

Detection of moving objects in videos using various Gaussian Mixture Models

P.Kabithapriyan and P.Sharon Femi

Abstract—To detect the moving objects in videos, the background subtraction has to be employed. Background subtraction is a process of extracting foreground objects in a particular scene. The most efficient algorithm for performing background subtraction is the Gaussian Mixture model (GMM). Gaussian mixture models are probabilistic algorithms used to perform background subtraction. The various GMM algorithms differ in the way they process each pixel and the procedure to update the model. The variants of GMM include simple GMM, Mixture of Gaussians, Fuzzy GMM, Adaptive SOM and Fuzzy adaptive SOM. The idea is to present a comparative evaluation of the results produced from the five different Gaussian Mixture models for live videos.

Index Terms— Background Subtraction, Gaussian Model, Fuzzy Gaussian Model, Self- Organizing Map

1 INTRODUCTION

THE need for video surveillance has grown tremendously in many areas, to maintain social control, recognize and monitor threats and prevent/investigate criminal activity. In addition to security applications, video surveillance is used to measure traffic flow, detect accidents in highways and in military applications. It alerts the security officers of a burglary in progress or a suspicious individual loitering in a restricted area helping to avoid threat. Detection of objects plays an important role in surveillance system. The objects that are introduced in the foreground have to be detected in time, so as to avoid dangerous situations. Identifying moving objects from a video sequence is a rudiment task for many computer-vision applications. A common approach is to perform background subtraction, which detects the foreground objects from the portion of video frame that differs from the background model.

Background modeling is used in numerous applications to model the background and detect moving objects in the scene as in video surveillance. It is a key step of background subtraction methods for use with static cameras. The simplest background modelling involves acquiring the background image with no moving object so that image subtraction can be done to determine the moving objects. But, the problem is that the background cannot be obtained when dynamic changes occur under situations like illumination changes, camera jitter and movement in the background. The movement in the background may be either objects being introduced or removed from the scene. The variants of GMM uses the following steps for background subtraction: Background modeling, background initialization, background maintenance and foreground detection.

A good background model should react to quick changes in background and adapt itself so as to accommodate changes occurring in the background. To be robust and adaptable, many background modelling methods have been developed. The background subtraction models should have a good foreground detection rate and should be capable of operating in real time.

The rest of this paper is organized as follows: Chapter 2 presents the simple Gaussian model with its parameters. Chapter 3 presents the pixel-wise Mixture of Gaussians along with the parameters. Chapter 4 presents the fuzzy Gaussian model. Chapter 5 presents the adaptive SOM model with its parameters. Chapter 6 presents the fuzzy and adaptive SOM model. Chapter 7 presents the experimental results of the various background models.

2 SIMPLE GAUSSIAN MODEL

The background is modeled by a single multivariate Gaussian probability density function based on pixel values of the source image [1]. Each pixel of the image is modeled using the Gaussian pdf given by

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

Where,

X_t is the pixel value at time t ,

μ is the mean background color and

Σ is the covariance matrix of pixel at time t

The pixels at each frame are classified as foreground or background by calculating the Mahalanobis distance between the source and background model pixels, and comparing this distance to a threshold.

The distance is calculated as,

$$D = \sqrt{(X_{i,t+1} - \mu_{i,t})^T - \Sigma_{i,t}^{-1} (X_{i,t+1} - \mu_{i,t})} < k\sigma_{i,t} \quad (2)$$

where k is set to 2.

The mean and covariance matrix of the Gaussian at each pixel is continuously updated as given below:

- P. Kabithapriyan is currently pursuing undergraduate degree program in Information Technology in Sri Venkateswara College of Engineering, India, PH-9790098897. E-mail: kabithapriyan@gmail.com
- P.Sharon Femi is currently working as Assistant Professor in the Department of Information Technology in Sri Venkateswara College of Engineering, India, PH-9551597650. E-mail: sharon@svce.ac.in

$$\mu_{i,t+1} = (1 - \alpha) \mu_{i,t} + \alpha X_{t+1} \quad (3)$$

$$\Sigma_{i,t+1} = (1 - \alpha) \Sigma_{i,t} + \alpha (X_{t+1} - \mu_{i,t}) \cdot (X_{t+1} - \mu_{i,t})^T \quad (4)$$

where α is the constant learning rate

This model can detect moving objects with simple static backgrounds in controlled lighting situations. Due to the blind update method employed, the model is less sensitive to initial conditions but stationary foreground objects are eventually incorporated into the background.

The various parameters used in simple Gaussian are Learning Rate - The rate at which it adapts to changes in the video image. High values make the model adapt quickly to scene changes.

Sensitivity - Determines the sensitivity to changes in the background. Low values enhance the detection of objects in the scene, but also make the model more sensitive to noise.

Noise Variance - Sets the minimum value of the variance for the Gaussian model. Higher values are good for videos with noisy images.

3 MIXTURE OF GAUSSIANS

Mixture of Gaussians implements a classic multivariate Gaussian mixture model where every pixel is represented by a mixture of four Gaussian distributions [2]. The modelling of the Gaussians is based on the Mahalanobis distance between the source and background model pixels. This model is designed to handle multimodal backgrounds with moving objects and illumination changes.

In Mixture of Gaussians, each pixel is characterized by its intensity in the RGB color model. The various steps involved in MoG are explained in the following subsections.

3.1 Pixel Characterization

The probability of each pixel value X_t is calculated as

$$P(X_t) = \sum_{i=1}^k w_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (1)$$

where,

k is the number of Gaussians (value may be 3 -5)

$w_{i,t}$ is the weight associated to the Gaussian i at time t

X_t is the pixel value at time t ,

μ is the mean value of the i th Gaussian distribution

Σ is the covariance matrix

η is the Gaussian probability density function defined as below:

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

n is the dimension of the intensity at the pixel X .

Each pixel has the same covariance matrix and is of the form $\Sigma_{i,t} = \sigma_{i,t}^2 I$ and thus each pixel is characterized by a mixture of k Gaussians.

3.2 Parameter Initialization

The various parameters involved in Mixture of Gaussians are $k, \Sigma, w_{i,t}$. In our system, k is set to 4, Σ is initialized to 50 and $w_{i,t}$ is initialized as in equation 3.

$$w_{i,t} = (1 - \alpha) w_{i,t} + \alpha \quad (3)$$

where α is the learning rate set to 0.001. The mean and covariance matrix of the Gaussian at each pixel is continuously updated.

3.3 Foreground Detection

Initial foreground detection is made and the parameters are updated. Initial foreground detection is made by determining the ratio $r = w/\sigma$ and order the Gaussians following this ratio. The first B distributions are considered as the background model, where

$$B = \arg \min_b \sum_{i=1}^b w_{i,t} > T \quad (4)$$

This ensures that a high weight with a weak variance refers to a background pixel. The other distributions are considered to present a foreground distribution. The pixels at each frame are classified as foreground or background by calculating the Mahalanobis distance between the source and background model pixels, and comparing this distance to a threshold.

When a new frame enters at time $t+1$, a matching test is performed for every pixel. The Mahalanobis distance between the source and background model pixels are calculated using the formula,

$$\text{Dist} = \text{sqrt}((X_{i+1} - \mu_{i,t})^T - \Sigma_{i,t}^{-1} (X_{t+1} - \mu_{i,t})) < k \sigma_{i,t} \quad (5)$$

where,

k and T are the threshold set to 2.5 and 0.5 respectively.

Two cases may occur as a result of the matching test:

Case 1: Match found with one of the k Gaussians.

In this case, if the Gaussian identified is one among the B distributions, the pixel is classified as background, else it is foreground pixel.

Case 2: No match with any of the k Gaussians

In this case, the pixel is identified as foreground. To proceed for the next foreground detection, the parameters must be updated.

Two cases occur in the foreground detection as below:

Case a: A match found with one of the k Gaussians.

The updation of values for the matched component is as follows

$$w_{i,t} = (1 - \alpha) w_{i,t} + \alpha, \quad (6)$$

where α is the constant learning rate

$$\mu_{i,t+1} = (1 - \rho) \mu_{i,t} + \rho X_{t+1} \quad (7)$$

$$\sigma_{i,t+1}^2 = (1 - \rho) \sigma_{i,t}^2 + \rho (X_{t+1} - \mu_{i,t+1}) \cdot (X_{t+1} - \mu_{i,t+1})^T \quad (8)$$

where

$$\rho = \alpha \cdot \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{9}$$

For the unmatched component, the μ and Σ remains unchanged and only the weight is updated as

$$w_{j,t+1} = (1 - \alpha) w_{j,t} \tag{10}$$

Case b: No match with any of the k Gaussians

In this case, the distribution k is replaced with the parameters

$$w_{k,t+1} = \text{low prior weight} \tag{11}$$

$$\mu_{i,t+1} = X_{t+1} \tag{12}$$

$$\sigma_{k,t+1}^2 = \text{large initial variance} \tag{13}$$

Once the parameters maintenance is made, foreground detection can be made and so on. The blind update employed by the method makes it less sensitive to initial conditions but tends to integrate stationary foreground objects into the background.

4 FUZZY GAUSSIAN MODEL

To handle dynamic changes in the background, fuzzy concepts have been recently introduced in the different steps of background subtraction method. [3]

Fuzzy Background modelling:

In GMM, the initialization is done using a training sequence, which may be either noisy or insufficient. This may cause false classification in the foreground detection mask. To overcome this, descriptions for uncertain parameters in GMM are introduced using a Type-2 Fuzzy Gaussian Mixture models. So, we generate the multivariate Gaussian with uncertain mean vector called T2-FMOG-UM defined as,

$$\eta(X_t, \tilde{\mu}, \Sigma) = \frac{1}{(2\pi)^{3/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} \left(\frac{X_{t,c} - \mu_c}{\sigma_c}\right)^2\right) \tag{1}$$

where ,

$$\tilde{\mu}_c \in [\underline{\mu}_c, \overline{\mu}_c] \text{ and } c \in \{R, G, B\}$$

and multivariate Gaussian with uncertain variance vector called T2-FMOG-UV

$$\eta(X_t, \mu, \tilde{\Sigma}) = \frac{1}{(2\pi)^{3/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} \left(\frac{X_{t,c} - \mu_c}{\sigma_c}\right)^2\right) \tag{2}$$

where

$$\tilde{\sigma}_c \in [\underline{\sigma}_c, \overline{\sigma}_c], \quad c \in \{R, G, B\} \text{ and } \tilde{\Sigma} \text{ denote uncertain covariance matrix.}$$

The factor k_m and k_v control the intervals in which the parameter vary as given below:

$$\underline{\mu} = \mu - k_m \sigma, \quad \overline{\mu} = \mu + k_m \sigma$$

$$\underline{\sigma} = k_v \sigma, \quad \overline{\sigma} = (1/k_m) \sigma$$

Both T2-FMOG-UM and T2-FMOG-UV are used to model the background

Training:

Training the T2-FGMM involves estimating the parameters

μ, Σ and the factors k_m, k_v . The parameter estimation includes 3 steps as follows:

- i. Select k value between 3 and 5.
- ii. Estimate the GMM parameters.
- iii. Add the factor k_m or k_v so as to produce T2-FGMM-UM or T2-FGMM-UV.

Foreground detection can be processed once the training is done.

Fuzzy foreground detection:

Initial foreground detection is made by determining the ratio $r = \omega/\sigma$ and order the Gaussians following this ratio. The first B distributions are considered as the background model, where

$$B = \arg \min_b \sum_{i=1}^b \omega_{i,t} > T \tag{3}$$

The pixels at each frame are classified as foreground or background by calculating the log-likelihood between the source and background model pixels, given by the following formula and comparing this distance to a threshold.

$$H(X_t) = |\ln(\underline{h}(X_t)) - \ln(\overline{h}(X_t))| \tag{4}$$

When a new frame enters at time t+1, a matching test is performed for every pixel. The log-likelihood length between the source and background model pixels are compared to a threshold as

$$H(X_t) < k\sigma \tag{5}$$

where, k equal to 2.5

Two cases may occur as a result of the matching test:

Case 1: Match found with one of the k Gaussians.

In this case, if the Gaussian identified is one among the B distributions, the pixel is classified as background, else it is foreground pixel.

Case 2: No match with any of the k Gaussians

In this case, the pixel is identified as foreground. To proceed for the next foreground detection, the parameters must be updated.

Fuzzy Background Maintenance:

The idea is to update the background following the membership of the pixel at the class background or foreground. This membership comes from the fuzzy foreground detection. This fuzzy adaptive background maintenance allows dealing robustly with illumination changes and shadows. The T2-FGMM maintenance is done same as that of GMM.

Two cases occur in the foreground detection as below:

Case a: A match found with one of the k Gaussians.

The updation of values for the matched component is as follows

$$\omega_{i,t} = (1 - \alpha) \omega_{i,t} + \alpha, \tag{6}$$

where α is the constant learning rate

$$\mu_{i,t+1} = (1 - \rho) \mu_{i,t} + \rho X_{t+1} \quad (7)$$

$$\sigma_{i,t+1}^2 = (1 - \rho) \sigma_{i,t}^2 + \rho (X_{t+1} - \mu_{i,t+1}) \cdot (X_{t+1} - \mu_{i,t+1})^T \quad (8)$$

Where

$$\rho = \alpha \cdot \eta (X_t, \mu_{i,t}, \Sigma_{i,t}) \quad (9)$$

For the unmatched component, the μ and Σ remains unchanged and only the weight is updated as

$$\omega_{j,t+1} = (1 - \alpha) \omega_{j,t} \quad (10)$$

Case b: No match with any of the k Gaussians

In this case, the least probable distribution k is replaced with the parameters

$$\omega_{k,t+1} = \text{low prior weight} \quad (11)$$

$$\mu_{i,t+1} = X_{t+1} \quad (12)$$

$$\sigma_{k,t+1}^2 = \text{large initial variance} \quad (13)$$

Once the parameters maintenance is made, foreground detection can be processed and so on.

5 ADAPTIVE SOM

To detect moving objects in extremely sensitive dynamic environment, it is desirable to construct a background model that automatically adapts to changes in a self-organizing manner and without a prior knowledge. The idea is to adaptively model the background using a competitive neural network similar to Self Organizing Map (SOM).[4] For each pixel, a neuronal map consisting of 3×3 weight vectors is defined. The incoming source pixels are mapped to the weight vector that is closest according to a Euclidean distance metric, and the weight vectors in its neighbourhood are updated. The set of weight vectors act as a background model that is used for background subtraction in order to identify foreground pixels.

Adaptive SOM involves two major steps:

i) Initial Background model

This step initializes the weight vector of the network. The first image of the video sequence is used for the approximation of the background and so, for each pixel, the corresponding weight vectors are initialized with the pixel value. The complete set of weight vectors for all pixels of an image with N rows and M columns is represented as a neuronal map with $n \times N$ rows and $n \times M$ columns, where the weight vectors for the generic pixel (x,y) of I are at neuronal map positions (i,j) , $i = n * x, \dots, n * (x+1) - 1$ and $j = n * y, \dots, n * (y+1) - 1$.

ii) Subtraction and Update of Background model

After initialization, each incoming pixel p_t of the t^{th} sequence frame I_t is compared to the current pixel model C to check whether there exists a weight vector that matches it. If a best matching weight vector c_m appears, it means that p_t belongs to the background and it is used as the pixel encoding approximation. Else, if no matching weight vector exists, p_t should not be used to update the weight vectors. It may be either a background or moving object.

The above described background subtraction and update

procedure for each pixel is given in the following algorithm.

Algorithm

Input: pixel value p_t in frame I_t , $t=1, \dots, \text{LastFrame}$

Output: pixel p_t is background/foreground

- 1) Initialize model C for pixel p_t and store it into A
- 2) for $t=1$ to LastFrame
- 3) Find best match c_m in C to current sample p_t
- 4) if (c_m found) then
- 5) p_t is background
- 6) update A in the neighbourhood of c_m
- 7) else if (p_t shadow) then
- 8) p_t is background
- 9) else
- 10) p_t is foreground

This algorithm includes two phases:

- a) calibration phase - This comprises of steps 1-6, which is executed for first few sequence frames K
- b) online phase- This comprises of steps 2-10, which is executed for the LastFrame - K sequence frames.

K depends on the number of initial static frames available for each sequence.

Finding best match c_m in C to current sample p_t

To find the best match weight vector c_m , the Euclidean distance between two pixels are determined. The weight vector c_m , for some m, gives the best match for the incoming pixel p_t if its distance from p_t is minimum in the model C of p_t and is no greater than a fixed threshold

$$d(c_m, p_t) = \min_{i=1, \dots, n^2} d(c_i, p_t) \leq \epsilon \quad (1)$$

The threshold allows to distinguish between foreground and background pixels, and is chosen as,

$$\epsilon = \begin{cases} \epsilon_1, & \text{if } 0 \leq t \leq K \\ \epsilon_2, & \text{if } t > K \end{cases} \quad (2)$$

with ϵ_1 and ϵ_2 small constants.

Update A in the neighbourhood of c_m

If a best match c_m is found for current sample p_t , the weight vectors in the $n \times n$ neighborhood of c_m are updated using the selective weighted running average given by,

$$A_t(i,j) = (1 - \alpha_{i,j}(t)) A_{t-1}(i,j) + \alpha_{i,j}(t) p_t(x,y) \quad (3)$$

The model can handle scenes containing multimodal backgrounds with moving objects and gradual illumination changes. It employs a selective updating procedure that prevents the inclusion of stationary foreground objects into the background. In order to work properly a good initial background is recommended for training.

6 FUZZY ADAPTIVE SOM

Fuzzy adaptive SOM is a modified version of adaptive SOM that uses a fuzzy rule to update the neural network background model.[5] It includes spatial coherence in terms of the intensity difference between locally contiguous pixels. This shows that, neighboring pixels with small intensity differences are coherent,

while neighboring pixels with high intensity differences are incoherent. The algorithm is formulated as a fuzzy rule-based procedure, where fuzzy functions are computed, pixel-by-pixel, on the basis of the background subtraction phase and combined through the product rule. The fuzzy updating of the background helps to make the model more robust to illumination changes in the scene.

Let N_p be the spatial square neighbourhood of pixel $p \in I$ having fixed width $k \in N$ and let Ω_p be the set of pixels belonging to N_p that have a best match in the background model. The neighbourhood coherence factor is defined as,

$$NCF(p) = \frac{|\Omega_p|}{|N_p|} \tag{1}$$

This factor gives a relative measure of the number of pixels belonging to the spatial neighbourhood N_p of a given pixel p that are well represented by the background model. If $NCF(p) > 0.5$, most of the pixels in such spatial neighbourhood are well represented by the background model, and this should imply that also pixel p is well represented by the background model.

Fuzzy rule

The background updating rule may be formulated in terms of a production rule of the type:

if (condition) then (action).

When the condition in the production rule is satisfied, the action is performed. For example, in a rule-based outdoor scene understanding system, a typical rule may be: if (a region is rather straight and highly uniform and the region is surrounded by a field region) then (confidence of road is high). The flexibility and power provided by fuzzy set theory for knowledge representation makes fuzzy rule-based systems very attractive when compared with traditional rule-based systems.

In detecting moving objects, the uncertainty resides in determining suitable thresholds in the background model. According to this way of reasoning, the fuzzy background subtraction and update algorithm for the generic pixel $p_t \in I_t$ can be stated through a fuzzy rule-based system as follows:

Fuzzy rule-based background subtraction and update algorithm
if ($d(c_m(p_t), p_t)$ is Low) and ($NCF(p_t)$ is Low) then
Update \tilde{B}_t
Endif

Let $F1(p_t)$ be the fuzzy membership function of $d(c_m(p_t), p_t)$ to the fuzzy set Low and $F2(p_t)$ be the fuzzy membership function of $NCF(p_t)$ to the fuzzy set Low.

$$F_1(p_t) = \begin{cases} 1 - \frac{d(c_m(p_t), p_t)}{\epsilon} & \text{if } d(c_m(p_t), p_t) < \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and

$$F_2(p_t) = \begin{cases} 2NCF(p_t) - 1 & \text{if } NCF(p_t) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The fuzzy rule becomes:

$$\alpha_{i,j}(t) = F1(p_t)F2(p_t)\alpha(t)w_{i,j} \tag{4}$$

Fuzzy algorithm

The proposed fuzzy background subtraction and update procedure can be stated as follows.

Given an incoming pixel value p_t in sequence frame I_t , the estimated background model \tilde{B}_t is obtained through the following algorithm:

Fuzzy background subtraction and update algorithm
Initialize weight vectors $C(p_0)$ for pixel p_0 and store it into \tilde{B}_0
for $t=1$ to LastFrame
 Find best match c_m in C to current sample p_t
 Compute learning factors $\alpha_{i,j}(t)$
 Update \tilde{B}_t in the neighborhood of cm
Endfor

7 COMPARITIVE RESULTS

The comparative results for detecting moving objects using the various GMM models studied for background subtraction are presented in Fig 1. It has been observed from the implementation that the simple GMM works in a simple background with no light effects. The Mixture of Gaussians works well in outdoor scenarios. The Fuzzy GMM can handle dynamic changes in the background. Both SOM and adaptive SOM, works for multimodal backgrounds.

The parameter evaluations for the various GMM models are summarized in Table 1.

TABLE 1
 PARAMETER EVALUATION

MODEL	FEATURES
Simple Gaussian	It works for simple background and does not work when the scene is dynamic.
Mixture of Gaussians	This works for dynamic outdoor scene and gives more false positives in indoor scenes where the illumination changes are drastic.
Fuzzy GMM	Fuzzy concepts are introduced in different steps of background subtraction method to handle dynamic changes in the background.
Adaptive SOM	It works for multimodal backgrounds with moving objects and gradual illumination changes. A good initial background is essential for training.
Fuzzy adaptive SOM	This algorithm is a variant of adaptive SOM which incorporates spatial coherence of the pixels. The fuzzy updating of the background helps to make the model more robust to illumination changes in the scene.

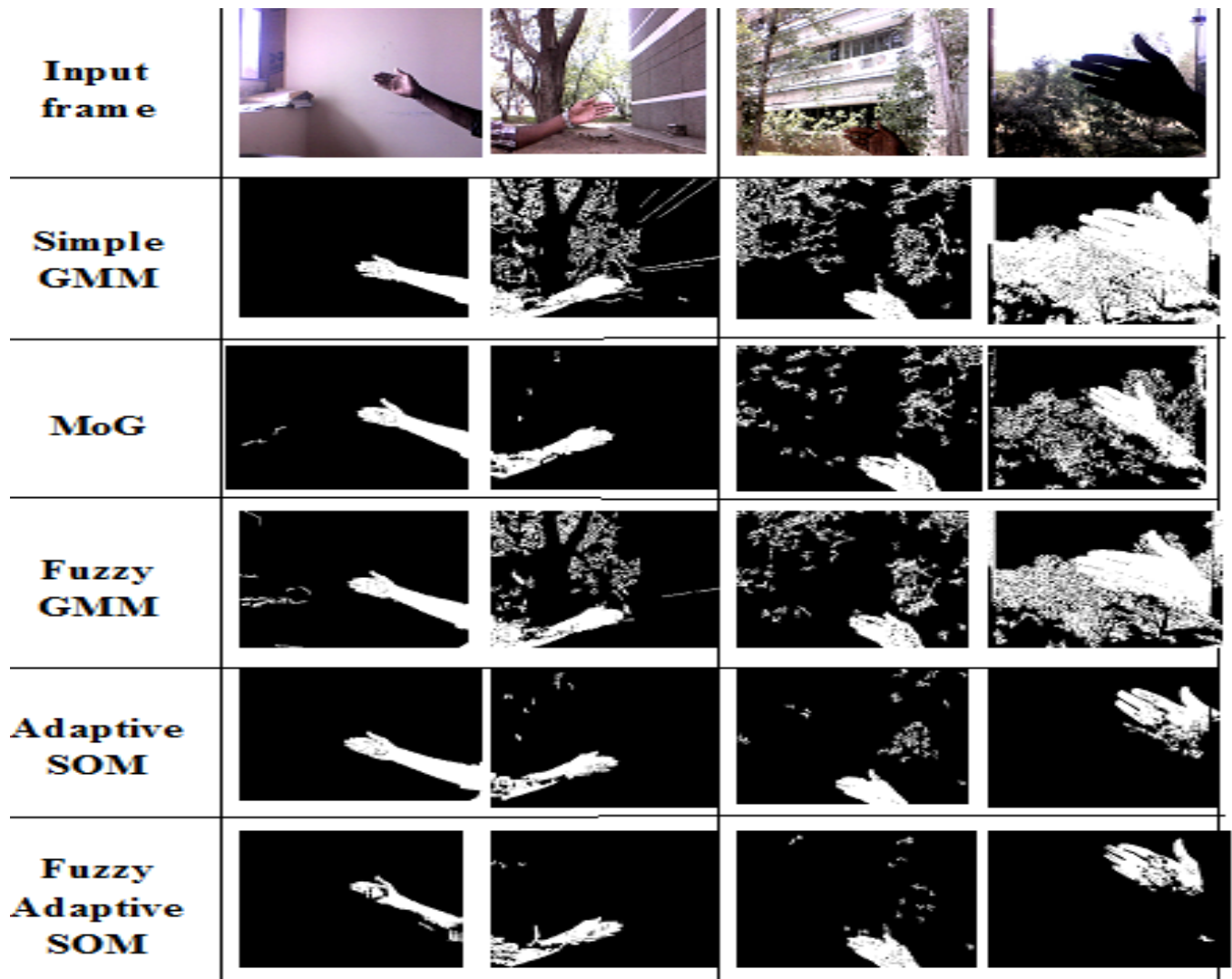


Fig 1: Detection of moving objects using various GMM Models-
 The first row represents the input frame and the remaining rows represent the output for the various GMM models

8 CONCLUSION

The detection of moving objects in live videos has been implemented using various GMM background subtraction methods. A detailed comparative evaluation of the five different models namely simple GMM, Mixture of Gaussians, Fuzzy GMM, Adaptive SOM and Fuzzy adaptive SOM is discussed in this paper. The results of the analysis reveal that the models are effective in the specific features described by the algorithm for live videos.

REFERENCES

- [1] Y. Benezeth, P.M. Jodoin, B. Emile, H. Laurent, C. Rosenberger "Review and Evaluation of Commonly - Implemented Background Subtraction Algorithms" 2008 Proc. International Conference on Pattern Recognition, pp. 1-4.
- [2] T. Bouwmans, F. El Baf, B. Vachon "Background Modeling using Mixture of Gaussians for Foreground Detection" - A Survey, Recent Patents on Computer Science 1, 3 (2008) 219-237"
- [3] T. Bouwmans, F. El Baf "Modeling of Dynamic Backgrounds by Type-2 Fuzzy Gaussian Mixture Models", 2010 MASAUM Journal of Basic and Applied Sciences 1, 2 (0000) 265-276"
- [4] Lucia Maddalena and Alfredo Petrosino "A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications" IEEE Transactions On Image Processing, Vol. 17, No. 7, July 2008
- [5] Lucia Maddalena Alfredo Petrosino "A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection" June 2009 Springer-Verlag Neural Comput & Applic (2010) 19:179-186.

[6]