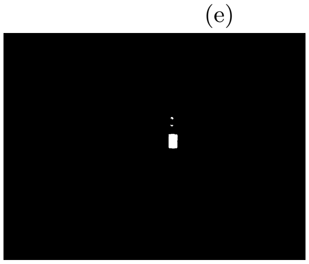
(d)



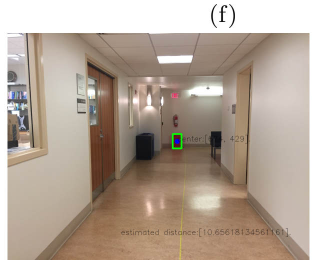
(e)



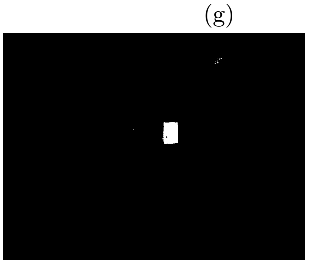
(f)



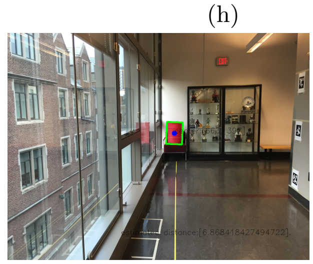
(g)



(h)



(i)



(j)

Figure 4: Performance of the proposed method on test images-Continue.