Assignment 3: Real-Time Tracking Using Mixture of Gaussians

Submitted By: Jinay Dagli [20110084]

(I) Introduction

The process of removing the static background from a series of video frames is known as background modeling. A method known as background subtraction enables the foreground of an image to be retrieved for subsequent processing (object detection, etc.) and is typically utilized after the backdrop has been modeled. Therefore, foreground extraction and analysis include background modeling as a crucial component. In this assignment, a real-time tracking algorithm has been implemented using Gaussian Mixture Models.

The Waving Trees database from Microsoft has been used for this assignment. It consists of 287 frames that could be converted into a video.
The Google Colab file for the same can be found here. It consists of the final code for the assignment.

(II) Algorithm of Real-Time Tracking Using Gaussian Mixture Models

The main idea behind the algorithm is modeling each pixel of a frame of a video as a mixture of various (*K*) Gaussians and using an online approximation to update the model. As has been described in [1], the steps for face recognition using eigenfaces consist of the following:

1. Since in real scenarios, each pixel results from multiple surfaces under different lighting conditions, a mixture of adaptive Gaussians would be necessary to model the pixel value.
2. At any time *t*, we know the history of a pixel $\{x_0, y_0\}$ as
$$\{X_1, X_2, \ldots., X_n\} = \{I(x_0, y_0, i) : 0<i<(t+1)\},$$
where 'I' is the image sequence. The value of each pixel represents a measurement of the radiance in the direction of the sensor of the first object intersected by the pixel's optical ray.
3. This recent history of pixels is modeled using a *K* mixture of Gaussians. The value of *K* generally lies between 3 to 5. The corresponding probability of observing the current pixel value is given by:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t})$$

4. The Gaussian probability distribution is given by:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \Sigma^{-1} (X_t - \mu_t)}$$

5. A new pixel value will, in general, be represented by one of the major components of the mixture model and used to update the model. If the pixel process could be considered a stationary process, a standard method for maximizing the likelihood of the observed data is expectation maximization.
6. If none of the K distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance, and a low prior weight. The prior weights of the *K* distributions at time t, $w_{k,t}$, are adjusted as follows:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t})$$

where α is the learning rate[2] and $M_{k,t}$ is 1 for the model which matched and 0 for the remaining models.

7. The parameters of the distribution which matches the new observation are updated as follows:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \quad (6)$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \quad (7)$$

where the second learning rate[3], ρ, is

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k) \quad (8)$$

8. The first B distributions are chosen as the background model, where

$$B = argmin_b \left(\sum_{k=1}^{b} \omega_k > T \right)$$

9. T in the previous expression is a measure of the minimum portion of the data that should be accounted for by the background. This takes the "best" distributions until a certain portion, T, of the recent data has been accounted for.

(III) Results and Inference:

On performing the steps described in the algorithm, we observe the following results:
The original Waving Trees video formed by combining the frames given in the dataset can be seen below:
YouTube Link to the original Video

The link to the output videos (both using the defined function and using the in-built function) are provided below. For the video, I have used a rate of 5 frames per second. It can be changed in the code shared through the Google Colab linked at the top section of the report.

YouTube Link to the Output Video (Defined Function)
YouTube Link to the Output Video (In-Built Function)

The user-defined Mixture of Gaussians function is compared with the in-built MOG2 function of the cv2 library. The two comparisons can be seen from the two snaps given below.
Two of the snaps where movement is tracked using the code written are shown below. They are also compared with the corresponding video obtained using the in-built function.
(Next Page)

I have used two metrics: Peak-Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) in order to compare my function and the in-built MoG function. Each frame is compared in terms of PSNR and SSIM. The average PSNR error difference between the two outputs comes out to be nearly 2.6 %. Similarly, for the SSIM metric, the maximum structural similarity between the two comes out to be nearly 48 %, and it changes with changes in the model parameters.

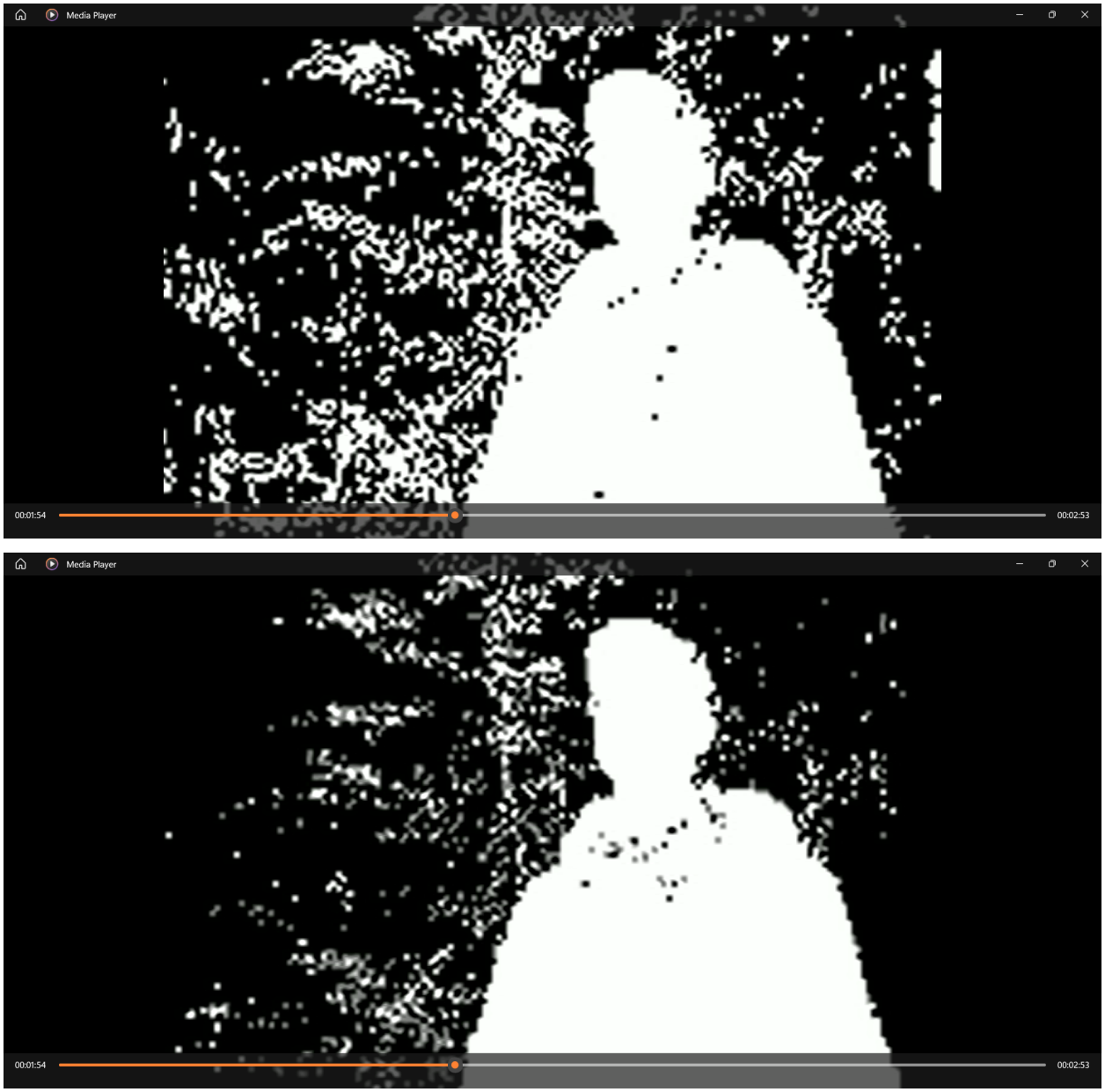

Fig. 1: A Snap from Video at same time for (a) Defined Function and (b) In-Built Function

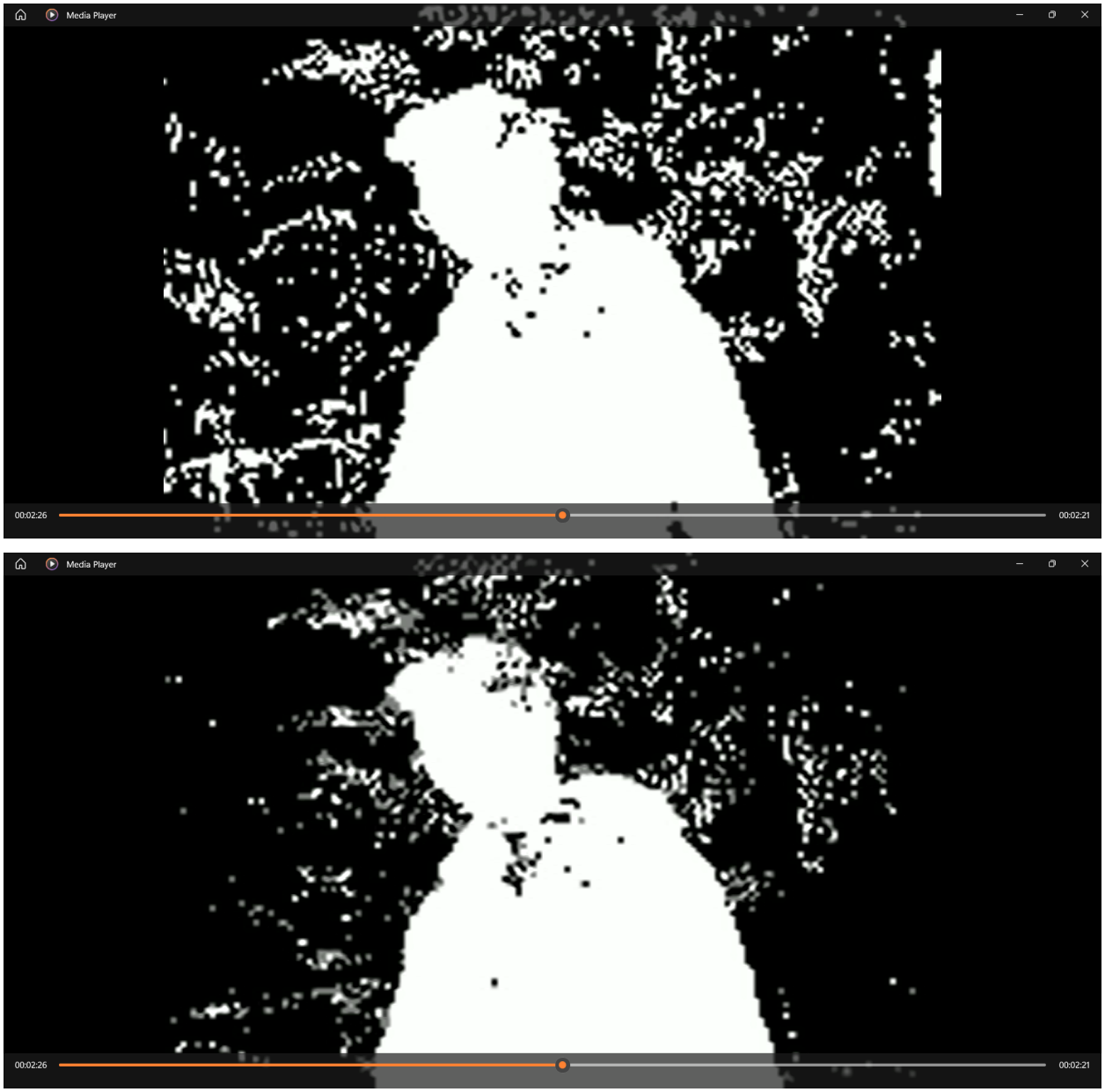

Fig. 2: Another Snap from Video at same time for (a) Defined Function and (b) In-Built Function

(IV) References:

[1] C. Stauffer and W.E.L Grimson, "Adaptive background mixture models for real-time tracking", Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[2] Belmar Garcia-Garcia, Thierry Bouwmans, Alberto Jorge Rosales Silva,
Background subtraction in real applications: Challenges, current models and future directions, Computer Science Review, Volume 35, 2020.

[3] GitHub Repo1: Link

[4] GitHub Repo2: Link

[5] GitHub Repo3: Link