Digital Video Processing CS6870

Background Subtraction using Gaussian Mixture Models[1]

Siddharth Nayak
EE16B073
Indian Institute of Technology Madras
Chennai, India

Advisor: Prof. Anurag Mittal

Abstract

Background Subtraction is quite important in video surveillance in shops, parking lots, subway stations, etc. One of the common methods used for real-time segmentation of moving regions in image sequences includes thresholding errors between an estimate of the image without moving objects and the current image. This report is on the implementation of the paper 'Adaptive background mixture models for real-time tracking' by Stauffer et al. The authors model each pixel as a mixture of Gaussians and use an online approximation to update the model. The Gaussian distributions of the adaptive mixture model are then evaluated to determine which are most likely to result from a background process. Each pixel is classified based on whether the Gaussian distribution which represents it most effectively, is considered part of the background model. This results in a stable, real-time outdoor tracker which reliably deals with lighting changes, repetitive motions from clutter, and long-term scene changes.

1 Introduction

The authors have come up with an adaptive model for background subtraction. They model each pixel as a mixture of Gaussians and use online approximation to determine which are most likely to result from a background. This model performs background subtraction in real-time.

2 Dataset

The datasets on which the model is tried out are:

- Run.avi: A 3-second video of a person running
- Jump.avi: A 5-second video of a person jumping
- Canoe: A 1200 frame dataset of video of a canoe in a river. This dataset has a dynamic background because of the flowing river.
- Highway: A 1700 frame dataset of video of cars moving on a highway. This dataset has a dynamic background because of the moving leaves on the trees.
- Subway(PETS): A 1200 fram dataset of a subway station.

[1]Implementation of Adaptive background mixture models for real-time tracking by Stauffer et al.

3 Implementation Details

The algorithm was implemented in python. The images are read using OpenCV.Note that OpenCV is used only for reading the images as matrix. None of the built-in functions in OpenCV are used. All of the implementation has been done using numpy. The image which is read has 3 channels (Red, Green and Blue) and is not converted to gray-scale.

Parameter Name	**Parameter Value**
K:Number of Gaussians	5
Standard Deviation Threshold	2.5
α : learning rate	0.005
T : Threshold for sum of weights	0.5
σ_{init} : Initial Standard Deviation	10
W_{init} : Initialised Weight Values	0.2

Table 1:Table of parameter values used in the implementation.

3.1 Simplification in equations

The gaussian used in the model is given by:

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

Here, $\Sigma = \sigma^2 \mathbf{I}$
Therefore, Σ^{-1} is given by $\frac{1}{\sigma^2}\mathbf{I}$ as it is a diagonal matrix.
Therefore, while implementing it in python we can just take the matrix used to store the values of σ and get the matrix Σ^{-1} by just using the above method.

4 Observations

The algorithm works well for normal videos like 'run.avi' and 'jump.avi'. The person in the video is detected as foreground with a fairly good performance. The model is slightly less accurate for videos which have a dynamic background as in the 'Highway dataset' and 'Canoe Dataset' as they have backgrounds like moving water(river) and moving trees and shadows of the trees.

4.1 Run Video

The model does background subtraction with a fairly good accuracy. The person is tracked as shown in Figure 1(a).

4.2 Jump Video

The model does detect the motion but does miss a few pixels on the shirt of the person because of very little change in the texture of the shirt color. The output is shown in Figure 1(b).

(a) Run.avi.

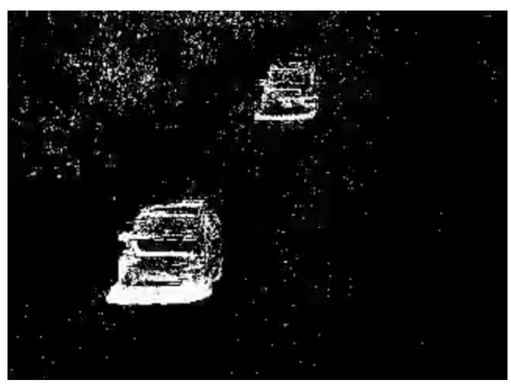
(b) Jump.avi.

Figure 1: Outputs for the Run.avi and Jump.avi video

4.3 Highway Dataset

The model does detect the moving cars well but it also detects the moving leaves on the trees as well as a little bit of shadow of the tree. This detection of moving leaves on the trees can be seen in Figure 2(a) and Figure 2(b) on the top left side of the frames where there are quite a few white dots.

(a) One of the frames.

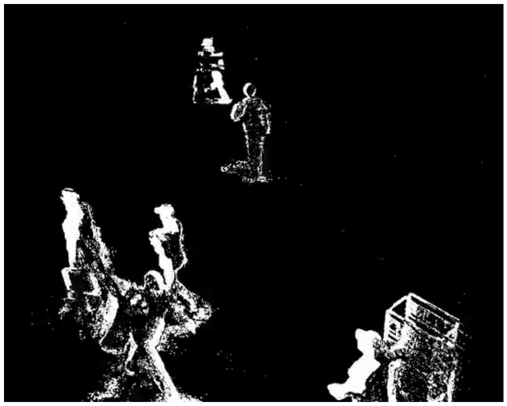


(b) One of the frames.

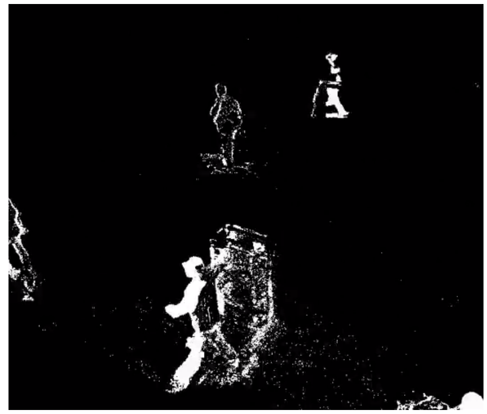
Figure 2: Outputs for the 'Highway dataset'.

4.4 PETS Dataset

The model performs very well in detecting the moving pedestrians on the subway station. One of the problems with the model's detection is that it also detects the reflections of the pedestrians on the floor. The output is shown in Figure 3(a) and Figure 3(b).



(a) One of the frames.

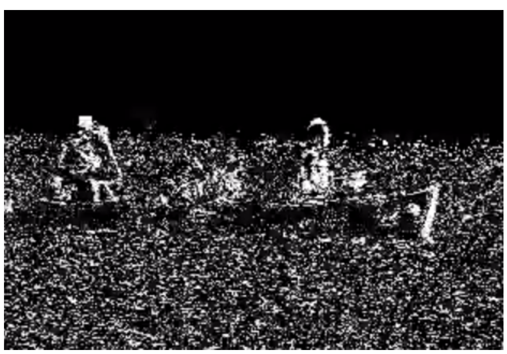


(b) One of the frames.

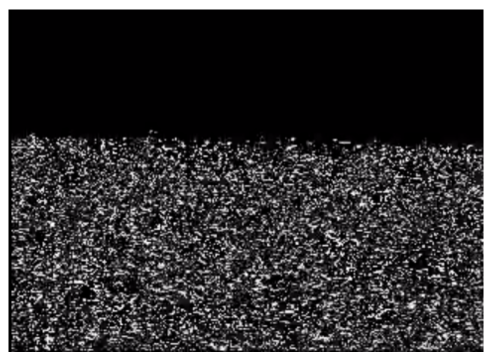
Figure 3: Outputs for the 'PETS dataset'

4.5 Canoe Dataset

The model does not work well in this dataset as the moving river in the frames also gets detected in the foreground which should not ideally happen. The canoe and the people along with the flowing river are detected as foreground as shown in Figure 4(a). In Figure 4(b) the model just detects the flowing river as foreground.



(a) One of the frames with the canoe.



(b) One of the frames without the canoe.

Figure 4: Outputs for the 'Canoe dataset'

5 Effect of variation of parameters

The model requires manual tuning of the above parameters. The variation of the above parameters causes a lot of changes in the output of the background detection.

5.1 Effect of the learning rate (α)

Changing the value of α causes the background to aggregate and causes it to be in the next few frames as well if the value is quite high. Choosing a very small value causes the background to be very sparse. e.g. the person in run.avi(Figure 5) is not detected completely and there are a few black patches within the body.

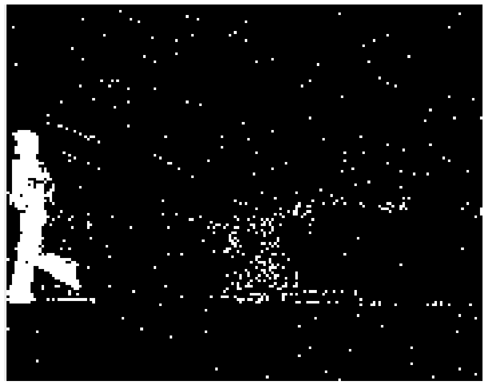


Figure 5: With $\alpha = 0.01$ there is a trail of the person left behind

5.2 Effect of the threshold : T

Increasing the value of the threshold makes whole image as the foreground. Decreasing its value causes it to detect even small variations in pixel values even if they do not belong to the foreground.

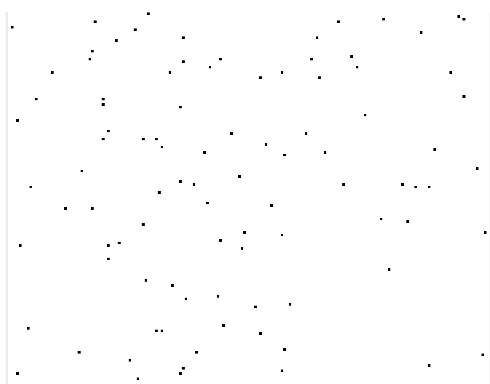


Figure 6: With $T = 1$ everything is detected as foreground

5.3 Effect of initial value of standard deviation (σ_{init})

The value of initial standard deviation also plays a role in the amount of pixels detected as background. Using a large value of makes the model treat the whole image as background and using a small value makes the model treat even small variations in pixel values as foreground.

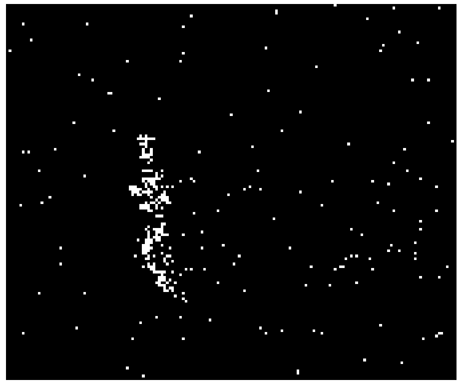


Figure 7: With $\sigma_{init} = 30$ major part of the person's body is treated as background

6 Conclusion

Overall the method if modelling pixels as mixture of Gaussians works well as it adaptively sets the background and foreground. The model does not work well in videos which have dynamic backgrounds. It can be imrpoved by adding other constraints to the model. Also a lot of parameter tuning is required to get a decent output which can be tedious.

7 References

[1] C. Stauffer, 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] Discussed the paralellisation of code with Gouthaman(T.A) and Rahul Chakwate(AE16B005)