ECE 442 Introduction to Multimedia Signal Processing Winter 2021 Laboratory #4, Background Subtraction, March 12, 2021 Report Submission Due Date: March 24, 2021

1. Please submit in softcopy at eclass by 5 pm of March 24th as a single zip file in FirstName_LastName_lab4.zip. The submission is expected to contain the following files: your answer sheet, Matlab code, and input/output files to be reproducible on our side.
2. Questions that should be answered in your lab report are numbered and marked as bold within the following text. Please number your answers accordingly.
3. Certain questions ask for images to be uploaded. To save images from MATLAB to the file system, use imwrite(Im, 'filename.png'). Do this instead of saving an image from a figure window. Always use a lossless extension, e.g. png.
4. Make sure your Matlab code is bug-free and works out of the box. Please be sure to submit all main and helper functions. Be sure to not include absolute paths. Points will be deducted if your code does not work at our side.

Background Subtraction

Background subtraction is an important preprocessing step in computer vision applications. This technique is used to extract foreground moving objects from a given video. For instance, to find the number of people visiting a room or to extract information from traffic cameras about vehicles, we need to extract the moving foreground from static background.

The simplest method to remove the background of a video is to subtract the Mean Frame from all of the frames. Then, we use a threshold to capture the foreground. Despite its simplicity, this method has not good performance in real world applications. For real world applications, we may use other methods like modeling the distribution of each pixel as a mixture of Gaussians.

In this Lab, we will start with investigating the performance of Mean Frame method. Then, you will implement and investigate the performance of mixture of Gaussians (MoG) for background modeling.

Here, we will consider the first 240 frames of the given video as the training set and model the distribution of each pixel value using MoG. Then, we use this model to predict background/foreground for the test frame(s). This approach is unable to consider the dynamic background changes e.g. changes in the illumination or introducing/removing objects. To consider these dynamic changes, we should have an online background modeling framework which is out of the scope of this lab.

1) Mean Frame Method

Read the video "street.mp4" into v1 using the following Matlab command:

```
v1 = VideoReader('street.mp4');
```

To find the number of frames, height, and width of video you can use v1.NumberOfFrames, v1.Height, and v1.Width, respectively. In order to read the ith frame of video, you can use `read(v1,i)` command.

Using these commands, compute the mean frame for the training frames of the given video. Extract the foreground image using thresholding for testing frame 420.

Question1: Find the optimum threshold value to extract the foreground by Mean Frame method for frame 420 using trial-and-error. Observe the result for three different threshold values: one lower than the optimum value, optimum value, and one higher than the optimum value. Based on your observation, describe the effect of changing threshold briefly. Include the mean frame and foreground image obtain by optimum value of threshold in the zip file. (25 points)

2) Mixture of Gaussian Method

In this method, we model the distribution for each pixel of training frames as a mixture of Gaussians with K components:

$$P(X = I(x,y)|\ \mu, \Sigma) = \sum_{i=1}^{K} w_i \eta(X|\ \mu_i, \Sigma_i)$$

$$\eta(X|\ \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{\frac{d}{2}}\ |\Sigma_i|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(X-\mu_i)^T \Sigma_i^{-1}(X-\mu_i)\right)$$

where X=I(x,y) is the pixel value at position (x,y) and μ, Σ are MoG mean and covariance matrix, respectively. In this lab, you should fit a MoG to each pixel in training frames which contain the static background and evaluate the performance of this background model to extract foreground for the testing frame(s). You can implement MoG from scratch or use Matlab fitgmdist(Y, K) function to fit a MoG with K components to Y data. If we store training samples of a particular pixel in vector Y, then the following commend fits a MoG model with k components to our data.

```
GMModel = fitgmdist(Y,K)
```

Since fitgmdist function uses iterative Expectation-Maximization (EM) algorithm, you may need to change the default number of iterations to make sure convergence is achieved. To do that you can use the following Matlab command for fitgmdist:

```
iter = 200;                              % 200 iterations is just an example
GMModel = fitgmdist(Y,k,'RegularizationValue', 0.1
,'Start','randSample','Options',statset('Display','off','MaxIter',iter,'To
lFun',1e-6));
```

Hint: fitgmdist may raise an error of convergence if iteration number or regularization value is not suitable.

Question2: Fit a Mixture of Gaussians with K=5 to pixel(360,640) of the first 240 frames. Choose an iteration number to make sure that EM algorithm has converged. (15 points).

Let us assume we have found the MoG model (GMModel) and want to calculate the probability of the pixel X has value x. First, we need to define an interval centered at x, find the pdf for this interval based on the MoG model and then calculate the probability by integrating on the pdf. In the following example we define an interval of size 1 around x with step size 0.0001 and find the probability for learned MoG:

```
interval = (x-0.5):0.0001:(x+0.5);
PDF_x = pdf(GMModel,interval');
probability = trapz(interval,PDF_x);
```

In the next step, predict the class (foreground or background) for each pixel of testing frame(s) using thresholding on the developed MoG model for the background as follows:

$$\begin{cases} X \in \text{background} & \text{if} \quad \text{probability} > \text{threshold} \\ X \in \text{foreground} & \text{if} \quad \text{probability} < \text{threshold} \end{cases}$$

Question3: Test the performance of the MoG background model for pixel(360,640) considering 420th frame as the testing frame. Using K=5, find the minimum value of threshold (20 points).

Question4: Using the minimum value of threshold, compare the results for K=1, K=3, and K=5 (15 points).

Question5: Compute the MoG background model with K=5 for each pixel of training frames located at a 300*400 box centered at the center of the frames. Extract the foreground and background pixels of the box for testing frame using three different thresholds (0.0001,0.001,0.01), and attach the result to the zip file(25 points).

Bonus:

Question6: Record your own video using VideoRecorder.m (available on eclass). The training frames should contain the background and the testing frame should also have foreground moving object(s). Extract the foreground for the testing frame, and report the resulted extracted foreground as well as the number of chosen Gaussian components. Attach the original video to the zip file (10 points).