ENPM673 Perception for Autonomous Robots

PROJECT 3

Color Segmentation using Gaussian Mixure Models

Nalin Das (116698290)
Aditya Khopkar (116911627)
Nidhi Bhojak (116787529)

Contents

List of Figures

1 Introduction

The project is on the implementation of Gaussian Mixture Models and Expectations Maximisation techniques for Color Segmentation. The given input video sequence is an underwater scene consisting of three buoys of different colors i.e Yellow, Green and Orange. A buoy is uniquely colored floating device, used for the purpose of underwater navigation. Our goal is to segment and detect buoys of each color from the video sequence.The conventional segmentation based on the color thresholding would not work for this case as the buoys are underwater. This would make buoys experience various lighting conditions and noises in different frames.

Thus to overcome this, we used Gaussian Mixture Models to take into consideration various color distribution throughout the video. And use the generated model to detect and segment the buoys.

The Output Video files can be accessed from the drive link:
https://drive.google.com/drive/folders/1BQdnzuijcS8Z-Qyx6NJY_FNJZQ4Cw7JU?usp=sharing

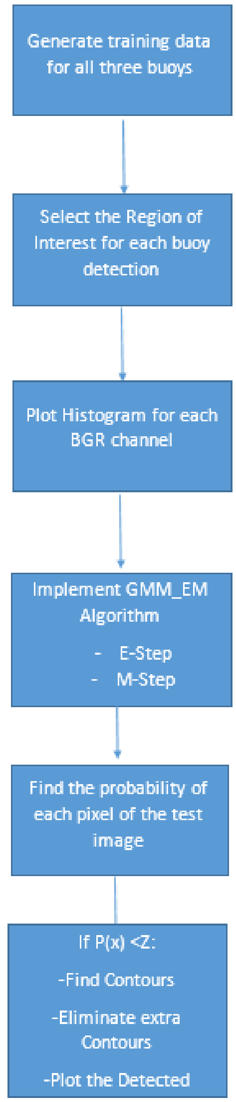

Figure 1: General Pipeline for the Project

2 Data Preparation

The first step is to extract each frame from the given video sequence. Upon extraction we get 200 frames from the video. Now, we need to pull out samples of each colored buoys for each frame and arrange them in a separate folder. This data is further divided into two part i.e Training Set and Testing Set in ratio 70:30.
Test images are further cropped to get only the details about the specific buoy so that we have minimal noise. Later, for each buoy the pipeline is divided into Training the dataset using EM algorithm and Testing.

3 Choosing the Approach

There are two approaches to train the model: extracting the channel of interest and using 1D Gaussian to train the model or using all the three channels to train the model using 3D Gaussian. Here we choose to use the 1D Gaussian method.

First we take the images from training dataset and then plot the histograms for each channel to get an idea of the number of clusters present. To determine the total number of clusters(K), we can add up the clusters from each of the channel. The total number of clusters(K) will let us determine the number of gaussians needed to detect the particular colored buoy. For eg, K=3 for Green Buoy, K=2 for Orange Buoy and K=2 for Yellow Buoy.

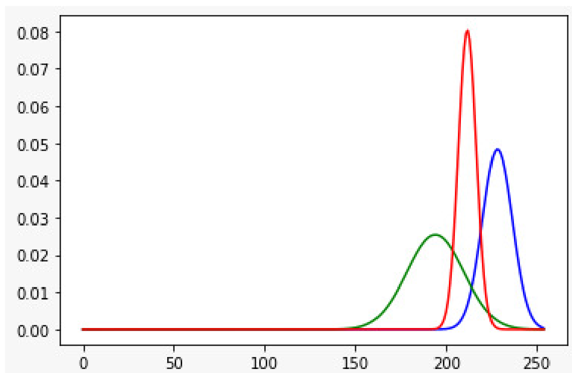

Figure 2: Gaussian Distribution for Green Buoy (K=3)

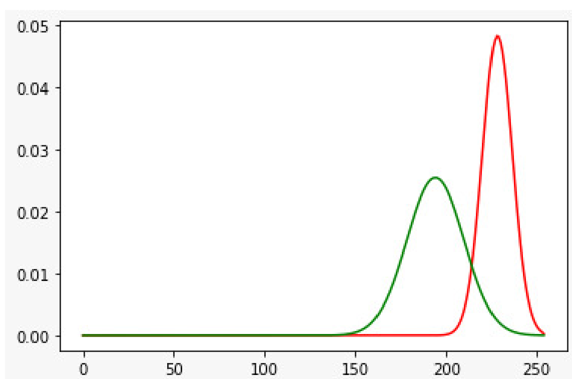

Figure 3: Gaussian Distribution for Yellow Buoy (K=2)

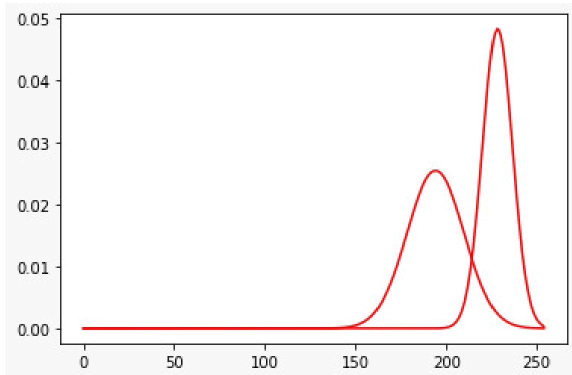

Figure 4: Gaussian Distribution for Orange Buoy (K=2)

4 Gaussian Mixture Models and Expectation Maximisation Problem

1. **Gaussian Mixture Models:** Gaussian Mixture Models are a probabilistic model for representing normally distributed Gaussian (sub-populations) within an overall data(population). Here, the data is unlabelled i.e we cannot identify which point came from which Gaussian distribution.

2. **Expectation Maximisation:** It is an iterative maximisation technique to find maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. The method consists of two steps:

   (a) **Expectation-** Estimate the missing variables in the dataset

   (b) **Maximisation-** Maximize the parameters of the model in the presence of the data

   First, we take random datapoints with some K distributions with predefined mean and variances. Then, computing the probability of each point as if the point is generated by each of the distributions, we repeatedly update the parameters by maximising the log-likelihood of the data. With sufficient number of iterations the algorithm converges at the local optimum and thus gives us the mixture of Gaussians which accurately fits the data. Steps to compute the GMM Parameters using EM Algorithm:

   - Initialise the mean, variance and weights with random values and evaluate the log likelihood of the initial points.
   - Compute the expected values of the hidden variable for the given parameter values.

   $$\gamma(x) = \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\Sigma_{i=1}^{k} \pi_j N(x|\mu_k, \Sigma_k)}$$

   - Update the parameters of the model based on the latent variable we got from the previous step.

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)x_n}{\gamma_j x_n}$$

$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)(x_n - \mu_j)(x_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(x_n)$$

- Evaluate the log-likelihood and if it is not converging return to E-step

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

Here, N refers to the number of samples and K is the number of Gaussians. Hence, using the above algorithm we determine the true values of parameters for the Gaussian.

5 Segmentation Using 1D Gaussian

After we have the histograms for all the 3 channels, we need to build 1D Gaussians from them in order to segment the pixels of the image.

As interpreted from the histogram we can extract out the details we want from the corresponding channel of interest for each buoy. For example, to detect red buoy we can use the histogram of red channel to generate the 1D gaussians, to detect the green buoy we will extract the histogram for green channel and to detect Yellow buoy we take histogram of both the red and green channel.

The Gaussians can be generated by calculating the mean and variance from the histogram data of the main color channel for respective buoys. Once we have the gaussians are generated, the next steps are as follows:

- Iterate over each pixel of each image in the test dataset.
- Extract the intensity value for the corresponding channel of interest for the buoy to detect.
- Check whether that intensity falls under the Gaussian distribution for that color channel.
- If yes, apply thresholding function to segment out the pixel from the image.
- Through the above step, we generate the binary image corresponding to each buoy.

6 Buoy Detection

Now we need to detect the contour of the obtained detected mask image. First, we blur the image then we use *findContours()* function to highlight the detected buoy. Once, the contours are detected we sort them. After that, we use the *minEnclosingCircle()* function to get the approximate center and radius of an enclosing circle to the buoy. We finally use the *cv2.circle()* to plot the circle on the detected buoy.

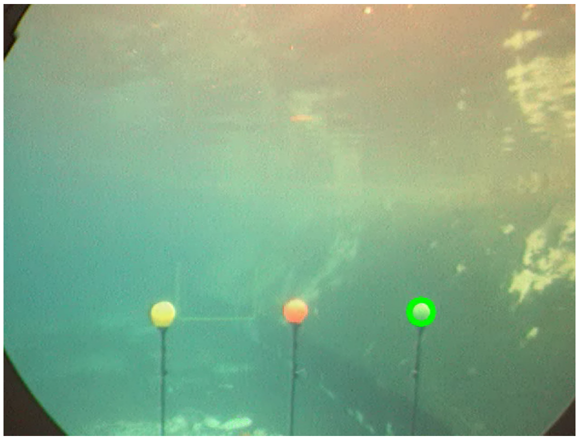

Figure 5: Detected Green Buoy

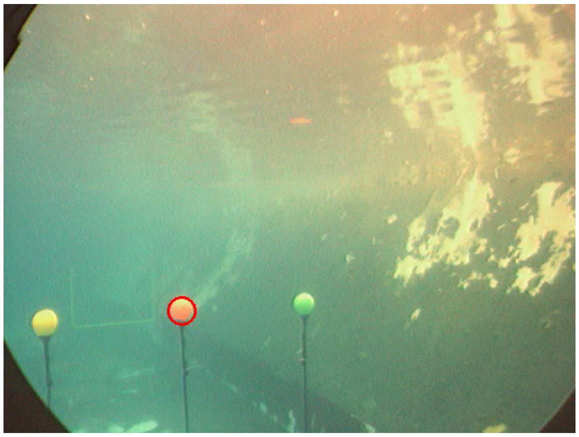

Figure 6: Detected Orange Buoy

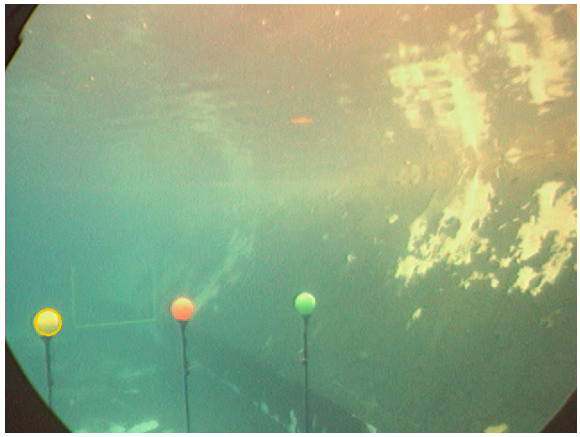

Figure 7: Detected Yellow Buoy

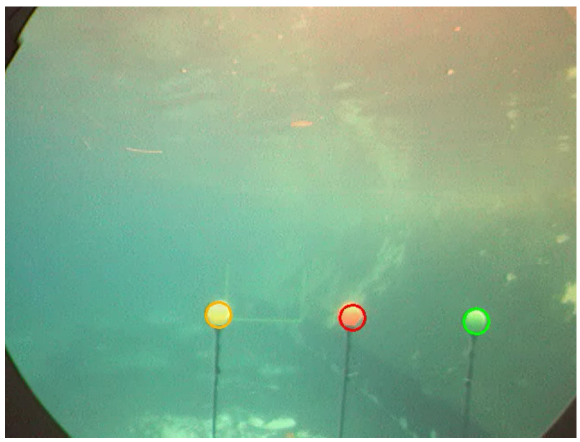

Figure 8: Detected All Buoys

7 Challenges Faced

- Preparation of dataset : Initially there were noises due to boundaries around buoys which caused problems in obtaining histograms.It was then corrected by improving the datset using tighter buoy area.
- Suboptimal detection of the Buoy: There were glitches observed due to the presence of similar colored details in the background. Thus, during contour detection, some background area was also detected which contained the same intensities as that of the buoy to be detected.

8 References

- https://www.python-course.eu/expectation_maximization_and_gaussian_mixture_models.php
- https://en.wikipedia.org/wiki/Expectation-maximization_algorithm
- https://youtu.be/qMTuMa86NzU