DRA-based Background Subtraction*

Sandeep MANANDHAR[1] and Anas MHANA[2]

I. INTRODUCTION

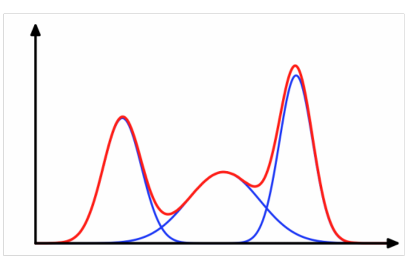
In this report, we present our understanding and implementation of method proposed in [1]. The method describes autonomous detection and tracking under illumination changes, occlusions and moving scene. But our work only focuses on detection phase which [1] accomplishes by Dyanamic Reverse Analysis(DRA) approach where background is modeled as MoGs [2] in forward and backward directions of frame sequence.
We have implemented gaussian mixture model for background modeling. The MoGs are obtained for both forward and backward sequence using Expectation Maximization on a set of grayscale images. The foreground pixels are then classified using these gaussians .Finally, the obtained results from both directional sequence are used to create similarity measure to find the change in illumination in the frames and then create a hybrid background model[1].

II. GAUSSIAN MIXTURE MODEL

Mixture model is a distribution that represents the probabilities distribution of observations in the overall population. Mixture models are normally used to make statistical set of the features of sub-populations given only some observations without knowing the identification about these sub-populations.

$$p(x) = \sum_{n=1}^{k} P_k N(x|m_k, C_k) \tag{1}$$

[3]Mixture of Gaussian is weighted sum of Gaussian densities. In the equation 1, the weight (P_k) represents the importance of each distribution. C represents the co-variance and m represents the center. Because the densities have to be integrated to one, the weights should be normalized. GMM



has been used in our implementation by considering the thickest Gaussian distribution of pixels, that corresponds to the largest variance, and represents the foreground.

1) Outline of Algorithm: To separate foreground pixels from a scene using mixture models, we need to train a set of data to fit a number of gaussians in it. In our case, we use 3 gaussian models for a background. To train the set, we use Expectation Maximization to get the parameters of the gaussians, namely, $mean(\mu)$, $variance(\sigma)$ and $weights(wt)$. It is to be noted that these gaussians are sorted in ascending order of ratio $\frac{wt}{\sigma}$. Reason being, the more dominant a model is the more wt it obtains and the lesser the variance, lesser the value of σ giving it more probability of being a background.

Considering the training is done, we initialize the classification as:

- Load the mixture modeling parameters, $\mu's$, $\sigma's$ and wts.
- Retain the first B gaussian's which exceed some threshold T.

$$M = \sum_{i-1}^{B} (wt_{i,t} > T) \tag{2}$$

  We use the weights starting from the largest weighted gaussian. Here, we do not have to go till the least weighted model because adding it makes sure the sum of weights i.e. 1 will be higher than any T we define.
- Using these models in M, check pixels distance from each $\mu's$ in M and compare against some multiple k of $\sigma's$.

$$X_{i,t} - \mu_{i,M} < k\sigma_{i,M} \tag{3}$$

  The threshold is ideally set to 2.5 but in our case because of the poor training of some sequences we had to choose higher value of k.
- In case a pixel matches with one of the B gaussians, the matched gaussian m receives an update as described in [4].

$$wt(i_{t+1}, m) = wt(i_t, m) + \alpha(1 - wt(i_t, m)) \tag{4}$$

$$\mu(i_{t+1}, m) = (1-\rho)\mu(i_t, m) + \rho X_t \tag{5}$$

$$\sigma(i_{t+1}, m)^2 =$$

$$(1-\rho)\sigma(i_t, m)^2 + \rho(X_t - \mu_(i_t, m))^T (X_t - \mu_(i_t, m)) \tag{6}$$

$where$

$$\rho = \alpha\eta(X_t|\mu_(i_t, m), \sigma(i_t, m)) \tag{7}$$

*Based on method Proposed by H. Bhaskar et al.

[1]Master in Computer Vision, Université de Bourgogne, France. manandhar.sandeep at gmail.com

[2]Master in Computer Vision, Université de Bourgogne, France. anasm87 at gmail.com

If a match is not found, the pixel is set as the mean value in least weighted model and the corresponding variance is set to be high and weight is set to be low[4],[2].

- The updated mean, variance and weights are resorted by the order of ascending ratio of $\frac{wt}{\sigma}$.
- The unmatched pixel is classified as a foreground pixel.
- Further morphological operators like open, close, dilate, etc. can be used to remove noise if present.

A. Expectation Maximization

We have implemented Expectation Maximization in Matlab for 1D and then extending it to 2D case for images. We only consider grayscale image because of lack of resource and computation required for color images. Our EM algorithm only runs for few hundred iterations regardless of convergence.

1) E-Step: In this step, we calculate likelihood of pixels belonging to a model. We test it against 3 models. In case of a constant pixel in a location through out the time, variance seems to be near 0. In such case we readily set its likelihood to 1 making a background pixel. the likelihood is:

$$q_{nk} = p(z = k|x_n) \tag{8}$$

2) M-Step: In this step, the parameters of mixture gaussian model is updated. the new updated weight is:

$$P_k = \sum_{n=1}^{k} q_{nk} \tag{9}$$

the new updated center is:

$$m_k = \sum_{n=1}^{k} q_{nk} X_n \tag{10}$$

where X_n is the data points. the new updated covariance matrix is:

$$C_k = \frac{1}{NP_k} \sum_{n=1}^{k} q_{nk}(X_n - m_k)(X_n - m_k)^T \tag{11}$$

We repeat this step until these parameters converge or for some fixed iterations.

III. Dynamic Reverse Analysis

The background modeling requires maintenance of the statistics i.e. $(\mu, \sigma, weights)$. Because of the dynamic scene and varying illumination, the model needs to adapt/update itself. The update happens as the scene flows in sequence. The key notion of DRA is that foreground pixels should be classified as such regardless of which direction the scene flows. We test a frame with model generated in forward direction as well as backward direction.

Forward direction uses model generated using frames from 1 to N to test $(N+1)^{th}$ frame whereas backward direction uses model generated using frames M to $(N+2)^{th}$ to test $(N+1)^{th}$ frame. Here, the sequence lasts till M frames.

The entire sequence comes from microsoft's wallflower database, namely $LightSwitch$ [5]. The scene is an indoor scene which has 3 changes in illumination by turning on/off the household light.

The figure 1 shows the disparity in the foreground classification of both the directional models.

Fig. 1: left: current frame; mid: from forward model; right: from backward model

A. Forward Background Subtraction

Fig. 2: 3 forward background model with increasing weights from left to right

To create forward background model, we chose the first 50 frames which do not have much illumination change. Due to the lack of resource we could not use many frames with more iterations on EM step.

The figure 2 shows the 3 background models that got updated at frame 49. Here, the learning parameter α was set to 0.3. As we can see the least weighted model has more variance than the other two. The incoming frame whose pixels were not classified as foreground got into background model. Higher the value of α, the faster the new pixels will be learned by new model. The equation 12 gives the posteriori $P^f(X_t)$ using forward model.

$$P^f(X_t) = P(X_t|M^{\xrightarrow{\epsilon_{t-1}}}) \tag{12}$$

B. Backward Background Subtraction

Fig. 3: 3 backward background model with increasing weights from left to right

The backward model has been created by using last 50 frames. The figure 4 shows the models extracted at run-time on 30^{th} test frame. The learning parameter α has been set to 0.3 here. As the sequence goes backwards, the models are updated depending upon the weights and standard deviation associated with the previous models. The equation 12 gives the posteriori $P^b(X_t)$ using backward model.

$$P^b(X_t) = P(X_t|M^{\xleftarrow{\epsilon_{t-1}}}) \tag{13}$$

C. Similarity

After the directional models are applied to classify the foreground, the results are compared. The purpose here is to find the illumination change in the sequence. The method proposed in [1] is quite vague so we accomplish this using mutual information(MI) [6]. We have obtained similarity measure for each frame using average of MI between past couple of frames and current frame in both directional models. But for the MI to be calculated, we needed to preseve the gray level value of foreground classified pixels. The graph

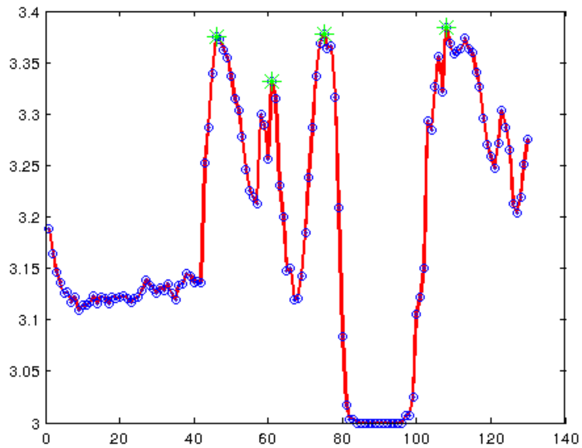


Fig. 4: 3 Similarity curve for 132 sequences of frames

4 describes a similarity curve generated from MI analysis of each frame in sequence of 132 frames with past 2 frames in each direction.

$$\chi = 2 * N - \frac{1}{N}\sum_{n=1}^{N} MI(I_n^f, I^c) + MI(I_n^b, I^c) \tag{14}$$

The equation 14 describes our similarity measure. MI is the mutual information, I_n^f is n^{th} output of forward model, I_n^b is n^{th} output of backward model. The function MI normalises the measure to 2 so we subtract if from $2 * N$ to invert the measure for plotting purpose.

The graph 4 has few peaks which shows the highest degree of change in illumination. The method in [1] does not propose any peak detection scheme. Instead it speaks of landmarking them which we assume that it is done manually. We used matlab's $findpeaks()$ to locate the peaks in the humps and consider first few high peaks above a threshold. The first peak in figure 4 depicted in $*$ corresponds to the 46^{th} frame. Reason beng, in the immediate next frame, a person occludes the scene in significant manner. Similarly, the next peak at 62^{nd} frame is due to the lights being turned off. The 3^{rd} peak happens at 71^{st} frame when the person returns back in the dark sequence and occludes the scene again. The peaks around 4^{th} peak fluctuates because around 108^{th} a person enters the room increasing intensity slowly from one direction and suddenly turning on the light.

IV. Hybrid model

After analysing the similarity curve, we can see where the illuminations have changed. To classify a frame, the method in [1] proposes a mapping function for the number

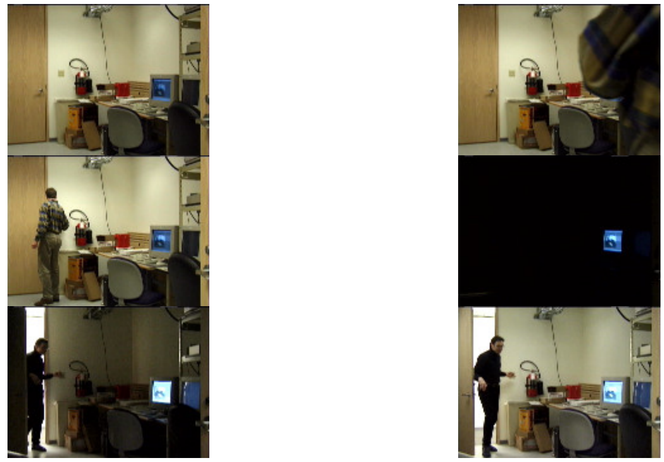


Fig. 5: from top in sequence: 46^{th} and 47^{th}(occlusion), 62^{nd} and 63^{rd}
(lights off), 109^{th} and 110^{th}(lights on)

of frames in forward and backward direction but fails to describe it mathematically. So we experimented with the number of frames to create a hybrid model. The method outlines that for each new frame, few frames from both direction is used to train MoGs for that particular frame, but it seems computationally exhaustive if we do it for each frame. While we do not have any method to perform this, we have arbitrarily chosen frames around the frame with drastic illumination change and tested if the results come up with better classification. For a scene transiting from low to high illumination condition, we chose to keep more of the forward frames with more illumination and few past frames with less illumination. But the results were not as expected. Then we proceeded to keep all the frames between $1st$ and $4th$ peak to obtain following result.

V. RESULT

Fig. 6: forground classification in backward model, forward model and hybrid model from left to right

Fig. 7: result of each frames using backward model(frames proceed in top to bottom order)

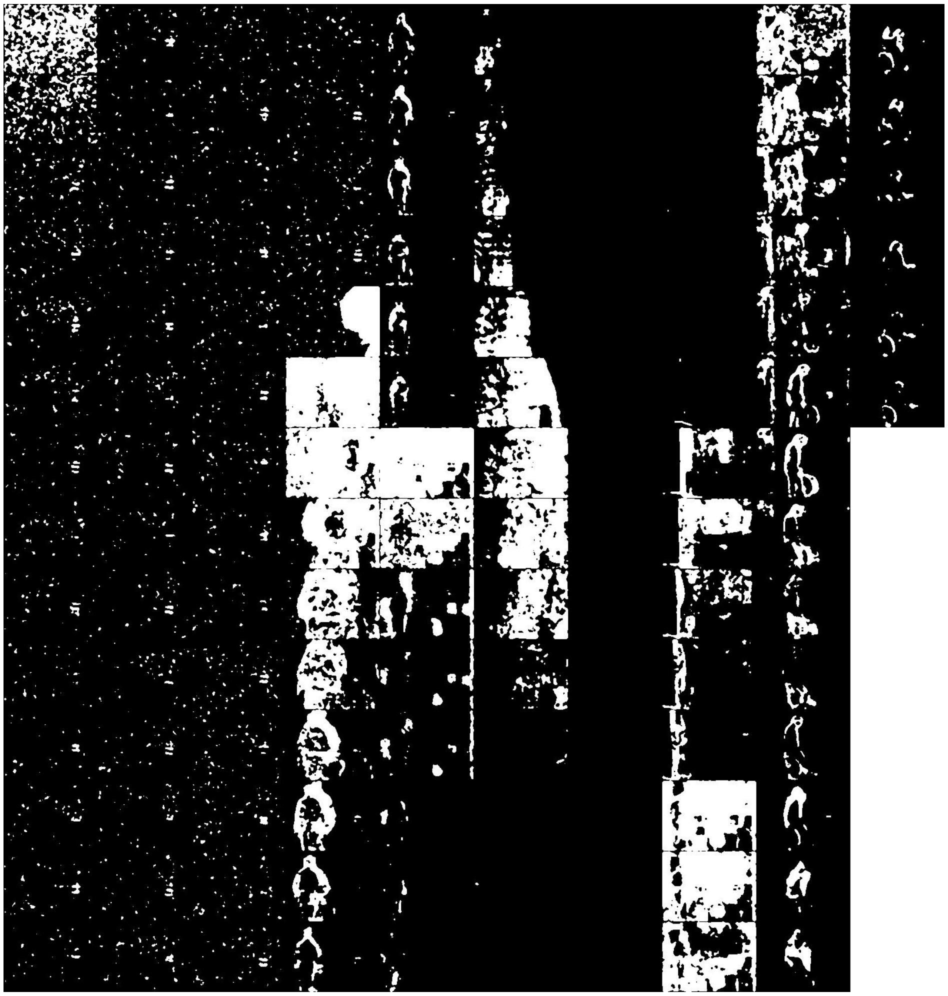


Fig. 8: result of each frames using forward model(frames proceed in top to bottom order)

VI. CONCLUSION

The method proposed in [1] seems promising but since it cannot explain much of the mathematics behind similarity measure and mapping of required frames, we had to improvise. The results were not very impressive given the resources we had to use for training the MoGs. We have used the $LightSwtich$ sequence from Microsoft wallflower database [5] but we could not obtain the ground truth set so the results went unverified.

REFERENCES

[1] D. P. D. M. A.-M. L. M. Harish Bhaskar, Kartik Dwivedi, "Autonomous detection and tracking under illumination changes, occlusions and moving camera," 2015.

[2] "Background subtraction." https://en.wikipedia.org/wiki/Background_subtraction. Accessed: 2016-01-08.

[3] . Jakob Verbeek, December 6, "Clustering with k-means and gaussian mixture distributions."

[4] W. G. Chris Stauffer, "Adaptive background mixture models for real-time tracking,"

[5] "Microsoftwallflower database." http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm. Accessed: 2016-01-08.

[6] "Mutual information in images, matlab implementation." http://www.mathworks.com/matlabcentral/fileexchange/4145-automatic-image-registration-using--normalized--mutual-information-for-users-of-ip-toolbox. Accessed: 2016-01-08.