ENPM673 - Perception for Autonomous Robots
Project 3 - Buoy Detection

Abstract

Buoys are distinctly colored and shaped floating devices used as underwater markings for navigation and can be segmented using color segmentation. However owing to the environmental noise and change in light intensities it is challenging and difficult to segment the buoys using color threshold techniques. Hence, we aim to get a tight contour over the segmented buoys using Gaussian Models to learn the color distribution and using the model to segment and detect the buoys.

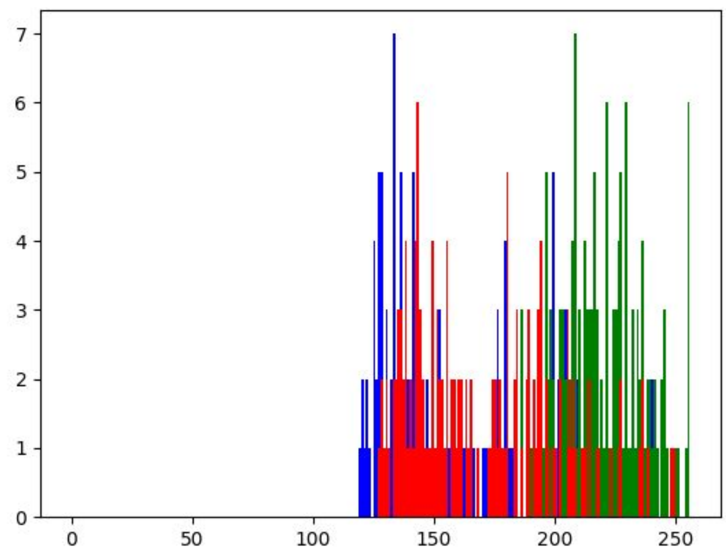
1. Detection Pipeline

1.1 Preparing Data

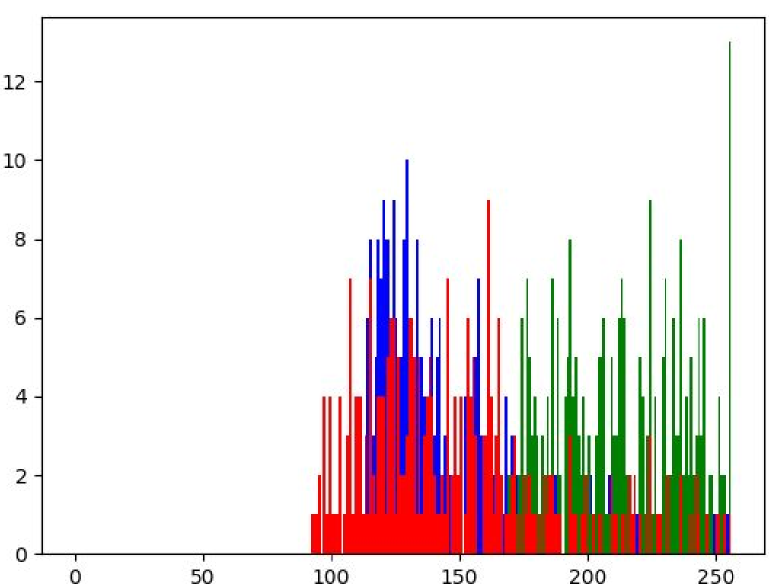
- For fitting a color model of the buoy we need sample data. The given video has 5 frames per second. We consider every consecutive 10th image such that 70% of the cropped buoys images are used for training and remaining for testing.
- We use the training frames to extract the model parameters and testing frames are used to evaluate its performance.
- Each cropped buoy image was extracted by marking the contours and applying it as mask to extract the data set.
- The data set is stored in seperate folders based on the color of the buoy respectively which we will be using in the next step.
- For each colored buoy, we group and visualize the color distribution of each channel of the sample data set RGB images.

1.2 Average Color Histogram

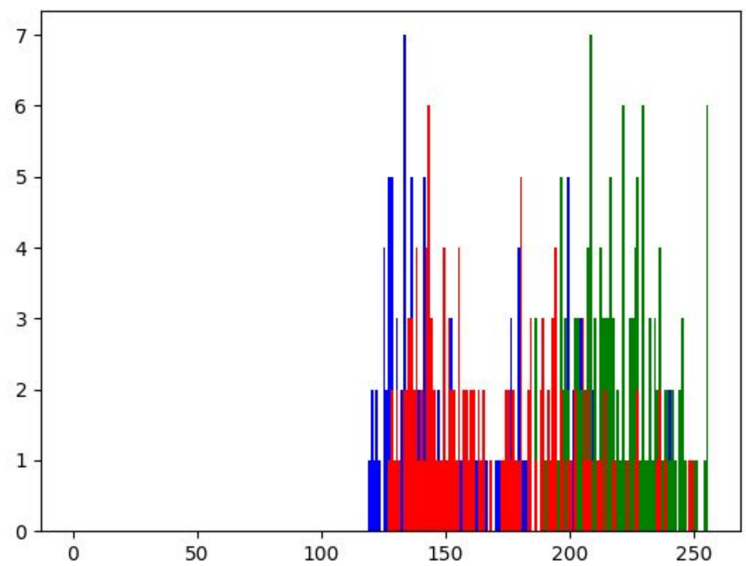
- To better fit the model on all samples of the color distribution we use average color histogram, which basically represents the distribution of colors in image. The color histogram is obtained for the RGB color space which gives count of pixels that have each color.



Average Color Histogram Red Buoy

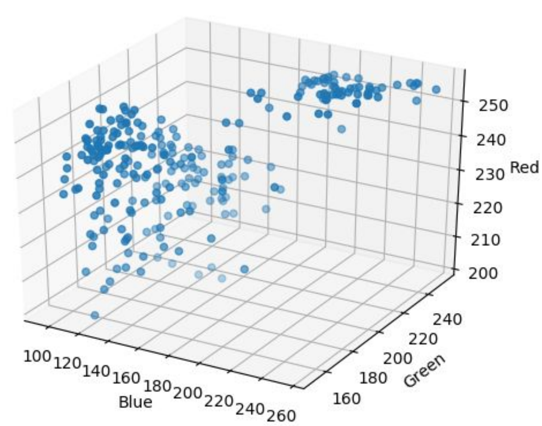


Average Color Histogram Green Buoy

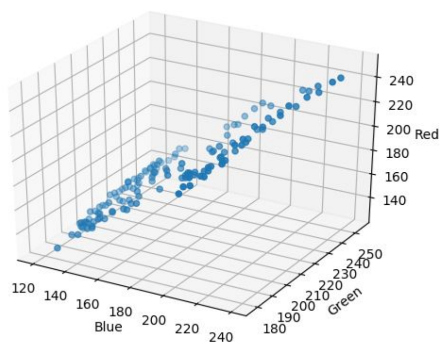


Average Color Histogram Yellow Buoy

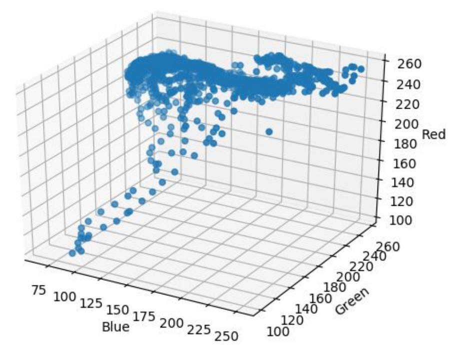
- From the average colored histogram of the Red Buoy, we can infer the red channel intensity to be in the range 180 to 255 due to higher presence of red color whereas the green and blue channel are in lesser intensity range.
- Similarly, for the Green Buoy we can see that the green channel intensity is in range 180 to 255 whereas the red and blue channel are in lesser intensity range.



Color Distribution of Red Buoy



Color Distribution of Green Buoy



Color Distribution of Yellow Buoy

- Also, since yellow buoy is a mixture of red and green, we can see that both the red and green channel colors are on higher intensity side.
- We will model these color distributions to tell whether a given pixel belongs to a certain colored buoy or not.
- Owing to the noise during cropping and impure color of buoy in the input sequence, there is increased intensity of non-dominant pixels.
- To get better idea of the average colored histograms, by equalizing the main color channel associated with the buoy has maximum intensities as expected.

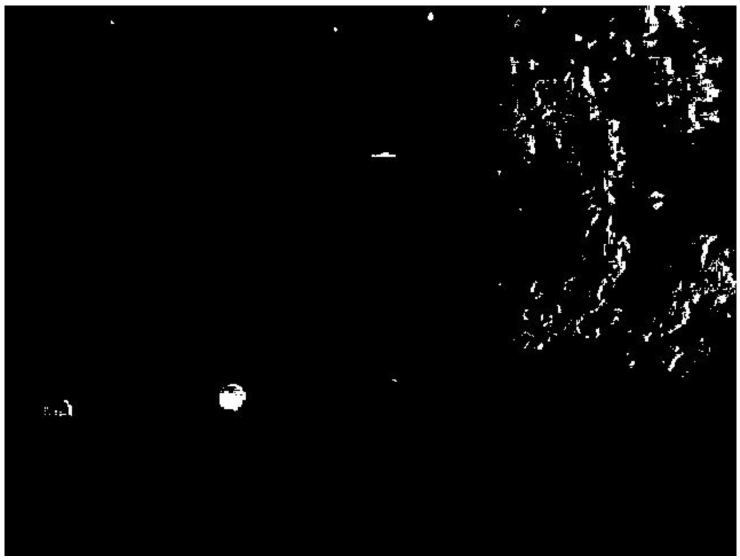
1.3 Using 1D Gaussian to model Color Distribution

- We used 1D Gaussian to model the set of samples using all pixels from cropped buoy, where the model parameters are mean and variance to form a gaussian.
- For each colored buoy, we compute the mean and variance of its respective main channel.
- Next we calculated the probability density function for each pixel of the image which would give the probability that the given pixel belongs to the color channel of the buoy, given by

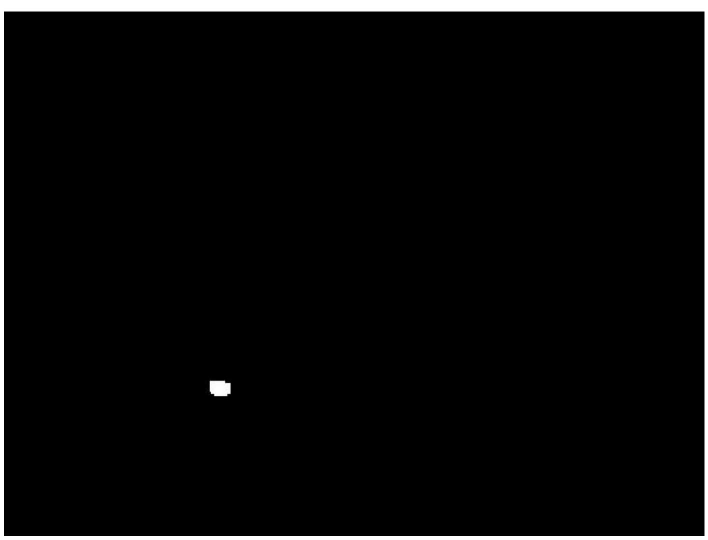
$$p(C_l|x) = \frac{p(x|C_l)p(C_l)}{\sum_{i=1}^{l} p(x|C_i)p(C_i)}$$

- Now by applying thresholding on the pdf to get maximum probability of belonging to needed color and hence eventually finding all the pixels containing that buoy, ideally around the standard deviation.
- After fine tuning the threshold limits on all R,G and B channel to get pixels that better identify the buoy, we switched to cumulative density function which gave us better results against the noise channels that kept picking up in the reflections.

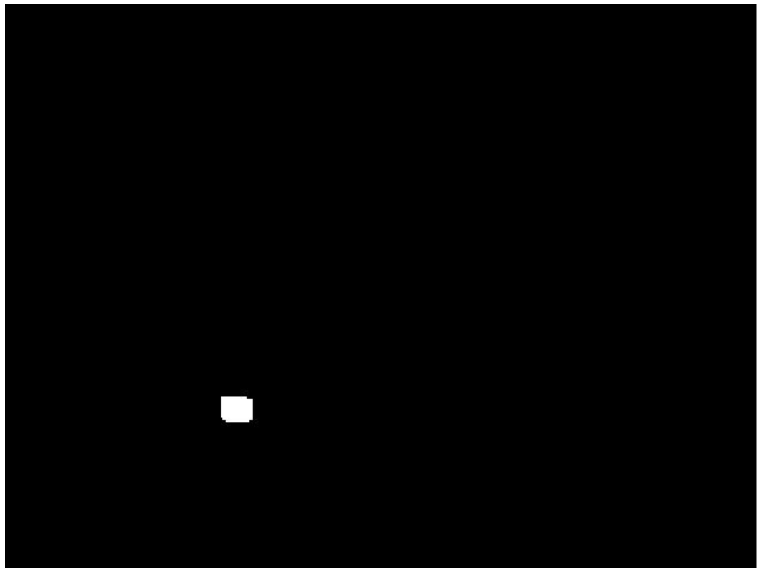
- Repeating the same steps, for all frames and performing morphological operations like morph(to remove noise), erode(to fine tune blobs), dilation(to get well defined blobs).
- Finally for a well defined blob of the buoy we get, we find the contours of it and the centroid associated with it.
- With the centroid we detect and draw the contour with respective buoy color.
- However, sometimes in the video the green circle shows somewhere else in the reflection/noise due to the inability to remove noise completely.



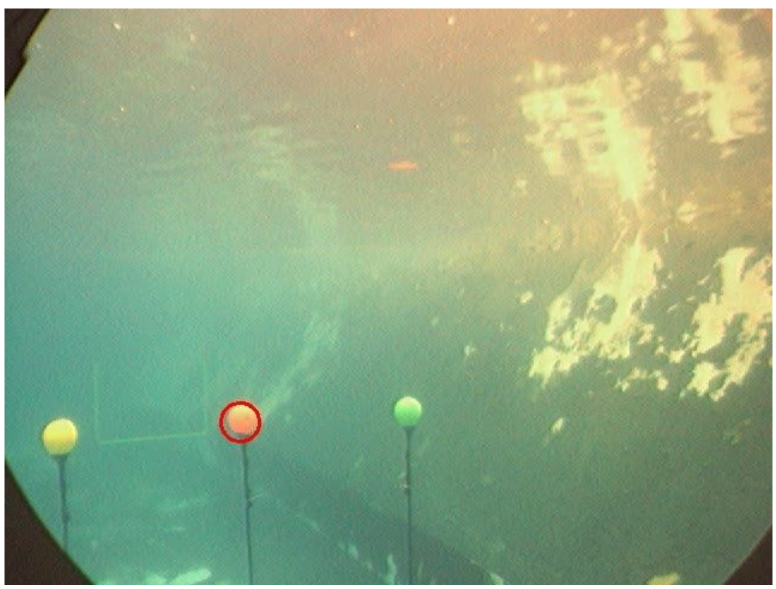
Thresholded Image



Morph Image



Dilated Image



Red Buoy Detected

2. Gaussian Mixture Models and Maximum Likelihood

In GMM,we recover the model parameters of the combined data to obtain three 1D gaussians. For the 1D gaussian to work we need to create multiple upper and lower bounds different colour channels and still it wouldn't be able to fit the exact ellipse. Also, when the lighting is different all over the frame the locating the exact buoys is not possible using this method. Since, the colors may not be bounded well by the ellipsoid.

2.1 Gaussian Mixture Models

Modeling the likelihood as gaussian is advantageous because small light variation makes light spread out in ellipsoid form where actual color is in middle and deviations is in all direction which is why single gaussian works well for color segmentation. However, In finding a color in varying lighting conditions, the single gaussian model will not be sufficient. In these cases maximum likelihood estimation can be achieved by leveraging the gaussians or a sum of gaussians to model our function given by,

$$p(C_l|x) = \sum_{i=1}^{k} \pi_i \mathcal{N}(x, \mu_i, \Sigma_i)$$

2.2 Expectation Maximization

The plan is to consider the RGB data channels and so the input data is 3 Dimensional. Each image from the training data is taken and stored as array in row major form. Now all these rows are appended. So, the input to the GMM trainer is just an 3xN array. Here, N is the total number of pixels in all the training data.

Given that the data is 3-dimensional the covariance matrix is of order 3. Inputs are taken as follows:

- The mean is taken as 3x1 vector.
- Depending on the number of clusters needed (n), the n mean vectors are initialized randomly from the data points.
- n covariance matrices are initialized as scaled identity matrices.
- Weight vector of size n is initialized with 1/n as the seed value.

We alternate until convergence the following steps

- **Expectation or E-Step** where we assign points to the cluster based on weight given by

$$\alpha_{i,j} = \frac{\pi_i p(x_j|C_i)}{\sum_{i=1}^{k} \pi_i p(x_j|C_i)}$$

  We compute the expected values of the latent variable in this step. Essentially we compute the conditional probability of pixel belonging to a cluster. We compute the fraction of each weight wrt total sum of latent variables for all the initialized weights, mean and co-variances.

- **Maximization or M-Step** is where we evaluate best parameters that best fit points by

$$\mu_i = \frac{\sum_{j=1}^{N} \alpha_{i,j} x_j}{\sum_{j=1}^{N} \alpha_{i,j}}$$

$$\Sigma_k = \frac{\sum_{j=1}^{N} \alpha_{i,j}(x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^{N} \alpha_{i,j}}$$

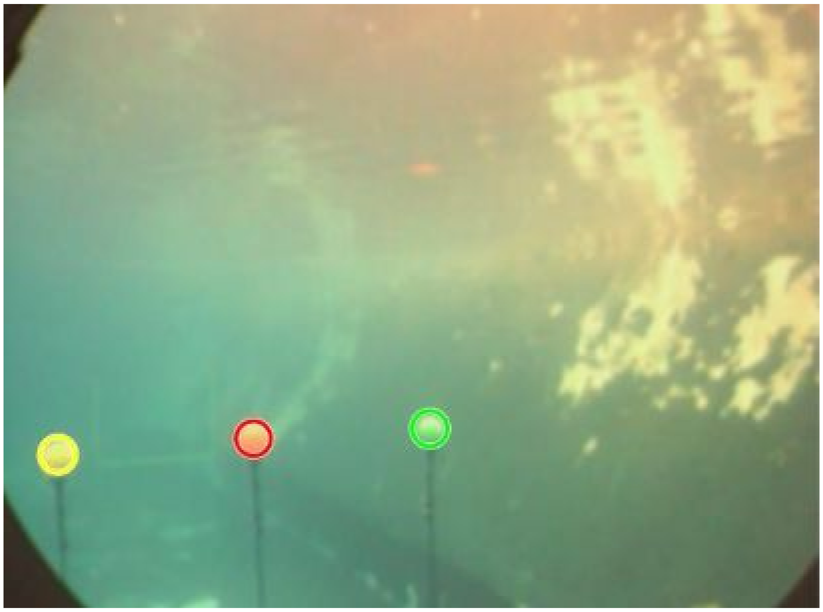
$$\pi_i = \frac{1}{N} \sum_j \alpha_{i,j}$$

The estimated latent value is used to compute the new mean, covariance and weights for the next iteration using the formula stated above.

Now using the newly updated parameters to compute the latent variable and the log likelihood. The difference between the past and the current likelihood is checked to be less than epsilon. Parameters are estimated until the difference converges to epsilon.

- We estimate and identify the pixels that belong to the respective buoy by the user defined threshold and morphological operations to obtain the buoys contours and hence draw the contours around them.

Thresholded Image



Buoys Contours Detected

2.3 Learning Color Models

- We find how many gaussians we need to fit the 1D data input for each main color of the buoy.
- Next we compute the model parameters for the 3D (RGB) input for each buoy with different gaussians.
- We compute by trial and error that the number of gaussians that best fit are decided by the cases where we start getting unnecessary peaks.
- By generating the binary image of the calculated pdf we can see that 3 gaussians mostly give us the best binary image output.

2.4 Buoy Detection

- When we compare the binary image output of the 3D vs 1D gaussian, we can see that the binary image of 3D has a lot of noise around the buoys and hence is difficult to segment it.
- Even after thresholding using standard deviation, noise around the buoys was hindering the segmentation process.
- Further morphological operations are performed to fine tune the buoy detection.

3. Report

3.1 Parameter Tuning

Likelihood range of each buoy is observed. For each frame the likelihood is computed and its filtered using the observed threshold range. This done for each buoy and circles are fitted using centroid of the buoy pixels identified.

3.2 Further Analysis and Observations

- Using the RGB color space holds good for pure color conditions. However for lighting, background color noises and motion make it difficult to segment the buoys. The use of HSV or LAB color space can be considered to be adaptive to such changes.
- The hue would stay relatively constant for buoy, differences in lighting would become value and colorfulness of each color will be saturation component. We tried using HSV, however it is haphazard and extreme for green buoy because of the green background noise.
- A good alternative to conventional RGB and HSV would be RGBY which includes additional yellow layer which provides clear identification of yellow color. However, if green noise was high, yellow buoy will be detected as green buoy. Also, additional yellow layer will attenuate sunlight noise and extra yellow tint.

How to run code

1. Unzip the folder which has the code, input sequences and the datasets.
2. Each of the following code parts needs to be separately run.
3. Run the codes in the following order

   python data_preparation.py

   python trainGMM.py

   This stores the parameters needed into the parameters file in .npy format
4. For 1D gaussian run

   python part2_3.py
5. For generating histogram plots run

   python part3_1.py
6. To detect buoy in the video frame run

   python part3_3.py

Project Team

1. Nantha Kumar Sunder (UID: 116104777)
2. Nithish Kumar (UID: 116316958)
3. Rama Prashanth (UID: 116428941)

References

1. Computer Vision, A Modern Approach, Forsyth and Ponce (http://cmuems.com/excap/readings/forsyth-ponce-computer-vision-a-modern-approach.pdf)
2. https://www.learnopencv.com/color-spaces-in-opencv-cpp-python/
3. https://cmsc426.github.io/colorseg/#gmmcases
4. https://www.python-course.eu/expectation_maximization_and_gaussian_mixture_models.php

Video Link:
https://drive.google.com/drive/folders/1uycS5PXgsG3qKZLo9W1OhZbHLsdWrR07?usp=sharing