Project 3 Report

Ashwin Goyal; UID: 115526297

April 23, 2018

Introduction

A buoy is a distinctively shaped and colored floating device, anchored to the bottom, for designating moorings, navigable channels, or obstructions in a water body. It is sometimes used as underwater markings for navigation. The data set for the project is a camera view of an underwater vehicle with three distinctly colored buoys in surrounding. The goal is to segment and detect the buoys in the given video sequence.

Methodology

The buoys are distinctly colored and shaped. So, one approach to segment and detect the buoys is to use color. But, because of noise and changing light intensities, it is difficult to segment the buoys using vanilla color threshold based techniques. Hence, Gaussian Models have been used to learn the color distribution, and use the model to segment and detect the buoys.

Part 0. Data Preparation and Vanilla Approach

The first part of this project is to extract and save the frames from the video sequence provided in the data set. The provided video has a frame rate of 5 frames per second. So, a total of 200 frames have been extracted from the video sequence. These frames have been divided into two sets, Training Set and Test Set, such that the training frames are used to extract model parameters using Expectation-Maximization algorithm and the test frames are used to evaluate the performance of the model. Every 10^{th} frame from the video sequence has been saved as a training frame so that the variances in the color intensities are recorded in the model but at the same time, the model is not over-fitted. These training frames have been further cropped using **roipoly** function of MATLAB. This step enables the algorithm to record the color intensities corresponding to each buoy, which is required to get the respective color models. Frame 21 of the video sequence along with the binary images corresponding to the location of green buoy, red buoy and yellow buoy in the frame is shown in Figure 1, Figure 2, Figure 3, and Figure 4 respectively.

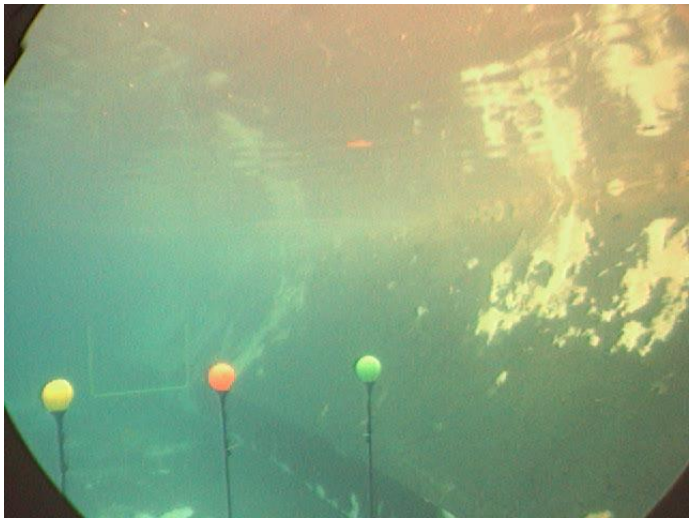

Figure 1: Frame 21 of the given Video Sequence

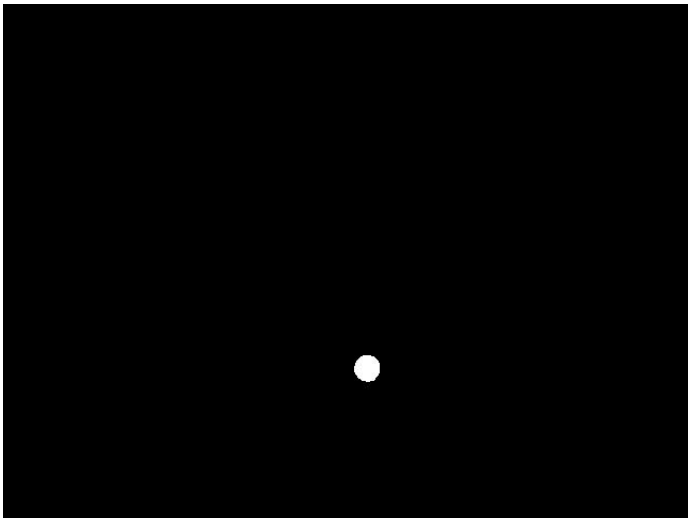

Figure 2: Binary Image with location of Green Buoy

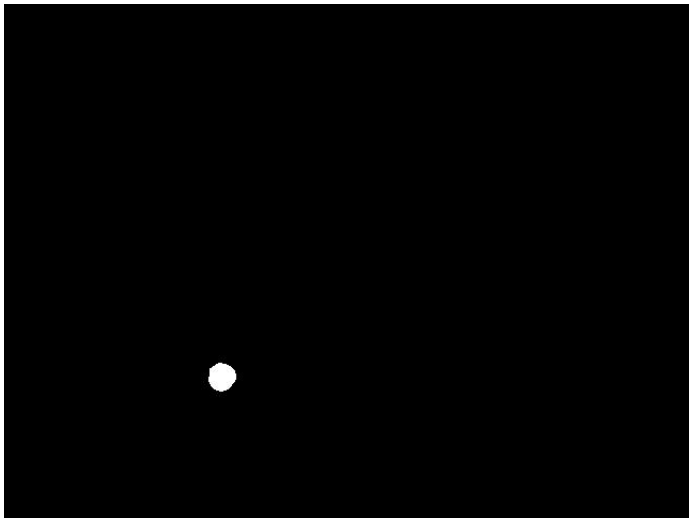

Figure 3: Binary Image with location of Red Buoy

Figure 4: Binary Image with location of Yellow Buoy

After preparing the training set, the next step is to compute average histogram for every color channel corresponding to each buoy. This step helps to visualize the average color distribution for each buoy. The algorithm to compute average histogram simply uses **imhist** function of MATLAB to get the histogram for every buoy in a single image. This data is stored for all the frames. In the end, the algorithm simply takes a weighted average of the computed histograms to get the final output. The average histogram computed for green buoy, red buoy, and yellow buoy is shown in Figure 5, Figure 6, and Figure 7 respectively. It should be noted that although the histograms are visualized only for **RGB** color space, but the algorithm saves the data for color distribution as well as average color histograms for five different color spaces:

1. **RGB.** Red – Green – Blue,
2. **HSV.** Hue – Saturation – Value,
3. **YIQ.** Luminance – Hue – Saturation,
4. **YCbCr.** Luminance – Blue-Reference – Red-Reference, and
5. **L*a*b.** Luminance – Green_Red – Blue_Yellow

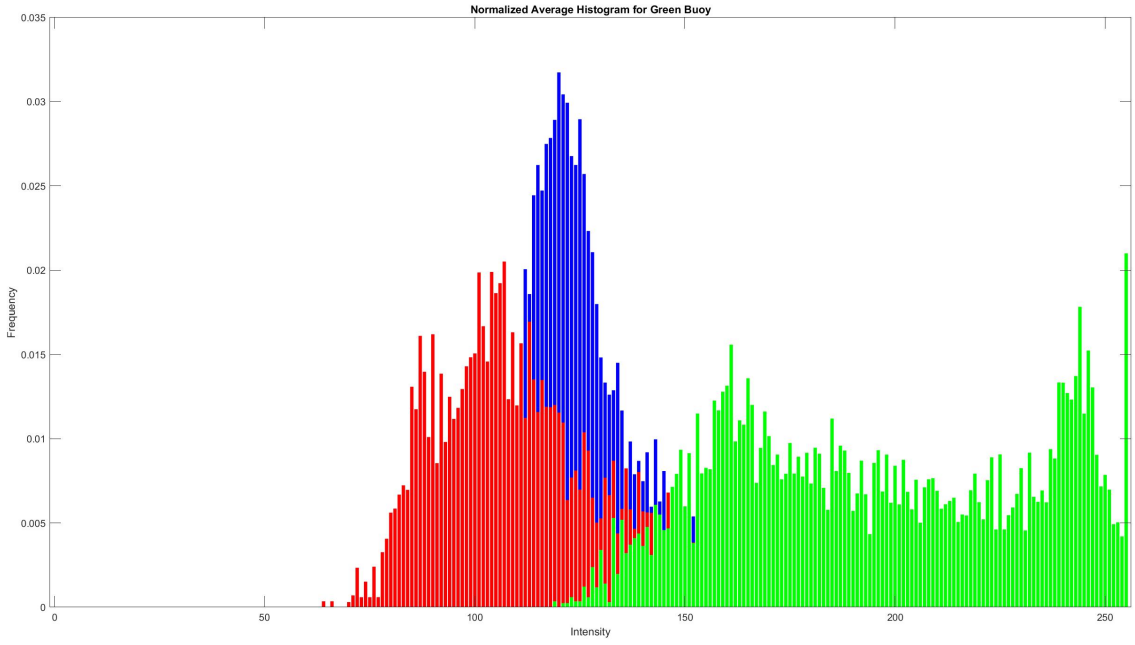

Figure 5: Histogram for Green Buoy

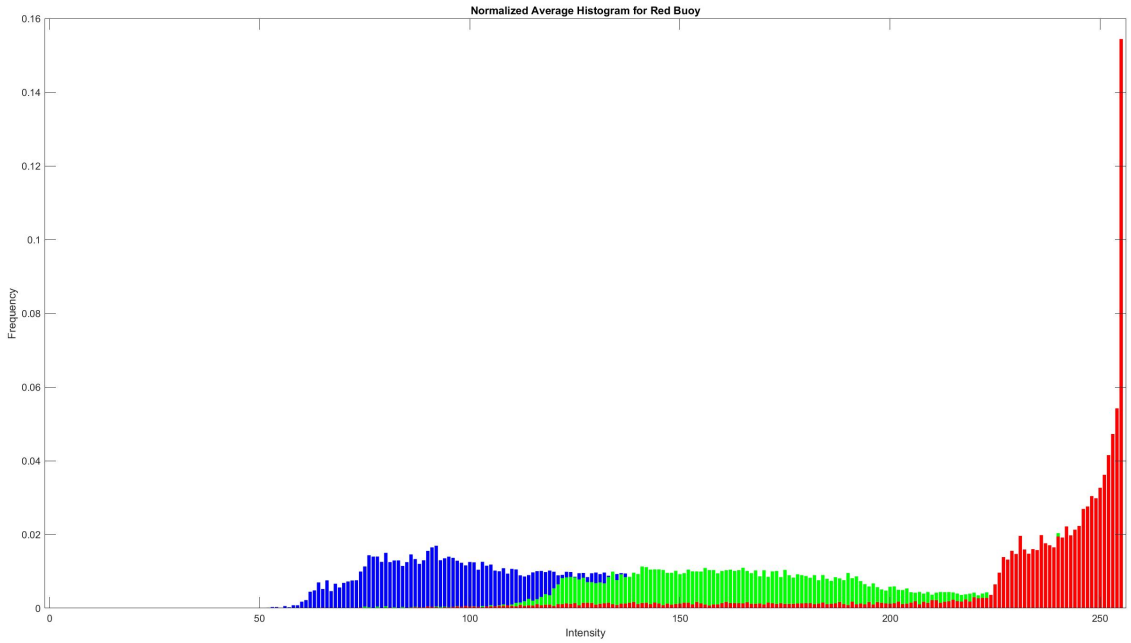

Figure 6: Histogram for Red Buoy

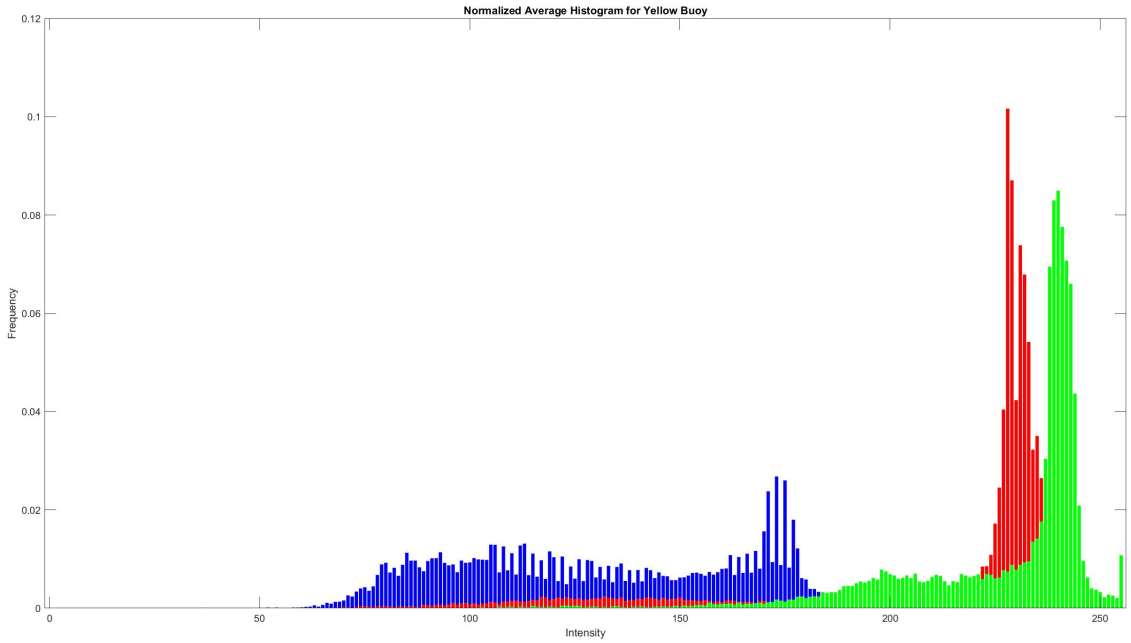

Figure 7: Histogram for Yellow Buoy

After the average histograms have been computed, this data is used to generate a model for the color distribution corresponding to each buoy using a 1-D Gaussian. From Figure 5, we can observe that green color channel for the green buoy is the most dominant. So, for green buoy, green color plane is used to generate a 1-D Gaussian (Figure 8). Similarly, from Figure 6, it can be noted that red color channel is the most dominant. So, for red buoy, red color plane is used to generate a 1-D Gaussian (Figure 9). From Figure 7, it can be observed that green color channel and red color channel are equally dominant. So, in case of yellow buoy, instead of using a single color plane, a mean of red color plane and green color plane is used to model the 1-D Gaussian (Figure 10). These Gaussian models are used to segment and detect the buoys in all the frames. Not only that, each detected buoy is encircled with a contour of the same color. Finally, a video is generated using the segmented frames as input. An example frame is shown in Figure 11.

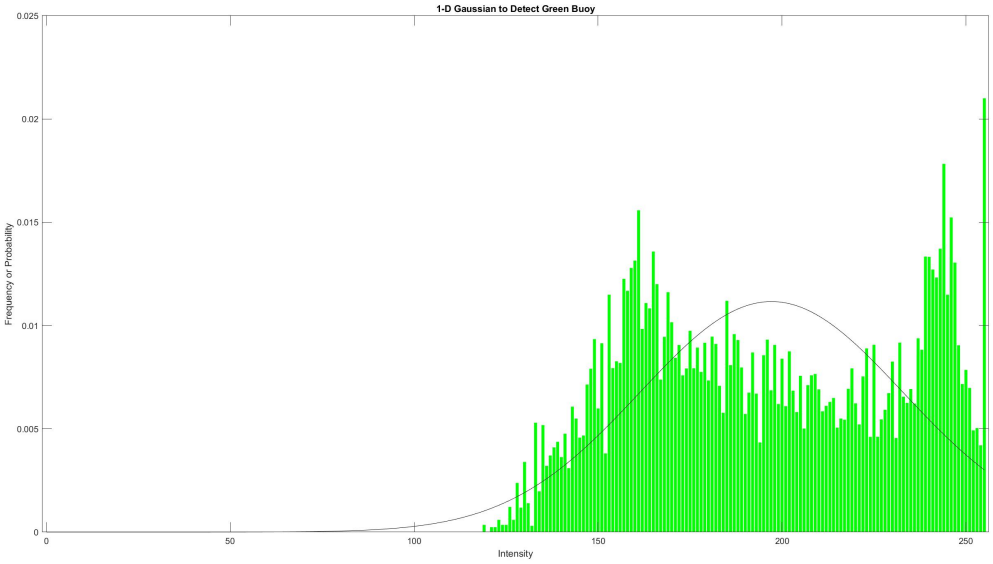

Figure 8: 1-D Gaussian used for Green Buoy

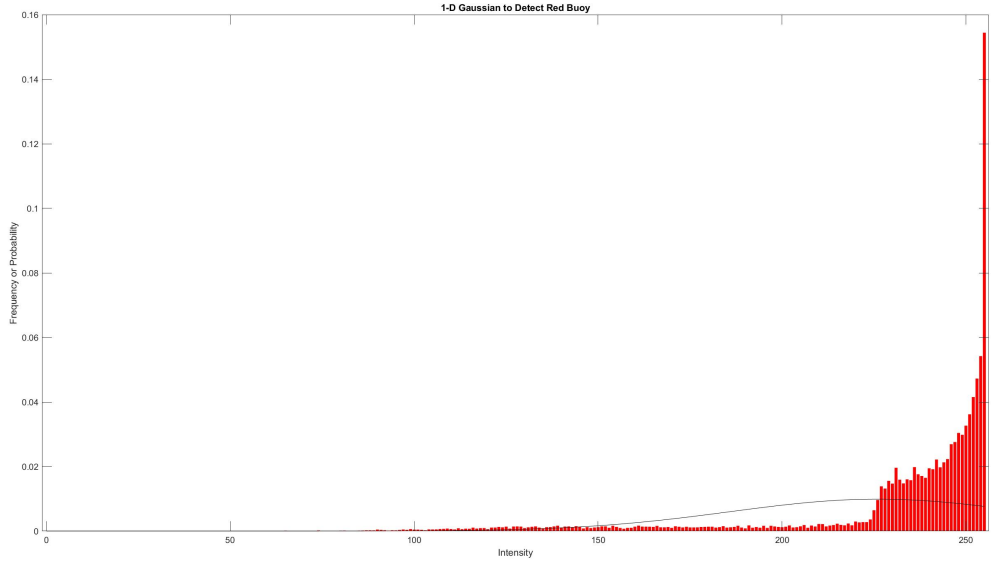

Figure 9: 1-D Gaussian used for Red Buoy

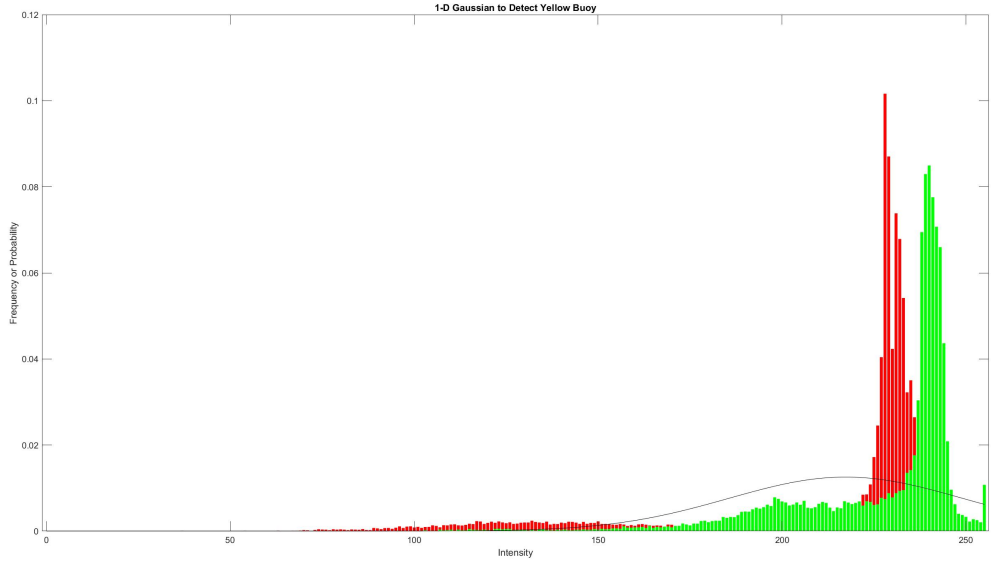

Figure 10: 1-D Gaussian used for Yellow Buoy

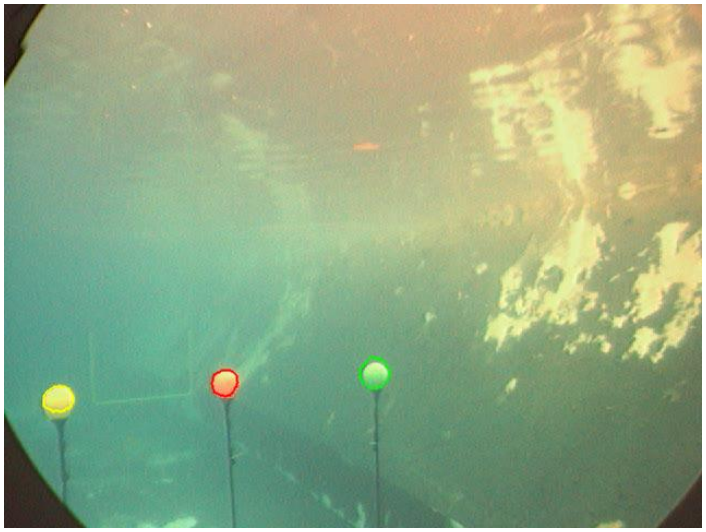

Figure 11: Segmented Frame using 1-D Gaussian

This simple segmentation model segments and detects the buoys to a good accuracy because the algorithm for segment-1D first creates binary images using the Gaussian models and then checks for overlap. This step is necessary because the yellow buoy is read by both red and yellow Gaussian models. Similarly, the red buoy is read by both yellow and red Gaussian models. So, in order to remove these overlaps, a small occupancy grid has been created which states whether a particular set of the three buoys is possible or not. This step improved the detection multi-folds. In fact, the detection from this simple 1-D Gaussian model is better than that done by models generated using Expectation-Maximization.

Part 1. Gaussian Mixture Models and Maximum Likelihood Algorithm

The next part of this project is to implement Expectation-Maximization to predict (model) data into a defined number of Gaussian.

A Gaussian mixture model is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. The main difficulty in learning Gaussian mixture models from unlabeled data is that one usually doesn't know which points came from which latent component. Expectation-maximization is a well-founded statistical algorithm to get around this problem by an iterative process. First one assumes random components and computes for each point a probability of being generated by each component of the model. Then, one tweaks the parameters to maximize the likelihood of the data given those assignments. Repeating this process is guaranteed to always converge to a local optimum. The steps for Expectation-Maximization in context of Gaussian Mixture Models are:

1. Initialize the means μ_k, co-variances Σ_k and mixing coefficients π_k, and evaluate the initial value of the log likelihood.

2. **E-step.** Evaluate the responsibilities using the current parameter values.

$$\gamma(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$$

   where

$$N(x|\mu, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} exp\{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T\}$$

   Here, D refers to the dimension of the data.

3. **M-step.** Re-estimate the parameters using the current responsibilities.

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})^T (x_n - \mu_k^{new})$$

$$\pi_k^{new} = \frac{N_k}{N}$$

where

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

Here, N refers to the number of samples of the data.

4. Evaluate the log likelihood and check for convergence of the log likelihood.

$$ln(p(X|\mu, \Sigma, \pi)) = \sum_{n=1}^{N} ln\{\sum_{k=1}^{K} \pi_k N(x_n|\mu_k, \Sigma_k)\}$$

Here, K refers to the number of Gaussian used to model the data.
If the convergence criterion is not satisfied return to step 2.

Using the steps listed above, an algorithm implementing Expectation-Maximization for Gaussian Mixture Models has been implemented. The algorithm takes the data and the number of Gaussian to be used as input and returns the parameters for each Gaussian. The algorithm has been tested to model the data generated using 3 1-D Gaussian using 3 Gaussian and the result is shown in Figure 12. The same data has then been modeled using 4 Gaussian and the result is shown in Figure 13.

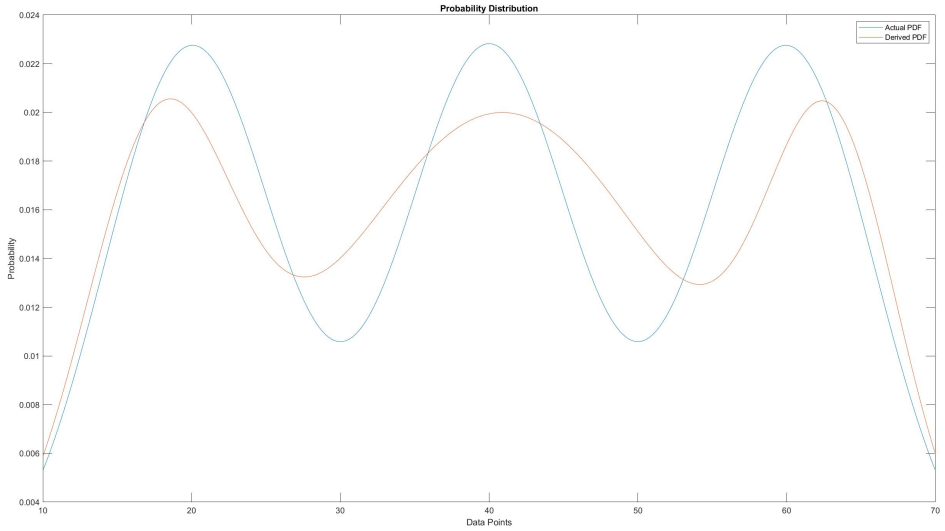

Figure 12: 1-D Gaussian with K = 3

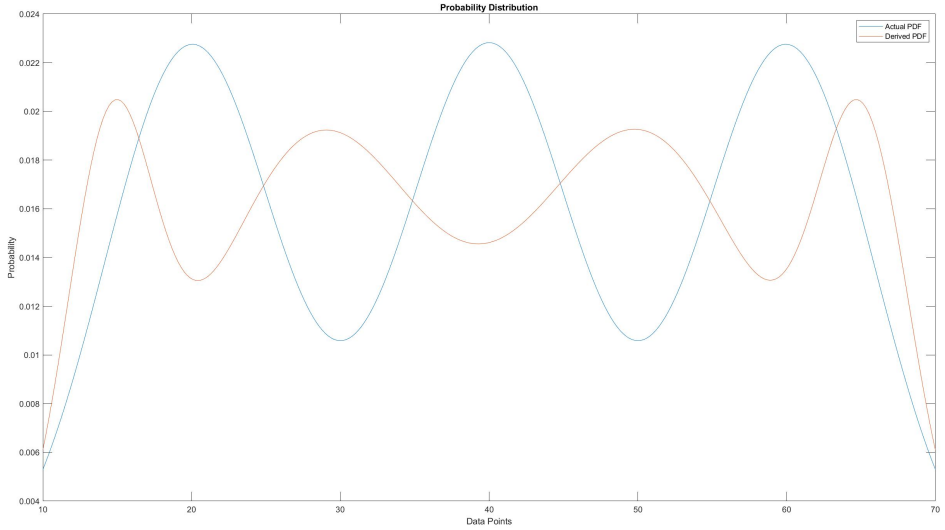

Figure 13: 1-D Gaussian with K = 4

The difference in the results as observed from Figure 12 and Figure 13 is due to the number of Gaussian used to represent the data. When the data is modeled using 3 Gaussian, the output almost matches the input, that is, the peaks occur around the same statistical point. But when the same data is modeled using 4 Gaussian, two of the peaks occur at the troughs of the input data. Such a behaviour is observed because the Expectation-Maximization algorithm finds the local optimum and in case of 4 Gaussian, two of these optimum occur at the local minima of the input data.

The algorithm has been checked for N-Dimensional data by using the Red-Blue color distribution of the yellow buoy. The data has been modeled using 5 Gaussian and the result is shown in Figure 14.

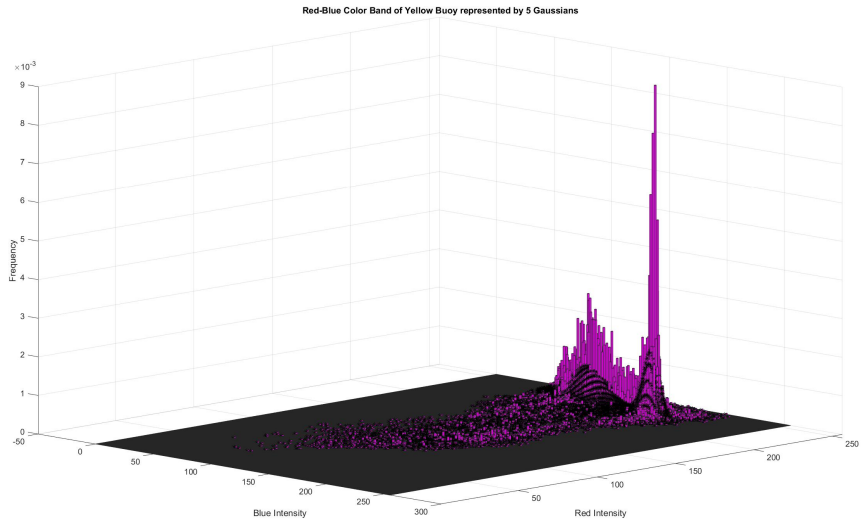

Figure 14: 2-D Gaussian with K = 5

Part 2. Color Model Learning

This part of the project focuses on the use of the clustering concept developed in Part 1 for our goal of color segmentation. We first visualize the color histograms for each buoy and based on this data, we decide upon the number of Gaussian to be used to model a particular buoy. We also decide the dimension of the Gaussian to be used. It was much difficult to just decide upon the desired model looking at the color histograms. So, instead, I applied all the possible models with K ranging from 1 to 5 on the images and decided upon the model based upon the quality of output per model. The final models used for green buoy, red buoy and yellow buoy are shown in Figure 15, Figure 16, and Figure 17 respectively.

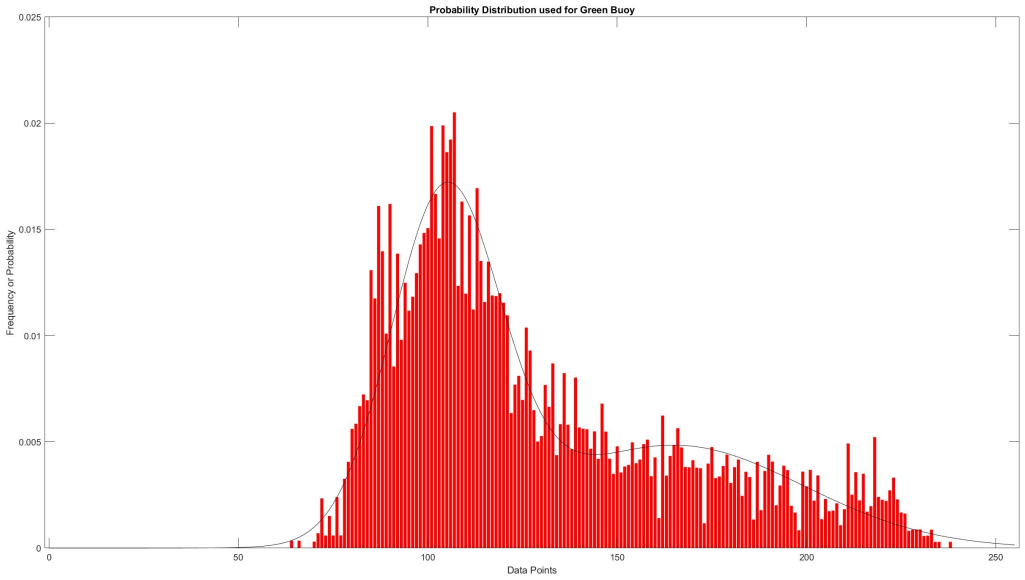

Figure 15: Gaussian Model used for Green Buoy

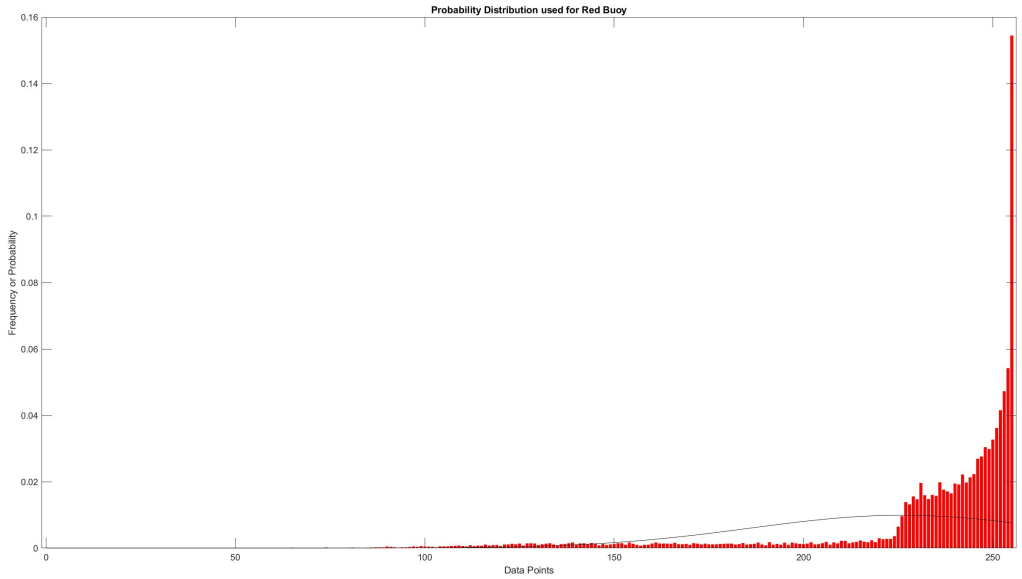

Figure 16: Gaussian Model used for Red Buoy

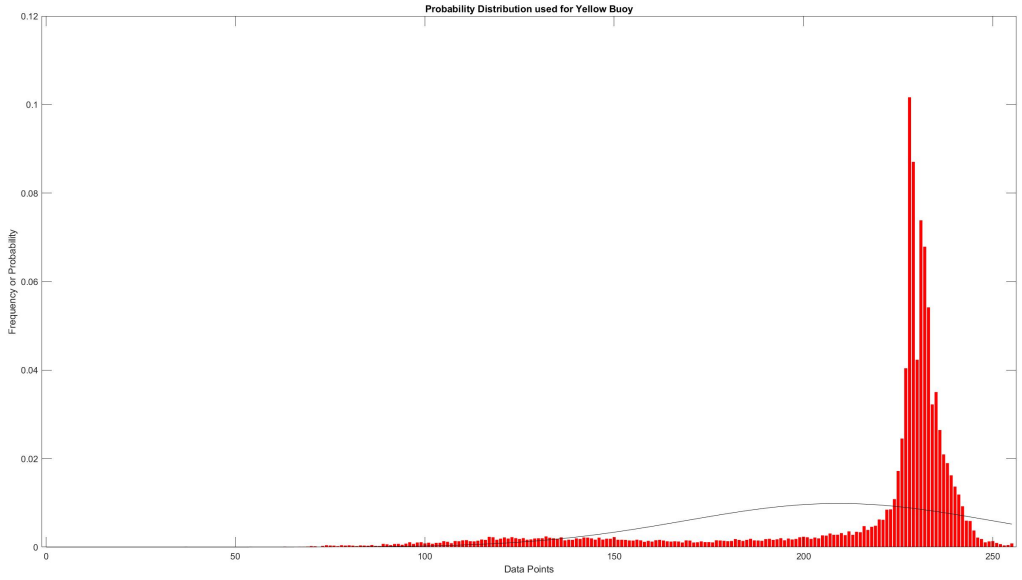

Figure 17: Gaussian Model used for Yellow Buoy

Part 3. Buoy Detection

Using the Gaussian models generated in Part 2 and an algorithm similar to segment-1D, we generated a video of segmented image frames. The algorithm also generates a set of binary images showing the location of the three buoys in the frames. An example frame is shown in Figure 18. It has been observed that the most difficulty is faced in the detection of green buoy. So, the final output of Part 3 has no significant improvement as compared to the output of Part 0. There are several reasons for the same.

1. The criteria used to segment and detect the buoys is not efficient.

2. The models generated are not good enough to model a RGB color space.

There have been several attempts made to resolve problem 1, but I could not figure out any criteria which improved the result significantly. So it has been concluded the result may improve if a different color space is used.

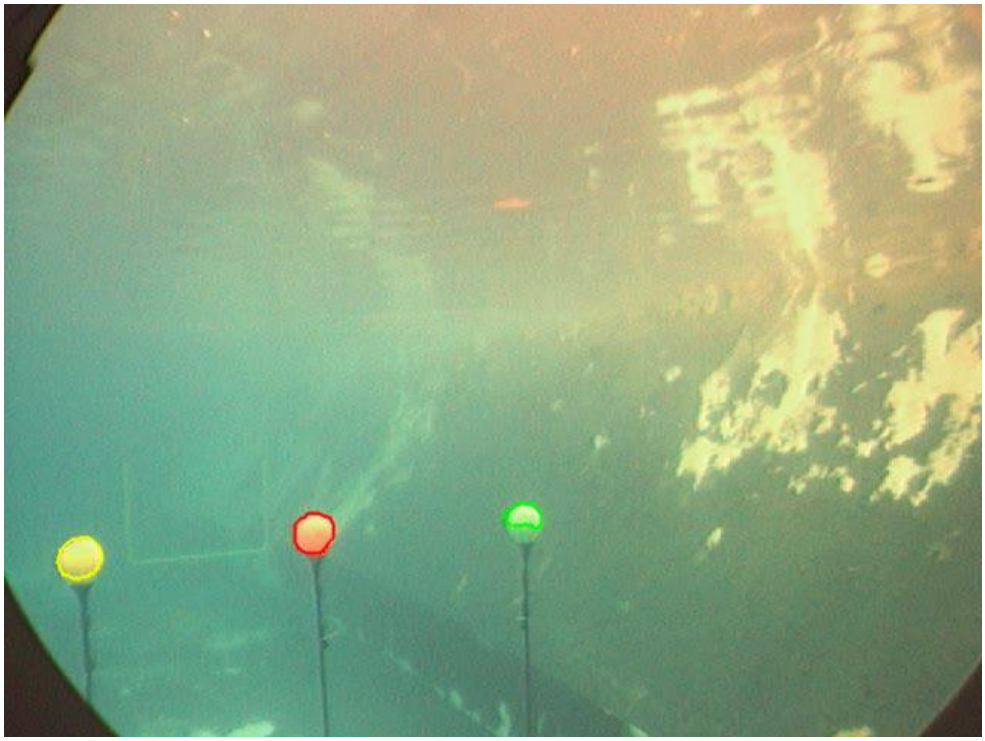

Figure 18: Segmented Image Frame using Gaussian Model

Alternate Approach. HSV Color Space

As it has been concluded in Part3, the output may be improved by using a different color space. MATLAB provides libraries for 5 different color spaces. All these color spaces have been incorporated in the algorithm for computation for average histogram. But after studying the results from each color space, it was concluded that using HSV color space should improve the segmentation and detection of the buoy.

Similar to Part 2 for RGB color space, I first applied all the possible models with K ranging from 1 to 5 on the images and decided upon the model based upon the quality of output per model. Note that, the data that has been used to generate the Gaussian models is in HSV color space. The final models used for green buoy, red buoy and yellow buoy are shown in Figure 19, Figure 20, and Figure 21 respectively.

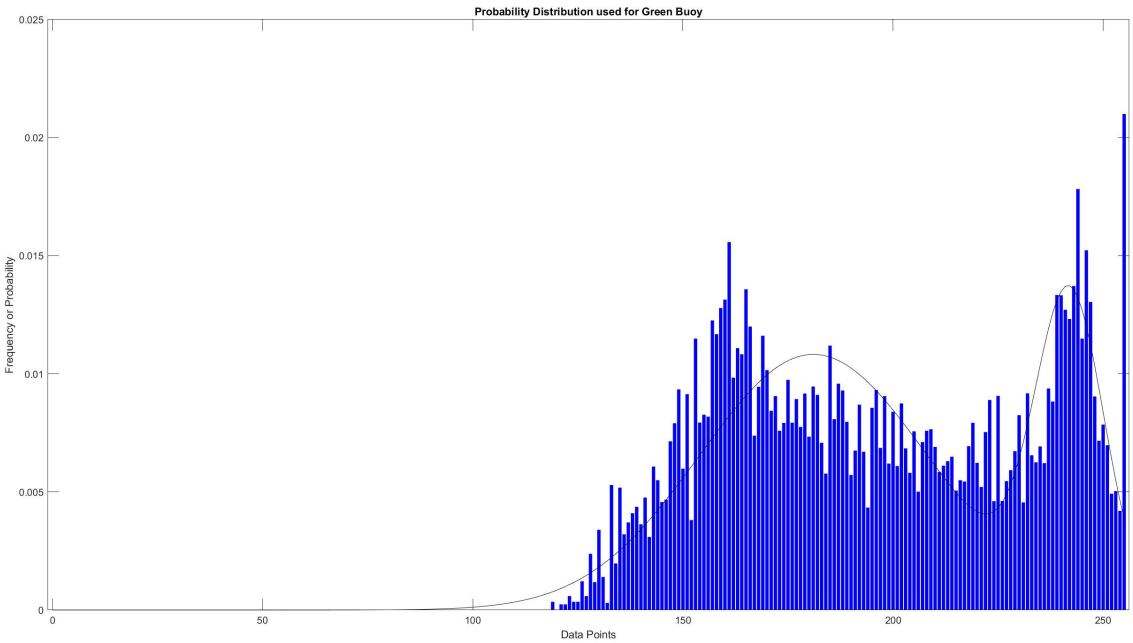

Figure 19: Gaussian Model used for Green Buoy (HSV color space)

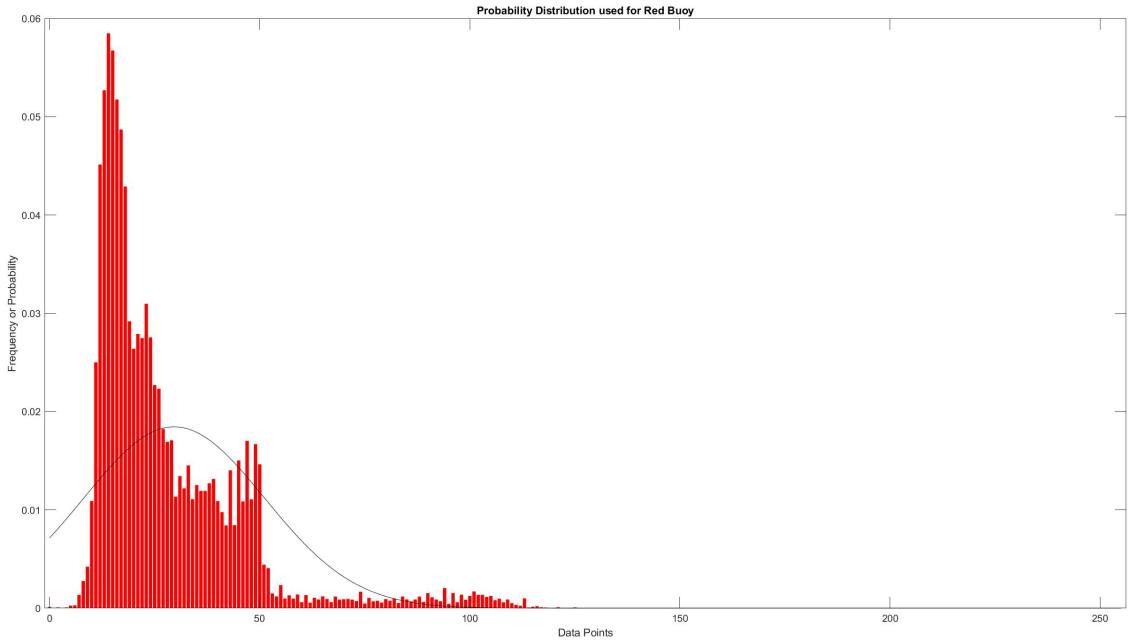

Figure 20: Gaussian Model used for Red Buoy (HSV color space)

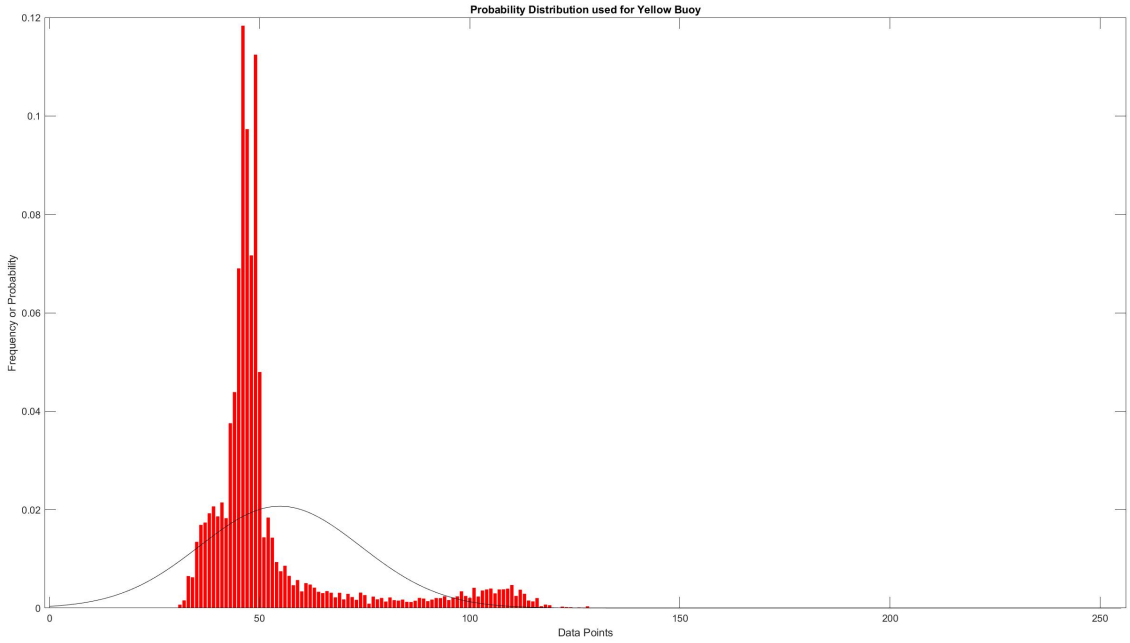

Figure 21: Gaussian Model used for Yellow Buoy (HSV color space)

Then, similar to Part 3 for RGB color space, using the Gaussian models generated for HSV color space and an algorithm similar to detectBuoy_RGB, a video of segmented image frames has been generated. The algorithm also generates a set of binary images showing the location of the three buoys in the frames. An example frame is shown in Figure 22. It has been observed that, similar to RGB color space, the most difficulty is faced in the detection of green buoy. But, still the output is better as compared to the output from RGB color space. More portion of the green buoy is detected in case of HSV color space as compared to the portion detected in RGB color space. This result is observed because HSV color space is slightly impartial to the intensity of the image which is not true in case of RGB color space. So, the final output has some improvement as compared to the output of Part 0.

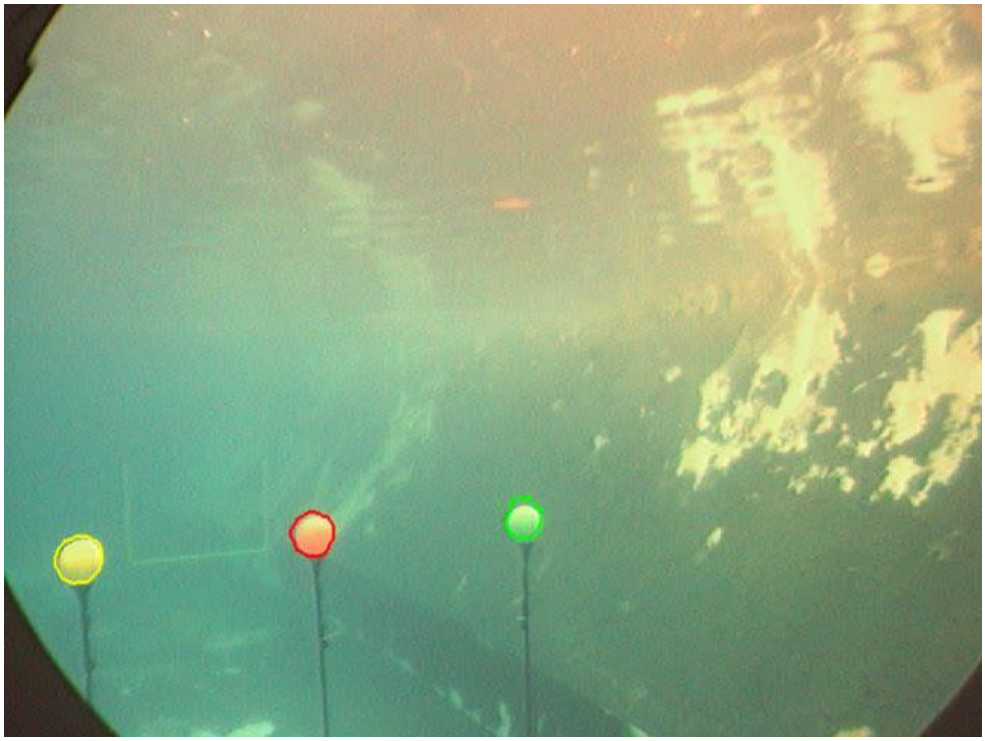

Figure 22: Segmented Image Frame using Gaussian Model (HSV color space)

Similarly, the output can be generated for other color spaces with no significant changes to the existing algorithm.

Discussion

In this project, the major portions that took time were the implementation of Expectation-Maximization algorithm and the decision for criteria for segmentation and detection of the buoy.

The implementation of Expectation-Maximization algorithm took time as it was difficult to weed out the errors in the algorithm. In the end, it was found that the implementation was correct, it was the data that created an issue. From studying the MATLAB **fitgmdist** code, I understood that the data being provided to the algorithm need to be in a certain format. Not only that, but the way the algorithm treats the data also matters. Using the knowledge gained from the study of the MATLAB code, I improved my implementation and it worked with the right kind of input.

The decision for criteria for segmentation and detection of the buoy took time because it is not just to implement the understanding of a concept but it is a tuning process. It is the part of the algorithm that always takes time. This is also the part which has to be modified when the color space is changed.

References

[1] Bishop, Christopher M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, 2006.

[2] E-M, *http://www.cs.umd.edu/ djacobs/CMSC828seg/EM.pdf*

[3] MATLAB Documentation, *https://www.mathworks.com/help/matlab/*

[4] ShareLaTeX, Online LaTeXEditor, *https://www.sharelatex.com/*