GMM Based Color Segmentation for Barrel Detection

Varun Gupta and Nikolay A. Atanasov

***Abstract*— Object detection is one of the most essential skill for robots to accomplish the most trivial of tasks which is obstacle avoidance. It is used for various other functionality, such as localization and mapping, path planning and interacting with environment. In this paper, a Gaussian Mixture Model Based Color segmentation technique is implemented that enables object detection when paired with the knowledge of object shape features. It is one of the most widely-used techniques to identify and detect objects.**

I. INTRODUCTION

This project was accomplished to perform barrel detection using color segmentation as a part of the course project in ESE 650: Learning in Robotics at University of Pennsylvania.

The process of partitioning a digital image into multiple segments is defined as image segmentation. Segmentation aims to divide an image into regions that can be more representative and easier to analyze. Such regions may correspond to individual surfaces, objects, or natural parts of objects. Typically color segmentation is the process used to locate objects and boundaries in images using color information. Furthermore, it can be defined as the process of labeling every pixel in an image, where all pixels having the same label share certain visual characteristics which in this case is a generic color.

Color image segmentation that is based on the color feature of image pixels assumes that homogeneous colors in the image correspond to separate clusters and hence meaningful objects in the image. As the segmentation results depend on the used color space, there is no single color space that can provide acceptable results for all kinds of images. In this paper, we work with the RGB color space and expect the trained color models to take care of brightness and illumination.

The color segmentation is achieved using a machine learning technique of Gaussian mixture models. The objective of this technique is to train different color models and represent them as a mixture of Gaussians in the 3 dimensional color space of red, green and blue channels. The technique has been tested to detect red colored barrels in different settings and for this purpose a set of 5 different color models have been trained. The next section will explain the entire procedure followed to achieve barrel detection.

II. PROCEDURE

A. Training Data

To achieve decent object detection, it is important that the training images incorporates all the factors that could hinder object detection, especially the factors such as varied illumination, reflection and shadows that could affect color detection and indirectly affect object detection; or occlusion that could directly affect object detection. The training images in this project have been taken by a camera at different locations with varying brightness. Also the distance to the object from the camera has been recorded.

The training images helped decide the minimum number of color models to be trained to achieve decent color segmentation. Five different color classes namely; red, black, brown, yellow and red but not barrel; were chosen. The red but not barrel was chosen because a lot of images had red colored objects such as ball, a robotic arm, or a vending machine which were red but not barrel. All these cases have been shown in the results to justify the need for this class of color.

The RGB data for each color class was obtained using polygonal regions of interest traced manually. An automatic method for obtaining the training data is the GrabCut segmentation technique[1] which does not require user intervention even for training. However to study the influence of choice of training data, a manual approach was chosen. A total of 50 training images were used and the regions of interest traced in each image resulted in a lot of RGB data, often repetitive, for each color class. Unique data points were extracted so that the clustering algorithm would not be biased towards any particular data point. To construct a model to fit each class, a set of randomly sampled data was taken from this unique data set in order to speed up the learning process. Roughly 10,000 RGB values were taken to train red, and 2,500 for all the other colors.

B. Clustering Algorithm

The next step is to construct a cluster in the 3 dimensional RGB space for each color. Using one cluster would ideally work well for color segmentation but not for identification of objects because objects that are red but not barrel would have a small set of points that are within the red cluster, and therefore will be classified as red and treated as a pixel belonging to barrel despite being something else. But if multiple clusters were used, these few data points belonging to red but not barrel would create a separate cluster of their own amidst the red and barrel. Also, since the RGB values depended on other factors such as illumination, there are pixels that belong to barrel but appear very similar to black. To incorporate these, a soft probabilistic clustering technique called the Gaussian Mixture Model was chosen.

Gaussian mixture models assumes that a data is sampled from a multiple number of Gaussian distributions. In this project, a mixture of 3 different Gaussian clusters were constructed for each color class in order to account for the three general lighting conditions, that is, the dark, normal and bright settings. The ideal number of clusters could be obtained using a silhouette analysis of the training data but we will not discuss about it in this paper. A GMM model was constructed using the Expectation Maximization Algorithm which optimizes the marginal likelihood of the data (likelihood with hidden variables summed out). It is an iterative algorithm, alternating between two steps, the E and M steps, which correspond to estimating hidden variables given the model and then estimating the model given the hidden variable estimates. The algorithm is mathematically depicted below.

ALGORITHM

Initialize $\alpha_k^{(i)}$, $\mu_k^{(i)}$ and $\Sigma_k^{(i)}$

$(E - Step)$

$$r^{(i)}(k \mid x) = \frac{\alpha_k^{(i)} \phi(x; \mu_k^{(i)}, \Sigma_k^{(i)})}{\sum_{m=1}^{k} \alpha_m^{(i)} \phi(x; \mu_m^{(i)}, \Sigma_m^{(i)})}$$

$(M - Step)$

$$\mu_k^{(i+1)} = \frac{\sum_x r^{(i)}(k \mid x) x}{\sum_x r^{(i)}(k \mid x)}$$

$$\Sigma_k^{(i+1)} = \frac{\sum_x r^{(i)}(k \mid x)(x - \mu_k^{(i+1)})(x - \mu_k^{(i+1)})^T}{\sum_x r^{(i)}(k \mid x)}$$

$$\alpha_k^{(i+1)} = \frac{1}{\mid D_\alpha \mid} \sum_x r^{(i)}(k \mid x)$$

In the above algorithm $\phi(x; \mu_k^{(i)}, \Sigma_k^{(i)})$ denotes the likelihood of point x being sampled from the Gaussian cluster k with mean μ_k and covariance Σ_k at the i^{th} iteration. The α_k denotes the probability of the cluster in the class model and marginalizing over all the clusters yields the membership probability $r(k \mid x)$.

The EM algorithm may converge to a sub-optimal solution depending on the initialization of the parameters. So to obtain a decent GMM model, the means and the co-variances are initialized using the K-Means algorithm which is another clustering algorithm. The weights for each cluster is initialized to be equal, that is, $1/3$ in case of 3 clusters.

C. Training the camera parameters

For a robot, after the detection of the obstacle, the objective becomes to identify the distance of the obstacle from itself. To be able to estimate the distance, the robot needs to have knowledge of the camera parameters. In case the knowledge of the camera is unknown, some training images with knowledge of distance in the world frame can be used to estimate all the camera parameters using projective geometry. In this paper, we use a least squares estimate to obtain just one parameter which is sufficient to get an approximate measure of the distance of the obstacle. We call it the focal length though it is not exactly the focal length of the camera but is constant and proportional to focal length. This parameter can be mathematically expressed as

$$f_x = \frac{w}{W} \times d = f \times s_x$$

$$f_y = \frac{h}{H} \times d = f \times s_y$$

f_x and f_y denote the focal length along the orthogonal axes in the plane of the camera. The w and h denote the width and height of the barrel in the projected plane. The W and H denote the true width and height of the barrel. The s_x and s_y denote the scale factors that appear because of the dimensions of the receptive cells in the camera. To eliminate the computation of these variables and simplify the estimation of the distance, we assume that the scale factors are proportional to the frame width and height. From this, we gather that $\frac{w}{W_f} \times \frac{d}{f} = W$ is a constant. Similarly $\frac{h}{H_f} \times \frac{d}{f} = H$ is also a constant, where W_f and H_f denote the width and height of the frame. But since we approximated for the scale factors, we cannot assume any of the equations above to be explicitly true. So we create another parameter which is a function of the true width and height of the barrel and enforce it to be a constant. This parameter can be defined as:

$$\begin{aligned} C &= \sqrt{W^2 + H^2} \\ &= \sqrt{\left(\left(\frac{w}{W_f} \right)^2 + \left(\frac{h}{H_f} \right)^2 \right) \times \left(\frac{d}{f} \right)^2} \end{aligned}$$

From these expressions, we obtain the camera parameter called scaled focal length s_f that we define to be $C \times f$.

$$s_f = \sqrt{\left(\left(\frac{w}{W_f} \right)^2 + \left(\frac{h}{H_f} \right)^2 \right)} \times d$$

To estimate s_f, we do least squares error minimization and find that s_f is the mean of the term on the right which is a function of the width, height (in the image) and distance of the object in the training phase. This equation is then inverted during the test phase to obtain the distance in the world frame. This method produces an approximate estimate of the distance but the results obtained are reasonable.

D. Object Detection

After obtaining the color clustering models and the camera parameters, the main task of object detection has to be achieved which still remains a challenging task despite the heavy pre-processing based on color segmentation. At this stage, a binary mask is constructed where the white pixels indicate color Red and the black pixels indicate every other color model. Then OpenCV's contour detector is applied on this binary mask to obtain the contour that has maximum area and a minimum area bounding box is fitted to the contour. This handles detection of inclined bins as well. It is assumed that the color segmentation model is perfect and therefore do not apply shape filters. Shape filters based on aspect ratio or the dimensions are very specific to object and does not work for a general object detection task. However dilation is used to ensure minimally occluded objects get fully detected. This method doesn't work well for highly occluded images especially if the occluding object completely separates the object into distinct red patches. This implementation does not detect multiple barrels either.

III. PERFORMANCE

The performance of the algorithm for both training and test images have been shown in the paper to demonstrate the effectiveness and the drawbacks of this color segmentation algorithm. Fig1 shows the bounding box around the barrel for various training images. The barrel is clearly detected despite occlusion as can be seen in Fig1(c) or presence of other objects of similar color as can be seen in Fig1(d), Fig2(f) and Fig2(g). The algorithm is able to detect inclined barrels as can be seen in Fig1(a) as well as distant barrels as can be seen in Fig1(b). The bounding box around the barrels is perfect even in the presence of shadows directly in contact with the red of the barrel. This can be seen in Fig2(e) and Fig2(g). However in Fig2(h), the vending machine which has a color extremely close to the red of the barrel gets detected as a barrel. It is for these cases that shape filters are required to distinguish a barrel from other objects in the environment.

The estimated distances have been reported and compared for the training images in Table I. The estimates are nearly accurate except for Fig2h where the vending machine gets detected instead of the barrel. In almost every other case, the errors in the distance estimates are in the range 0.01 to 0.5 approximately.

The algorithm was tested on 10 of the test images having a single barrel in varied environments. Fig3 shows the performance of the color segmentation algorithm on 3 of the test images having varied lighting and/or inclined barrels. The barrel is detected clearly in these cases and the distance estimate according to TableII appears to be as expected. Fig4 shows that the algorithm performs decently on distant barrels. The distance estimates appear to be reasonable in these cases as well.

The algorithm did not perform well in 5 of the test images. In 3 of those, only a portion of the barrel was detected due to occlusion as can be seen in Fig5. The distance estimate is reasonable for one of these cases but exorbitantly high or low for the other two cases. Fig6 shows cases where barrel is detected. But the red of the barrel is not clearly distinguishable from the red of the coinciding objects or the reflections of the barrel itself. This may be because of insufficient training of the red-but-not-barrel color class or over-fitting of the red class due to higher sampling from particular RGB range. The distance estimates however are not as terrible because the bounding box encloses the barrel and has aspect ratio very close to that of the barrel in the frame. This may not be the case for every image.

TABLE I
DISTANCE ESTIMATION FOR TRAINING IMAGES

Image	Actual Distance	Estimated Distance
a	2	2.283675
b	14	13.843785
c	2	2.040400
d	2	1.785714
e	4	3.954432
f	5	4.696898
g	6	6.326621
h	5	1.717614

TABLE II
DISTANCE ESTIMATION DURING GOOD DETECTION

Image	Estimated Distance
001	4.080800
004	2.002284
005	7.109848
007	8.573919
010	9.261644

TABLE III
DISTANCE ESTIMATION DURING POOR DETECTION

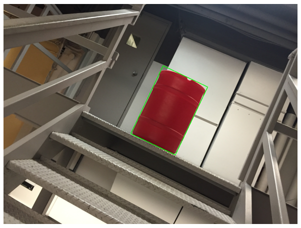
Image	Estimated Distance
002	16.146999
003	1.527188
006	0.971540
008	4.815822
009	6.743281

IV. CONCLUSIONS

In this project, a Gaussian mixture model based color segmentation technique was implemented to obtain a binary mask from which a red colored barrel was extracted using contour detection. A method was implemented to estimate the distance of the barrel from the camera and the results obtained were reasonable when the detection of the barrel was good.

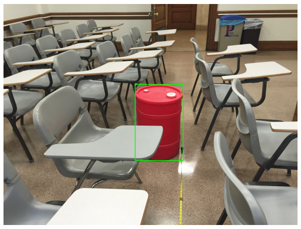
ACKNOWLEDGMENT

Thanks to University of Pennsylvania for providing me an opportunity to work on interesting projects by conducting a



(a) a

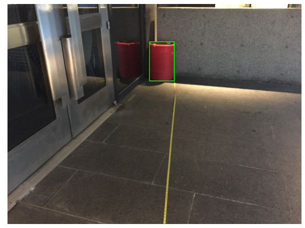
(b) b



(c) c

(d) d

Fig. 1. Performance on Training images



(a) e

(b) f

(c) g

(d) h

Fig. 2. Performance on Training images

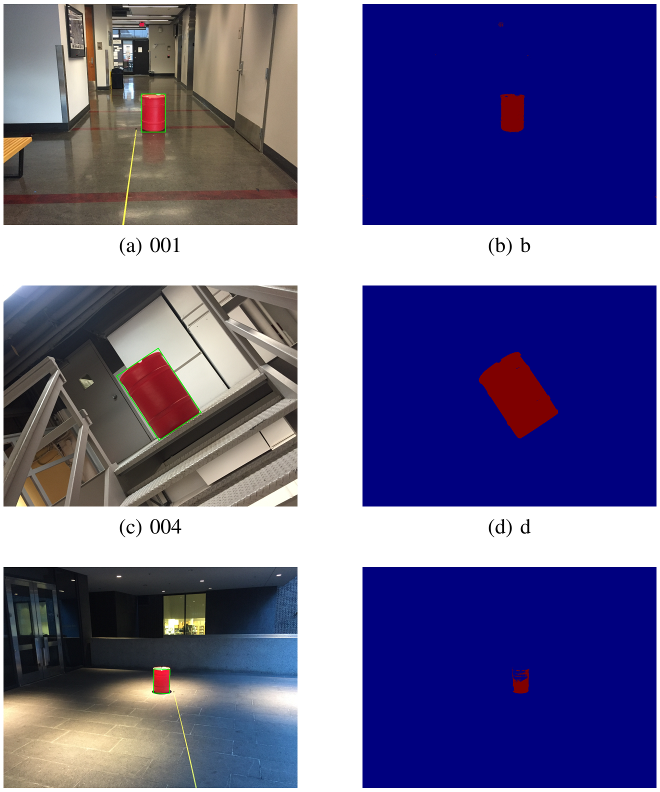
course on Learning in Robotics. Special thanks to Professor Nikolay A. Atanasov and the Teaching Assistants for their assistance during the project.

References

[1] Khattab, Dina et al. Color Image Segmentation Based on Different Color Space Models Using Automatic GrabCut. The Scientific World Journal 2014 (2014): 126025. PMC. Web. 1 Feb. 2017.

[2] Nikolay A. Atanasov. Lecture Notes for ESE 650: Learning in Robotics, 2017.

[3] Lyle Ungar. Lecture articles for CIS 520: Machine Learning, 2016.

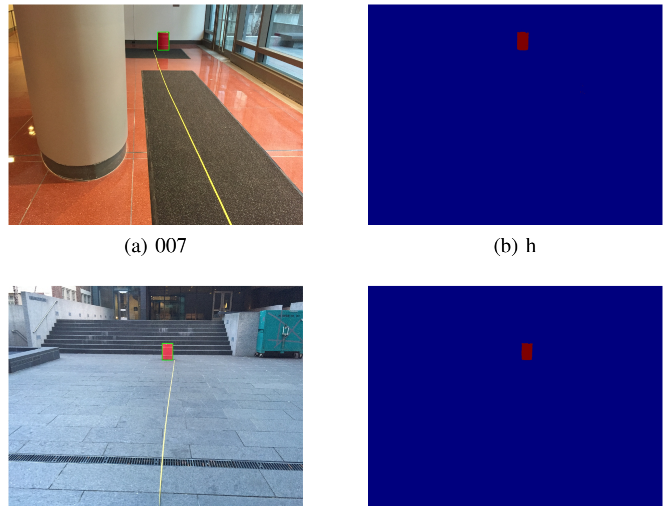


(a) 001 (b) b

(c) 004 (d) d

(e) 005 (f) f

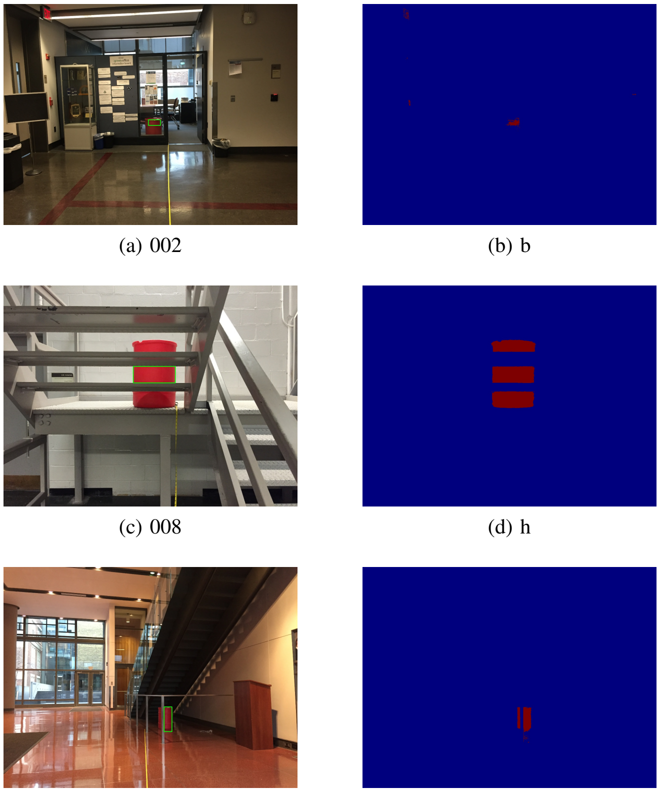
Fig. 3. Performance on Test images: Good results



(a) 007 (b) h

(c) 010 (d) j

Fig. 4. Performance on Test images: Good Results

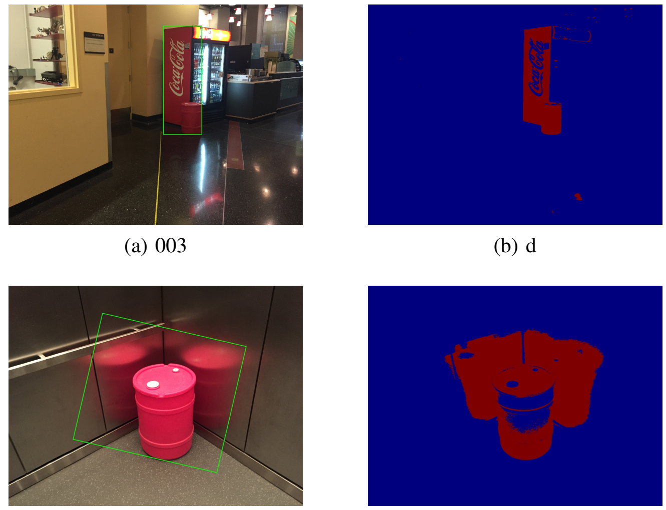


(a) 002 (b) b

(c) 008 (d) h

(e) 009 (f) j

Fig. 5. Performance on Test images: Bad Results



(a) 003 (b) d

(c) 006 (d) f

Fig. 6. Performance on Test images: Bad Results