

Review of background subtraction methods using Gaussian mixture model for video surveillance systems

Kalpana Goyal¹ · Jyoti Singhai¹

Published online: 30 January 2017
© Springer Science+Business Media Dordrecht 2017

Abstract Foreground detection or moving object detection is a fundamental and critical task in video surveillance systems. Background subtraction using Gaussian Mixture Model (GMM) is a widely used approach for foreground detection. Many improvements have been proposed over the original GMM developed by Stauffer and Grimson (IEEE Computer Society conference on computer vision and pattern recognition, vol 2, Los Alamitos, pp 246–252, 1999. doi:[10.1109/CVPR.1999.784637](https://doi.org/10.1109/CVPR.1999.784637)) to accommodate various challenges experienced in video surveillance systems. This paper presents a review of various background subtraction algorithms based on GMM and compares them on the basis of quantitative evaluation metrics. Their performance analysis is also presented to determine the most appropriate background subtraction algorithm for the specific application or scenario of video surveillance systems.

Keywords Background Subtraction · Background modeling · Gaussian Mixture Model · Foreground detection · Video surveillance

1 Introduction

Analysis and understanding of video sequences is an active research field. The first step in many computer vision applications like smart video surveillance, traffic monitoring and analysis, human detection and tracking, automatic sports video analysis, and gesture recognition in human-machine interface etc. is to detect the moving objects called foreground objects in the scene. Precise localization of foreground objects in a video is a fundamental and critical task for such applications. A common method to identify foreground objects is background subtraction. In background subtraction method, each video frame is compared with a background model and the pixels whose intensity values deviate significantly from the background model are considered as foreground.

✉ Kalpana Goyal
kalpana.goyal19@gmail.com

¹ MANIT, Bhopal, MP, India

Accurate foreground detection for complex visual scenes in real time is a difficult task because real-world video sequences contain several critical situations. Some key challenges experienced in background subtraction methods are: dynamic background, moving shadows of moving objects, sudden illumination changes, camouflage, foreground aperture, noisy image, camera jitter, bootstrapping, camera automatic adjustments, paused and slow moving objects (Bouwman 2014).

Many researchers have proposed various background subtraction methods and different types of background models (image of background which does not include any moving object) to deal with different challenges. Some commonly used video background modeling approaches are basic models, background estimation, background clustering, subspace learning, kernel density estimation and Gaussian Mixture Model (GMM).

In basic Models, the background is modeled using an average (Power and Schoonees 2002), a median (McFarlane and Schofield 1995) or an histogram analysis over time (Zheng et al. 2006). Once the model is computed, pixels of the current image are classified as foreground by thresholding the difference between the background image and the current frame. These methods require a large memory and a training period where foreground objects are present. Difference of consecutive frames (Roy et al. 2010) is also used to detect the foreground objects. These simple techniques are very fast, but their performance is poor in complex background scenes (Bouwman 2011) which is a major challenge in videos used in surveillance systems.

Background estimation methods predict the position, value or orientation of pixels or block from a previous frame. Any pixel of the current frame that deviates significantly from its predicted value is classified as foreground. Wiener filter (Toyama et al. 1999), Kalman filter (Messelodi et al. 2005), Tchebychev filter (Chang et al. 2004), particle filters (Zhang et al. 2006), and optical flow (Yu et al. 2011) are used for background estimation. These models are good for the scenes having gradual illumination changes. But due to their high complexity and high sensitivity to noise, these models are not suitable for real-time applications. Their foreground-background discriminant ability is also limited.

Cluster models assume that each pixel in the frame can be represented temporally by clusters. Incoming pixels are compared against the corresponding cluster group and if a match occurs, those pixels are classified as background. The clustering approach consists of K-means algorithm (Butler et al. 2003) and Codebook model (Ilyas et al. 2009). However, these models seem well adapted to deal with dynamic backgrounds and noise from video compression but cannot deal with a scene which contains sudden illumination changes, slow-moving objects and moving shadows (Shah et al. 2014). These methods fail when foreground objects have similar color to background and when the illumination variations occur because these methods only use the pixel color or intensity information to detect foreground objects (Wang and Pan 2011).

Statistical models are the mostly used models due to a good compromise between their performance and their computation cost. Subspace learning (Bouwman 2009), kernel density estimation (Elgammal et al. 2000) and GMM (Stauffer and Grimson 1999) are three main statistical modeling techniques.

Subspace learning techniques used for video background models are principal component analysis, independent components analysis and matrix factorization. The subspace learning models perform better for illumination changes. But, these models do not perform well in scenes containing irregular quasi-periodic background motions and moving shadows and have a high processing time (Bouwman 2009).

The kernel density estimation is a non-parametric background model which estimates the density of pixels using kernel function. It is a better representation of pixels density than

that of parametric methods because the probability density function for pixel intensity is estimated directly from the data without any assumptions about the underlying distribution. But it is computationally expensive and cannot be used in real-time applications because it requires many samples to correctly estimate the pixel density over time (Shah et al. 2014). This method performs well when the background is static for a significant length of time and objects are moving continuously with a consistent speed. However, they fail to handle slow-moving objects, multiple moving objects, and multi-modal backgrounds (Stauffer and Grimson 1999).

The GMM (Stauffer and Grimson 1999) is a robust method for dynamic backgrounds. It is mostly used due to its robustness to various background variations like multi-modal, quasi periodic and gradual illumination changes. However, it has a number of limitations, including slow recovery from failures and unsatisfactory performance for sudden illumination changes and for irregular background motions (Bouwmans and Baf 2010). Furthermore, it is a parametric model and parameters has to be tuned which makes the procedure complex and time consuming that makes it less attractive for real-time applications (Figueiredo and Jain 2002; White and Shah 2007). Moreover, the accuracy in selection of these parameters affects the efficiency of the system. The parameters of concern are: the number of components (K), learning rate (α) and classification threshold (T). Many researchers have proposed improvements in GMM to overcome some challenges faced by videos used in surveillance systems. In this paper, review on various modifications suggested in GMM are discussed in detail.

Previously, various survey papers for background subtraction techniques have been presented based on implementation techniques or applications. In 2004, Piccardi (2004) presented a review on implementation techniques of background subtraction based on speed, memory requirement and accuracy and compares the complexity of the different methods. Cheung and Kamath (2004) presented a review of background subtraction algorithms based on recursive and non-recursive techniques. Bouwmans (2014) and Sobral and Vacavant (2014) presented a survey of background subtraction techniques based on mathematical tools used. Bouwmans et al. (2008) presented a survey of background subtraction techniques based on GMM which classifies improvements proposed in GMM (Stauffer and Grimson 1999) in various algorithms. But none of them critically analyze the GMM based background subtraction algorithms to deal with various challenges associated with video surveillance systems. In this paper, a review of different background subtraction algorithms based on GMM has been presented with their brief description, comparative analysis and scope to improve them.

Section 2 describes the various background subtraction algorithms using Gaussian Mixture Model while their critical analysis based on various evaluation parameters is discussed in Sect. 3. Section 4 gives the conclusion and future scope.

2 The reviewed GMM approaches

GMM models the values of a particular pixel over time as a mixture of Gaussians. Based on the persistence and the variance, Gaussians are classified as background and foreground. Pixel values that do not fit the background distributions are considered foreground until there is a Gaussian that includes them with sufficient, consistent evidence supporting it. Some of the proposed background subtraction algorithms using GMM are discussed in this section and list of these algorithms is given in Table 1.

Table 1 List of the algorithms discussed in this section

References	Name of algorithms	Features
Stauffer and Grimson (1999)	GMM	RGB
KaewTraKulPong and Bowden (2002)	Improved adaptive GMM	RGB
Lee (2005)	Effective GMM	RGB
Zivkovic and Heijden (2006)	Efficient GMM	RGB
Shah et al. (2010)	Localized adaptive learning GMM	RGB
Chan et al. (2011)	Generalized Stauffer–Grimson GMM	Dynamic texture
Shah et al. (2014)	SURF based GMM	YUV
Chen and Ellis (2014)	Self-adaptive GMM	RGB

2.1 GMM

Stauffer and Grimson (1999) used the Mixture of Gaussians (MOG) to model dynamic backgrounds. The recent history of the intensity values (in RGB color space) of each pixel X_1, \dots, X_t is modeled by a mixture of K Gaussian distribution. The probability of observing the current pixel value is given by the formula:

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

where K gives the number of Gaussian distributions, $\omega_{k,t}$ is the weight of the k_{th} Gaussian in the mixture at time t having mean $\mu_{k,t}$ and covariance matrix $\Sigma_{k,t}$ and η is a Gaussian probability density function which is given by

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

where n is the dimension of the color space.

For computational reasons, authors in Stauffer and Grimson (1999) assumed that the red, green and blue color components are independent and have the same variances. Hence, the covariance matrix is assumed to be as:

$$\Sigma_k = \sigma_k^2 I \tag{3}$$

Due to this assumption, a costly matrix inversion is avoided at the rate of some accuracy. The distribution of intensity value of each pixel is characterized by a mixture of K Gaussians. Authors in Stauffer and Grimson (1999) proposed the value of K from 3 to 5.

By using an Expectation Maximization (EM) algorithm, parameters of the Gaussian mixture model, i.e., the weight, the mean and the covariance matrix, can be initialized. But implementation of an exact EM algorithm is very costly. Hence authors used the K-means algorithm for real time consideration.

Once the parameters are initialized, the K Gaussians are ordered following the ratio ω/σ . This implies that a background pixel corresponds to a high weight with a weak variance because the background is more present in a scene than moving objects and its value is practically constant. The first B Gaussian distributions which exceed certain threshold T are

retained for a background distribution:

$$B = \arg \min_b \left(\sum_{k=1}^b \omega_k > T \right) \tag{4}$$

The other distributions correspond to foreground. Every new pixel value, X_t , is checked against the existing K Gaussian distributions, until a match is found. A pixel matches a Gaussian distribution if its value lies within 2.5 standard deviation of a distribution.

A pixel is classified as background if it matches with the Gaussian distribution which is identified as background and it is classified as foreground if it matches with the Gaussian distribution which is identified as foreground. If it does not match with any of K Gaussians then it is classified as foreground. Hence a binary mask is obtained. For next foreground detection, parameters of the Gaussian must be updated.

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{5}$$

where α is the constant learning rate and $M_{k,t}$ is 1 for the matching Gaussian components and 0 for the remaining components. The mean and variance for the unmatched components remain unchanged and for the matched component, they are updated as given below:

$$\mu_{k,t} = (1 - \rho)\mu_{k,t-1} + \rho \cdot X_t \tag{6}$$

$$\sigma_{k,t}^2 = (1 - \rho)\sigma_{k,t-1}^2 + \rho(X_t - \mu_{k,t-1})^T \cdot (X_t - \mu_{k,t-1}) \tag{7}$$

where $\rho = \alpha \cdot \eta(X_t, \mu_{k,t}, \Sigma_{k,t})$. If a pixel does not match with any of the K Gaussians, then the distribution with the least probability is replaced with the new parameters. The least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance, and low prior weight.

2.2 Improved adaptive GMM

KaewTraKulPong and Bowden (2002) presented a method to improve the adaptive background mixture model. This method is different from GMM (Stauffer and Grimson 1999) in the update equations, initialization method and the introduction of a shadow detection algorithm based on a computational color space. By reinvestigating the update equations, different equations are utilized at different phases. Due to this, system learns faster and more accurately as well as adapts effectively to changing environments.

The GMM is estimated by expected sufficient statistics update equations and then instead of comparing all frames in time t only L -recent window samples are processed which increases the processing speed of the algorithm. In the beginning, the expected sufficient statistics update equations provide a good estimate before all L samples can be collected. The accuracy of the estimate and the performance of the tracker is improved using this initial estimate. It also allows fast convergence on a stable background model. The L -recent window update equations give priority to recent data therefore the tracker can adapt to changes in the environment effectively. The online EM algorithms by expected sufficient statistics are shown below:

$$\omega_{k,t} = \omega_{k,t-1} + \frac{1}{t} \left(M_{k,t} - \omega_{k,t-1} \right) \tag{8}$$

$$\mu_{k,t} = \mu_{k,t-1} + \frac{M_{k,t}}{\sum_{i=1}^t M_{k,i}} \left(X_t - \mu_{k,t-1} \right) \tag{9}$$

$$\Sigma_{k,t} = \Sigma_{k,t-1} + \frac{M_{k,t}}{\sum_{i=1}^t M_{k,i}} \left((X_t - \mu_{k,t-1})(X_t - \mu_{k,t-1})^T - \Sigma_{k,t} \right) \tag{10}$$

The online EM algorithms by L -recent window version are given below:

$$\omega_{k,t} = \omega_{k,t-1} + \frac{1}{L} \left(M_{k,t} - \omega_{k,t-1} \right) \tag{11}$$

$$\mu_{k,t} = \mu_{k,t-1} + \frac{1}{L} \left(\frac{M_{k,t} X_t}{\omega_{k,t}} - \mu_{k,t-1} \right) \tag{12}$$

$$\Sigma_{k,t} = \Sigma_{k,t-1} + \frac{1}{L} \left(\frac{M_{k,t} (X_t - \mu_{k,t-1})(X_t - \mu_{k,t-1})^T}{\omega_{k,t}} - \Sigma_{k,t-1} \right) \tag{13}$$

For identifying moving shadows, a color model is used that can separate chromatic and brightness components. This is done by comparing a non-background pixel against the current background components. If the difference between chromatic and brightness components lies within some threshold range, the pixel is considered as a shadow. This color model consists of a position vector at the RGB mean of the pixel background, E , an expected chromaticity line, $\|E\|$, a chromatic distortion, d , and a brightness threshold, τ . For a given observed pixel value, I , brightness distortion, a , and color distortion, c , from the background model can be calculated as

$$a = \arg \min_z (I - zE)^2 \tag{14}$$

$$c = \|I - aE\| \tag{15}$$

The standard deviation of the k th component, σ_k can be set equal to d if the Gaussian distribution is spherical. The calculation of a and c are trivial using vector dot product. A non-background observed sample is considered as moving shadow if it lies within $2.5\sigma_k$ and $\tau < c < 1$.

2.3 Effective GMM

Lee (2005) proposed an effective online learning algorithm that significantly improves modeling convergence and accuracy for adaptive Gaussian mixture modeling of dynamic distributions without compromising model stability. A modified learning rate ρ is used to achieve fast convergence and temporal adaptability which is computed for each Gaussian independently from the cumulative expected likelihood estimate. A counter c_k is also used, which is increased when the Gaussian parameters are updated and reset to 1 if the Gaussian is re-assigned.

In Stauffer and Grimson (1999), $\rho = \alpha \eta_k(X_t)$ where the value of $\eta_k(X_t) = P(X_t/G_k)$ is very small, which makes convergence slower. Hence, $\rho_{k,t} = \alpha P(G_k/X_t)$ is used. Weighing ρ by the expected posterior achieves the soft partitioning.

Mostly, a single best matching component is selected for parameter update in the other algorithms for efficiency reasons. However, if the distribution contains two significantly overlapping clusters, this strategy can lead to starvation where one Gaussian stretches with increasingly more weight to overdominate the others. In the algorithm proposed in Lee (2005), multiple Gaussians may be updated for a single sample. It uses soft-partition where all Gaussians that match a pixel value are updated by an amount proportional to their estimated posterior probability $P(G_k/X_t)$. It improves robustness in early learning stage for components whose variances are too large and weights are too small to be the best match.

At the pixel level, background segmentation involves a binary classification problem based on $P(B/X_t)$ where B represents the background.

Temporal distribution $P(X_t)$ is given by:

$$P(X_t) = \sum_{k=1}^K P(G_k)P(X_t/G_k) = \sum_{k=1}^K \omega_{k,t} * \eta(X_t, \mu_{k,t}, \Sigma_{k,t}) \tag{16}$$

The posterior probability can be expressed in terms of the mixture components $P(G_k)$ and $P(X_t/G_k)$, and a density estimate $P(B/G_k)$ as follows:

$$P(B/X_t) = \sum_{k=1}^K P(B/G_k)P(G_k/X_t) = \frac{\sum_{k=1}^K P(X_t/G_k)P(G_k)P(B/G_k)}{\sum_{k=1}^K P(X_t/G_k)P(G_k)} \tag{17}$$

In [Stauffer and Grimson \(1999\)](#), $P(B/G_k)$ equals to 1 for Gaussians with the highest ω/σ , and 0 for all others whereas in [Lee \(2005\)](#), a sigmoid function is used on ω/σ to approximate $P(B/G_k)$ using logistic regression which is given below:

$$P(B/G_k) = f(\omega_k/\sigma_k; a, b) = 1/(1 + e^{-a.\omega_k/\sigma_k+b}) \tag{18}$$

After $P(X_t)$ and $P(B/G_k)$ are estimated, the foreground and background pixels can be classified. The pixels for which $P(B/X_t) < 0.5$ are classified as foreground and others are classified as background.

2.4 Efficient GMM

Zivkovic and van der Heijden (2006) presented an improved adaptive GMM which uses model selection criterion to choose the right number of components for each pixel on-line and automatically fully adapt to the scene. The recursive update equations are same as the update equations proposed in [Stauffer and Grimson \(1999\)](#). The background model is estimated by first B largest clusters where B is given by

$$B = \arg \min_b \left(\sum_{k=1}^b \omega_k > (1 - c_f) \right) \tag{19}$$

where c_f is a measure of the maximum portion of the data that belongs to foreground objects without influencing the background model. The weight w_k is the fraction of the data that belongs to the k th component of the GMM. A constant $\alpha = 1/T$ defines an exponentially decaying envelope that is used to limit the influence of the old data. The adaptive update equation used in [Zivkovic and Heijden \(2006\)](#) is given below:

$$\omega_{k,t} = \omega_{k,t-1} + \alpha(M_{k,t} - \omega_{k,t-1}) - \alpha c_T \tag{20}$$

After each update weight w_k -s are normalized so that they add up to one. A GMM with one component is started which is centered on the first sample. For a new sample the ownership (matching function) $M_{k,t}$ is set to 1 for the matching component (if the Mahalanobis distance, MD from the component is less than a specified threshold) and for other components it is set to 0. The squared MD from the k_{th} component is calculated as:

$$D_k^2(x_t) = \delta_{k,t}^T \Sigma_{k,t}^{-1} \delta_{k,t} \tag{21}$$

where $\delta_{k,t} = x_t - \mu_{k,t-1}$. If there are no matched components, a new component is generated with $\omega_{k+1,t} = \alpha$, $\mu_{k+1,t} = x_t$ and $\sigma_{k+1,t} = \sigma_o$, where σ_o is an appropriately initialized variance value. If the maximum number of components is reached, the component with smallest

w_k is discarded. c_T is a negative prior evidence weight which suppress the components that are not supported by the data and ensures that components with negative weights are discarded. This also ensures that the mixing weights stay non-negative.

2.5 Localized adaptive learning GMM

In [Shah et al. \(2010\)](#), a pixel based recursively adaptive learning rate for Mixture of Gaussians is proposed. A separate learning rate is used for each pixel because different regions in a frame have different learning behavior and need a different learning rate.

Intensity of each pixel in the t^{th} frame denoted as $I(X_t) = (X_t^R, X_t^G, X_t^B)$, where (X_t^R, X_t^G, X_t^B) are R, G and B components of that pixel of the t th frame. Long-term history of intensity fluctuation of each pixel X_t is modeled as exponential weighted moving average and short term change in pixel intensity is modeled as an absolute difference in intensity of pixel in consecutive frames as shown below:

$$\delta = E \left(\left| X_t^R - X_{t-1}^R \right|, \left| X_t^G - X_{t-1}^G \right|, \left| X_t^B - X_{t-1}^B \right| \right) \tag{22}$$

Long-term intensities fluctuation is calculated using a practical and simple recursive formula:

$$\Delta_t = (1 - \gamma)\Delta_{t-1} - \gamma\delta \tag{23}$$

where $\gamma = 0.4$ is normally sufficient, which is selected empirically.

The proposed recursive procedure for automatic tuning of α only needs to keep last frame and removes the dependence on the initial value. In comparison to MoG model, the only extra storage required by this technique is the previous frame and α , a separate learning rate for each pixel, which is insignificant compared to performance gain. A learning rate α has strong correlation with fluctuation history of the pixels. Thus, α for each pixel is modeled as shown below:

$$\alpha_t = \begin{cases} \alpha_{t-1} + \frac{|\delta - \Delta_t|}{\delta + \Delta_t}; & \text{if } \delta > \Delta_t \text{ and } \alpha_{t-1} \geq 1 \\ \alpha_{t-1} - \frac{|\delta - \Delta_t|}{\delta + \Delta_t}; & \text{if } \delta < \Delta_t \text{ and } \alpha_{t-1} \geq 0 \\ \alpha_{t-1}; & \text{otherwise} \end{cases} \tag{24}$$

This algorithm is referred as 'MoG-LAL1', i.e. Mixture of Gaussians using Local Adaptive Learning, with the local learning rate tuned by intensity changes. In MOG-LAL1, adaptation step size is modeled using amount of change of pixels intensity, but in reality it depends on fluctuation frequency. Hence, authors in [Shah et al. \(2010\)](#) proposed another algorithm 'MOG-LAL2' i.e., Mixture of Gaussians using Local Adaptive Learning with learning rate tuned by fluctuation frequency. In MOG-LAL2, step size is modeled using fluctuation frequency as given below:

$$f = \begin{cases} 1; & \text{if } \delta > \phi \\ 0; & \text{otherwise} \end{cases} \tag{25}$$

$$F_t = (1 - \gamma)F_{t-1} + \gamma f \tag{26}$$

$$\alpha_t = \begin{cases} 0.9\alpha_{t-1} + 0.1 \frac{F_t}{N}; & \text{if } 0 \leq \alpha_{t-1} \leq 1 \\ \alpha_{t-1}; & \text{otherwise} \end{cases} \tag{27}$$

where f is short-term, F is long-term pixels intensity fluctuation frequency and ϕ is a threshold used to avoid change due to noise and typically set to some small value.

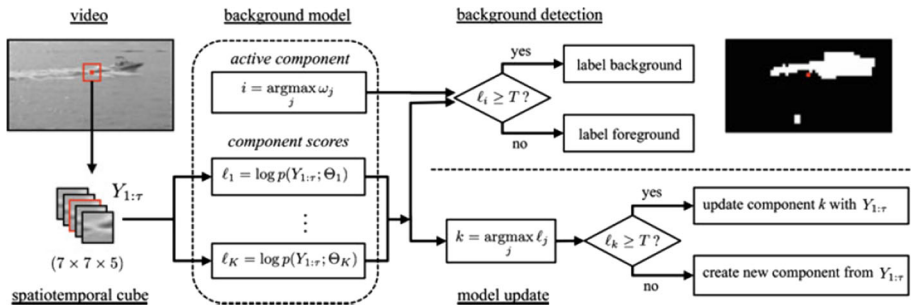


Fig. 1 Overview of generalized Stauffer–Grimson background modeling method (Chan et al. 2011)

2.6 Generalized Stauffer–Grimson GMM

Chan et al. (2011) proposed a Generalized Stauffer–Grimson (GSG) algorithm for dynamic backgrounds. A background modeling method based on dynamic textures is proposed in this paper which extends the background subtraction algorithm proposed by Stauffer and Grimson. The sufficient statistics required for online learning of the dynamic texture is derived and is used to generalize the GMM proposed in Stauffer and Grimson (1999) to dynamic scenes. This algorithm adapts to long-term variations via online estimation, it can quickly embrace new background motions through the addition of mixture components and easily discards outdated information by dropping mixture components with small priors (Chan et al. 2011). An overview of the proposed algorithm is shown in Fig. 1.

The background scene is modeled as a mixture of K dynamic textures, from which spatiotemporal volumes $Y_{1:\tau}$ are drawn. Chan et al. (2011) assumed the value of τ equal to 5. The j th dynamic texture is denoted by Θ_j , and a prior weight ω_j is associated with each dynamic texture such that $\sum_{j=1}^K \omega_j = 1$. Each video location is represented by a neighboring spatiotemporal volume. Firstly, the component of largest prior weight, $i = \arg \max_j (\omega_j)$, is selected as the active background component. The video location is marked as background if the log-likelihood of the corresponding spatiotemporal volume $Y_{1:\tau}$ under the active background component is above a specified threshold T .

$$\log p(Y_{1:\tau} | \Theta_i) \geq T \tag{28}$$

Next, the mixture component with the largest log-likelihood of generating $Y_{1:\tau}$ is updated using an approximation to online Expectation- Maximization algorithm, if the log-likelihood is above threshold. If the largest log-likelihood is below threshold then a new component is learned which replaces the component of lowest prior probability. The background mixture model is learned with an online K-means algorithm.

2.7 SURF based GMM

The model proposed in Shah et al. (2014) automatically learns dynamics of a scene and adapts its parameters accordingly. The framework of the model is shown in Fig. 2. It uses YUV color vector instead of RGB. A new matching function is proposed which checks the intensity (Y) and color (UV) values of the current pixel against the existing Gaussian components:

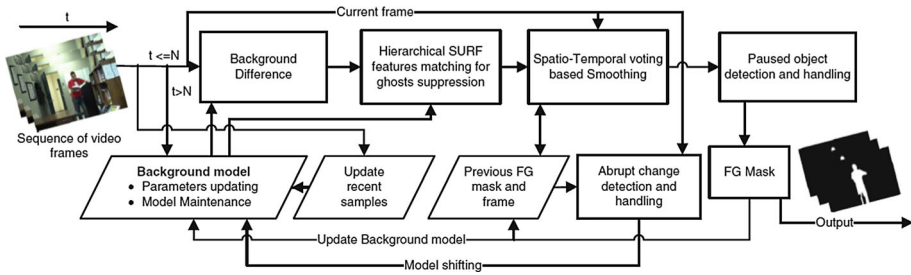


Fig. 2 Framework of the model proposed in Shah et al. (2014)

$$M_{k,t} = \begin{cases} 1; & \text{if } |\mu_{k,t-1}^Y - X_t^Y| < \lambda_{k,t} \sigma_{k,t-1}^Y \text{ and} \\ & |\mu_{k,t-1}^c - X_t^c| < \lambda_{k,t} \sigma_{k,t-1}^c \\ 0; & \text{otherwise} \end{cases} \tag{29}$$

where X_t^Y is the pixel intensity, X_t^c stands for a color component value (i.e., c is either U or V), and $\lambda_{k,t}$ is the local coefficient (typically set to 2.5) to the pixel value deviations ($\sigma_{k,t}^Y$ and $\sigma_{k,t}^c$) for thresholding.

Some background components occur less frequently, hence they have lower weights and cannot qualify for background component because of low ranking instead of being in the model for long time. Therefore, a counter ν is used which starts from zero when a component is created or replaced and is increased by 1 in every update. To differentiate between slow moving objects and paused objects, a winning frequency Γ is calculated for each component. Hence, if the matched component is not among the first B components, $\nu > TT$ and $\frac{1}{2} \geq \Gamma \geq \frac{1}{K}$ (K is the number of Gaussian Components), then the component is labeled as background. Here, TT is a threshold which is empirically as $TT = 100$. If the pixel does not match with any of the Gaussians then the component is replaced by new parameters given below:

$$\mu_{k,t}^c = (1 - \rho_{k,t})\mu_{k,t-1}^c + \rho_{k,t}X_t^c \tag{30}$$

$$\mu_{k,t}^Y = (1 - \rho_{k,t})\mu_{k,t-1}^Y + \rho_{k,t}X_t^Y \tag{31}$$

$$(\sigma_{k,t}^c)^2 = (1 - \rho_{k,t})(\sigma_{k,t-1}^c)^2 + \rho_{k,t}(\mu_{k,t-1}^c - X_t^c)^2 \tag{32}$$

$$(\sigma_{k,t}^Y)^2 = (1 - \rho_{k,t})(\sigma_{k,t-1}^Y)^2 + \rho_{k,t}(\mu_{k,t-1}^Y - X_t^Y)^2 \tag{33}$$

$$\rho_{k,t} = \frac{\alpha_t M_{k,t}}{\omega_{k,t}} \tag{34}$$

In this model, a local learning rate is used because it is very difficult to model a complex dynamic scene with a single global learning rate. A small learning rate is required for static background, whereas a higher learning rate is required for dynamic background scene. The fluctuation frequency is given by:

$$f_t = \begin{cases} f_{t-1} + 1; & \text{if } |X_t - X_{t-1}| > \lambda_{k,t} \\ f_{t-1}; & \text{otherwise} \end{cases} \tag{35}$$

where $\lambda_{k,t}$ is a threshold used to ignore minor changes due to noise.

$$\alpha_t = \frac{\sum_{i=1}^N f_{i,t}}{KN} \tag{36}$$

where N is the set of recent frames such that $10 \leq N \leq 20$.

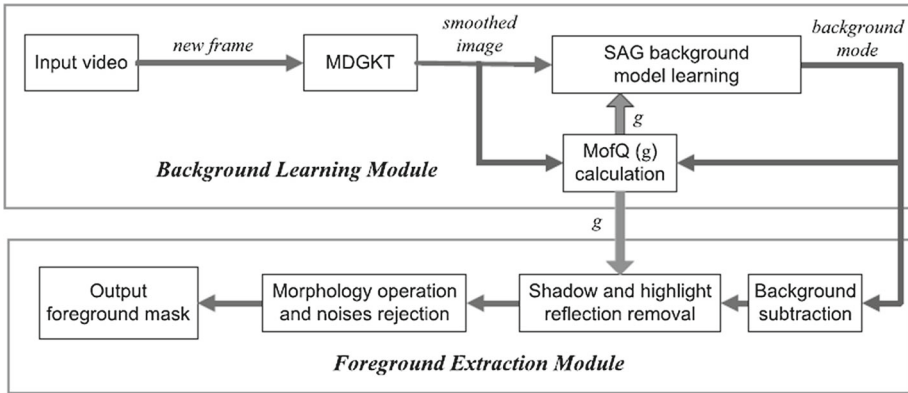


Fig. 3 Flowchart of SAG combined with MDGKT (Chen and Ellis 2014)

To handle rapid changes such as shadows and lighting switching, causing a large number of false positives ('ghosts'), a SURF features matching algorithm is used to suppress ghosts. The identified foreground regions are further matched with the background image using feature descriptors. Matched regions are then regarded as 'ghosts', re-classified as background, and removed from the foreground map. The SURF descriptors are illumination invariant and fast enough to serve the purpose of real-time processing, hence they are used for ghosts detection and suppression.

A new spatio-temporal filter is introduced to further refine the foreground detection results. To handle global illumination changes, global illumination detection and background adjustment techniques are introduced and paused objects are detected and handled using spatio-temporal history of foreground blobs.

2.8 Self-adaptive GMM

Chen and Ellis (2014) proposed a self-adaptive GMM (SAG). Flowchart of this model is shown in Fig. 3. This model uses dynamic learning-rate adaptation to cope with fast illumination changes. This algorithm is less sensitive to sudden changes in the global illumination.

In background learning process, image noise is suppressed using a spatio-temporal filter (MDGKT) which improves the stability and robustness of the algorithm and a global illumination changing factor (g) is computed by the MofQ applied to the smoothed image. The MofQ global illumination change factor g is defined as

$$g = \text{median}_{s \in S} \left(\frac{i_{c,s}}{i_{r,s}} \right) \tag{37}$$

for all pixels s in set S (the image), between the current image i_c and a reference image i_r . Then, the background is learned using self-adaptive GMM (SAG). SAG introduces the factor g between the learnt background and the current input image which keeps track of global illumination changes and a counter c_k which is increased when the Gaussian parameters are updated and if the Gaussian is re-assigned, it is reset to 1. When the parameters are updated, a new learning rate β_k is calculated as given below:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t} + C_T) \tag{38}$$

$$\beta_{k,t} = \alpha(l + c_k)/c_k \tag{39}$$

$$\mu_{k,t} = \mu_{k,t-1} + M_{k,t}(\beta_{k,t}/\omega_{k,t})\delta_{k,t} \tag{40}$$

$$\sigma_{k,t}^2 = \sigma_{k,t-1}^2 + M_{k,t}(\beta_{k,t}/\omega_{k,t}) \left(\delta_{k,t}^T \delta_{k,t} - \sigma_{k,t-1}^2 \right) \tag{41}$$

$$c_k = c_k + 1 \tag{42}$$

where l is a constant, $\alpha = 1/T$ is a constant that defines an exponentially decaying envelope and is used to limit the influence of the old data. C_T is a negative prior weight which suppress the components that are not supported by the data. $M_{k,t}$ is set to 1 for the matched component (if the MD from the component is less than a specified threshold) and for other components, it is set to 0. The squared MD is calculated as given below:

$$D_k^2(x_t) = \delta_{k,t}^T \Sigma_{k,t}^{-1} \delta_{k,t} \tag{43}$$

where $\delta_{k,t} = g \cdot x_t - \mu_{k,t-1}$ If there are no matched components, a new component is generated with $\omega_{k+1,t} = \alpha$, $\mu_{k+1,t} = x_t$, $\sigma_{k+1,t} = \sigma_o$ and $c_{k+1} = 1$, where σ_o is an appropriately initialized variance value. If the maximum number of components is reached, the component with smallest w_k is discarded.

This algorithm also detects shadows and highlights using spectral, spatial and temporal features either individually or in combination. The distorting effect of shadow and highlight in RGB space is decomposed into two components, brightness and chromaticity distortion. If the intensity value of the i th pixel is $I_i = [I_{Ri}, I_{Gi}, I_{Bi}]^T$ in RGB space, the estimated mean is $E_i = [\mu_{Ri}, \mu_{Gi}, \mu_{Bi}]^T$ and the pixel standard deviation is $\sigma_i = [\sigma_{Ri}, \sigma_{Gi}, \sigma_{Bi}]$, then the distortion of the brightness B_i and chromaticity CD_i are computed as:

$$B_i = \frac{g(I_{Ri}\mu_{Ri}/\sigma_{Ri}^2 + I_{Gi}\mu_{Gi}/\sigma_{Gi}^2 + I_{Bi}\mu_{Bi}/\sigma_{Bi}^2)}{(\mu_{Ri}/\sigma_{Ri})^2 + (\mu_{Gi}/\sigma_{Gi})^2 + (\mu_{Bi}/\sigma_{Bi})^2} \tag{44}$$

$$CD_i = \sqrt{((gI_{Ri} - B_i\mu_{Ri})/\sigma_{Ri})^2 + ((gI_{Gi} - B_i\mu_{Gi})/\sigma_{Gi})^2 + ((gI_{Bi} - B_i\mu_{Bi})/\sigma_{Bi})^2} \tag{45}$$

A foreground pixel is then classified according to the following condition:

$$\begin{cases} shadow; & \text{if } CD_i < \gamma_1 \text{ and } \gamma_2 < B_i < 1 \\ highlight; & \text{if } CD_i < \gamma_1 \text{ and } B_i > \gamma_3 \end{cases} \tag{46}$$

where γ_1 , γ_2 and γ_3 are the threshold values. $\gamma_1(0 < \gamma_1 \leq 1)$ distinguishes between chromaticity values of the GMM-learned background and the current image frame, $\gamma_2(0 < \gamma_2 < 1)$ differentiates between dark pixel values and shadows and $\gamma_3(\gamma_3 > 1)$ is used to detect highlights.

3 Evaluation metrics and analysis

3.1 Evaluation metrics and quantitative analysis

Background subtraction algorithms using GMM have limited performance for various challenges due to the assumption of parameters. The performance of algorithms vary with challenges. Generally used standard performance metrics for quantitative analysis are Recall (REC), Precision (PRE), F-measure (FM), Specificity (Sp), False Positive Rate (FPR), False Negative Rate (FNR) and Percentage of Wrong Classifications (PWC) (Goyette et al. 2012) which evaluate the performance of algorithm and its robustness against various challenges. The Recall, Precision and F-measure are based on the accuracy in detecting the pixels as

foreground and background and is measured by the number of foreground pixels classified as foreground (True Positives, TP), number of background pixels classified as foreground (False Positives, FP), number of background pixels classified as background (True Negatives, TN) and number of foreground pixels classified as background (False Negatives, FN).

Recall is also known as sensitivity and is used to measure the correctly identified foreground pixels with reference to the total number of pixels classified as foreground.

$$REC = \frac{\text{correctly classified foreground pixels}}{\text{total number of pixels classified as foreground}} = \frac{TP}{(TP + FN)} \quad (47)$$

Precision is the positive predictive value and it is the fraction of correctly classified foreground pixels to the actual number of foreground pixels.

$$PRE = \frac{\text{correctly classified foreground pixels}}{\text{foreground pixels in ground truth}} = \frac{TP}{(TP + FP)} \quad (48)$$

Both precision and recall are based on the measure of relevance. In terms of segmentation, a low recall indicates that the algorithm has over-segmented the foreground object, while a low precision indicates an under-segmented foreground object.

The F-measure considers both the Recall and the Precision. It is a weighted average of the Recall and the Precision. Higher FM means a good background subtraction algorithm.

$$FM = \frac{(2 * Pre * Rec)}{(Pre + Rec)} \quad (49)$$

The other performance metrics used are calculated with the help of TP, FP, TN, FN are Specificity, FPR, FNR, PWC. They are expressed as:

$$Sp = \frac{TN}{(TN + FP)} \quad (50)$$

$$FPR = \frac{FP}{(FP + TN)} \quad (51)$$

$$FNR = \frac{FN}{(TP + FN)} \quad (52)$$

$$PWC = \frac{100 * (FN + FP)}{(TP + FN + FP + TN)} \quad (53)$$

Based on these quantitative analysis parameters, comparison of some of the algorithms given in Sect. 2 on CDnet Dataset (Goyette et al. 2012) is presented in Table 2. Table 2 shows that algorithm proposed by KaewTraKulPong and Bowden (2002) and Zivkovic and Heijden (2006) has low recall value and high FNR compared to other algorithms i.e., it oversegments the foreground object and is not able to classify the different foreground objects according to their shape. Algorithm proposed by Lee (2005) has a very low value of precision and a high FPR which results in undersegmented foreground object. Overall, the algorithms proposed in KaewTraKulPong and Bowden (2002), Lee (2005), Stauffer and Grimson (1999) and Zivkovic and Heijden (2006) have a low FM score which shows the poor background and foreground detection in comparison to algorithm proposed by Shah et al. (2014) and Chen and Ellis (2014).

Visual results have also been presented in Fig. 4 which shows the segmentation results of the algorithms GMM (Stauffer and Grimson 1999), Efficient GMM (Zivkovic and Heijden 2006) and Self-adaptive GMM (Chen and Ellis 2014). These segmentation results has been obtained on PETS 2006 and Fall video sequences from CDnet dataset (Goyette et al. 2012) and SEQ111 and SEQ121 from BMC dataset (Vacavant et al. 2013). SEQ111 and SEQ121 are

Table 2 Comparative results for some algorithms given in Sect. 2 on CDnet dataset (Goyette et al. 2012)

Method	REC	PRE	FM	PWC	FPR	FNR	Sp
Stauffer and Grimson (1999)	0.7108	0.7012	0.6624	3.1037	0.0140	0.2892	0.9860
KaewTraKulPong and Bowden (2002)	0.5072	0.8228	0.5904	3.1051	0.0053	0.4928	0.9947
Lee (2005)	0.7794	0.6237	0.6475	3.7014	0.0244	0.2206	0.9756
Zivkovic and Heijden (2006)	0.6707	0.7882	0.6785	2.7954	0.0093	0.3293	0.9907
Shah et al. (2014)	0.8306	0.8576	0.8548	1.0609	0.0051	0.1916	0.9918
Chen and Ellis (2014)	0.9385	0.8026	0.8652	0.0020	0.0016	0.0615	0.9984

Bold values indicate the best performing models for that performance metric

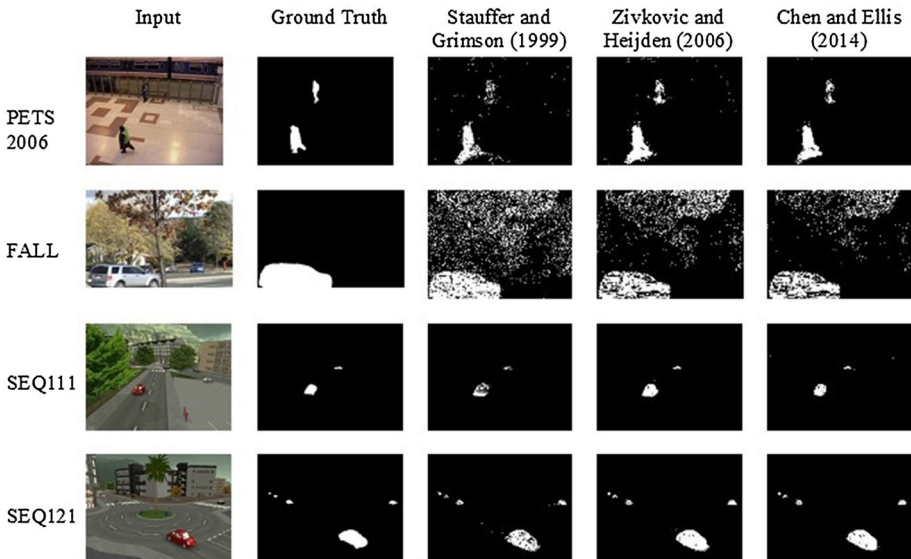


Fig. 4 Segmentation results

synthetic video sequences while PETS 2006 and Fall are real video sequences. For all video sequences, algorithm proposed by Chen in Chen and Ellis (2014) gives better segmentation results due to the use of varying learning rate. But for Fall video sequence, algorithm proposed in Chen and Ellis (2014) does not perform better because of the strong dynamic background.

3.2 Performance analysis

Background Subtraction for foreground detection in video surveillance systems faces challenges like dynamic background, illumination changes, processing speed, initial learning etc. Algorithms discussed in Sect. 2 are capable to overcome some of the challenges.

GMM proposed by Stauffer and Grimson (1999) can cope well with the illumination changes as well as it can handle the problem of objects being introduced or removed from the scene. This works well for a static background subject to gradual illumination changes

in the scene but it fails to handle dynamic changes in the background such as leaves swaying, or water waves. Background having fast variations cannot be accurately modeled with just a few Gaussians (usually 3–5), which causes problems for sensitive detection. It also suffers from slow learning at the beginning, especially in busy environments and cannot distinguish between moving shadows and moving objects (KaewTraKulPong and Bowden 2002).

GMM is improved by KaewTraKulPong and Bowden (2002) to accommodate the slow learning issue of GMM and also introduced a shadow detection technique using brightness and chromatic distortion cue's. KaewTraKulPong and Bowden in KaewTraKulPong and Bowden (2002) proposed new update equations using expected sufficient statistics at the initial learning stage to improve convergence speed and switching to recursive filter learning after sufficient samples were observed. But this approach is not helpful for learning new foreground objects at a later stage, where effective learning is most needed due to the small number of samples (Lee 2005).

In Lee (2005), Lee proposed an algorithm with improved speed and adaptation rate of the model. An adaptive learning rate is introduced for each Gaussian along with global learning rate. For efficiency reasons, it also uses a winner-take all option where parameters of the single best matching Gaussian component are selected for update and increases the learning rate of the winner Gaussian by some small fraction every time it wins. Although this technique achieves some improvements in segmentation but it is time and memory expensive due to introduction of additional learning rate for each matching Gaussian components, especially when number of Gaussian components are high. When the number of observations increases, it starts to behave like the standard GMM (Stauffer and Grimson 1999) model. This is because this additional fraction is normalized by the number of frames encountered until time t . It also fails in scenarios where there is a dynamic background and modes of the pixels are fluctuating rapidly (Shah et al. 2010).

Zivkovic and Heizden proposed an algorithm in Zivkovic and Heijden (2006) which adaptively select the number of Gaussians used to model each pixel and employs a recursive computation to update the model parameters and hence it is suitable for online (real-time) operation. But this technique uses a single learning rate which results in a trade-off in detection accuracy. For a high learning rate, the model updates too quickly, and slow-moving objects are absorbed into the background model, which results in a high false negative detection rate. A low learning rate will fail to adapt to sudden illumination changes which results in a high false positive rate (Chen and Ellis 2014).

In Shah et al. (2010), Shah et al. proposed a pixel based recursively adaptive learning rate for Mixture of Gaussians. A strong correlation between learning rate and the local temporal intensity variations has been explored in order to adapt this over time and a separate learning rate is used for each pixel. This is both time and memory efficient and gave significant improvements in segmentation results. But this algorithm uses fixed number of Gaussian components K . The number of Gaussian components depends upon possible modes in pixel values. As in real-world application, a prior value of modes of pixel cannot be determined, hence it is very difficult to set a value of K . Higher number of Gaussian components will make the algorithm computationally expensive because most of these components in most of the cases will be of no use and lower number of Gaussian components cannot accurately model the background and foreground. Consequently, there is a need for a dynamic procedure for the optimal selection of Gaussian components.

The GSG method (Chan et al. 2011) proposed by Chan et al. supports models with arbitrary component densities but these densities can only be summarized by sufficient statistics. This model is applied where the component densities are dynamic textures and produces an

adaptive background subtraction algorithm based on the mixture of dynamic textures, which is suitable for dynamic scenes and online video processing. This algorithm achieved high detection and low false positive rates. Since this algorithm used texture features, hence it is good to handle illumination variation. But texture features are sensitive to noise, hence it cannot be applied in low resolution video (Shah et al. 2014).

Another algorithm (Shah et al. 2014) proposed by Shah et al. automatically learns dynamics of a scene and adapts its parameters accordingly. This algorithm uses YUV color space along with SURF features to model the background and refine the foreground mask. SURF features are quite resilient to noise and better to deal with illumination changes. A SURF features matching algorithm is employed to suppress the ghosts and also introduces a new spatio-temporal filter to further refine the foreground detection results. Although this model achieved fairly good results for shadows detection while remaining computing inexpensive, but it cannot detect the strong shadows (Shah et al. 2014). The SURF based approach classifies the low texture regions as background and high texture regions as foreground. Hence, if very few features are detected for a region, it will be classified as background resulting in false positive detections. This algorithm also uses fixed number of Gaussian components.

The algorithm (Chen and Ellis 2014) proposed by Chen and Ellis has a dynamically adaptive learning rate, adaptively selects the number of Gaussians to model each pixel and is less sensitive to sudden changes in the global illumination. It also employs a spatio-temporal Gaussian smoothing algorithm and shadow removal algorithm. But, this proposed algorithm is not able to remove the ghosts. This algorithm uses RGB color features which are good for high-quality video but if foreground objects and the background have similar color, then color features will not give good results. It also assumes that red, green and blue pixel values are independent and have the same variance but these color components are not independent and so using a simplification of the covariance by a 3×3 identity matrix is not accurate and results in more false positive and false negative detections (Bouwman et al. 2008).

Table 3 presents the comparison of these algorithms in brief. This table gives the modification done in the standard GMM algorithm by the different authors. It also gives the advantages and limitations of various algorithms.

4 Conclusion and future scope

This paper presents a performance analysis of different background subtraction algorithms based on GMM. Quantitative analysis is also presented using standard performance metrics and also identifies the advantages and limitations of these algorithms to determine most suitable algorithm for a particular scenario in video surveillance systems. Algorithms proposed by Chen and Ellis (2014) and Shah et al. (2014) are most promising for the challenges faced by video surveillance systems like sudden illumination changes, dynamic background etc. but they have limited performance in videos with moving shadows. Hence their performance degrades for videos taken during bright sunlight. Most of the algorithms assumed that the dimensions of a pixel value are statistically independent and simplified the covariance matrix to an identity matrix which results in more false positive and false negative detections. Variances of red, green and blue pixel values are also assumed equal for RGB color space. Considering the pixel value dimensions dependent and computing the variance for each component (color or intensity component) separately will result in more accurate foreground and background detection. Generally, the parameters weight, mean and variance used in GMM are initialized based on assumptions suitable for the particular scenario. The

Table 3 Comparison of different algorithms using GMM

Algorithm	Modifications	Advantages	Limitations
Stauffer and Grimson (1999)	Standard Gaussian Mixture Model	Robust to multi-modal and quasi periodic background variations and gradual illumination changes	Unsatisfactory performance for sudden illumination changes and for irregular background motions
KaewTraiKulPong and Bowden (2002)	Modified Update equations	Improves Convergence rate	Not suitable for effective learning due to small number of samples
Lee (2005)	Modified learning rate	Improved speed and adaptation rate without compromising stability	Time and memory expensive
Zivkovic and Heijden (2006)	Adaptively selects the number of Gaussians	Fast compared to standard GMM	High false positive and false negative rate due to the use of a single learning rate
Shah et al. (2014)	Uses varying learning rate	Time and memory efficient	Computationally expensive due to the use of fixed number of Gaussian components
Chan et al. (2011)	Uses dynamic textures instead of RGB color intensity	Performs better for sudden illumination variations	Sensitive to noise and cannot be applied in low resolution video
Shah et al. (2014)	Adaptive learning rate and uses YUV color space instead of RGB	Better performance for sudden illumination changes, paused objects and moving shadows	Cannot detect strong shadows
Chen and Ellis (2014)	Uses varying learning rate and also adaptively selects the number of Gaussians	Performs better for dynamic background and sudden illumination variations	Cannot handle strong dynamic background and paused objects

value of these parameters can be more accurately determined for different scenarios using artificial intelligence. The deviation threshold is fixed which is set equal to 2.5 but this fixed value is not good enough for every case and depends upon the camera noise and variance in pixel values. Hence, a mechanism should be there to calculate this threshold automatically for the particular condition.

References

- Bouwman T (2009) Subspace learning for background modeling: a survey. *Recent Patents Comput Sci* 2(3):223–234
- Bouwman T (2011) Recent advanced statistical background modeling for foreground detection—a systematic survey. *Recent Patents Comput Sci* 4(3):147–176
- Bouwman T (2014) Traditional and recent approaches in background modeling for foreground detection: an overview. *Comput Sci Rev* 11–12:31–66
- Bouwman T, Baf FE (2010) Statistical background modeling for foreground detection: a survey. *Handbook of Pattern Recognition and Computer* pp 181–199
- Bouwman T, Baf FE, Vachon B (2008) Background modeling using mixture of Gaussians for foreground detection a survey. In: *Recent patents on computer science*, pp 219–237
- Butler D, Sridharan S, Bove VMJ (2003) Real-time adaptive background segmentation. In: *Multimedia and expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol 3, pp 341–344
- Chan A, Mahadevan V, Vasconcelos N (2011) Generalized Stauffer–Grimson background subtraction for dynamic scenes. *Mach Vis Appl* 22(5):751–766. doi:[10.1007/s00138-010-0262-3](https://doi.org/10.1007/s00138-010-0262-3)
- Chang R, Gandhi T, Trivedi M (2004) Vision modules for a multi-sensory bridge monitoring approach. In: *Intelligent transportation systems, 2004. Proceedings. The 7th international IEEE conference on*, pp 971–976
- Chen Z, Ellis T (2014) A self-adaptive Gaussian mixture model. *Comput Vis Image Underst* 122:35–46. doi:[10.1016/j.cviu.2014.01.004](https://doi.org/10.1016/j.cviu.2014.01.004)
- Cheung SCS, Kamath C (2004) Robust techniques for background subtraction in urban traffic video. In: Panchanathan V, Vasudev B (eds) *Proceedings Visual Communications and Image Processing, SPIE*, vol 5308, pp 881–892
- Elgammal A, Harwood D, Davis L (2000) Non-parametric model for background subtraction. In: Vernon D (ed) *Computer Vision-ECCV 2000*, vol 1843., *Lecture Notes in Computer Science*. Springer, Berlin, pp 751–767
- Figueiredo MAT, Jain AK (2002) Unsupervised learning of finite mixture models. *IEEE Trans Pattern Anal Mach Intell* 24(3):381. doi:[10.1109/34.990138](https://doi.org/10.1109/34.990138)
- Goyette N, Jodoin P, Porikli F, Konrad J, Ishwar P (2012) Changedetection.net: a new change detection benchmark dataset. In: *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pp 1–8
- Ilyas A, Scuturici M, Miguet S (2009) Real time foreground-background segmentation using a modified codebook model. In: *Advanced video and signal based surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*, pp 454–459
- KaewTraKulPong P, Bowden R (2002) An improved adaptive background mixture model for real-time tracking with shadow detection. In: Remagnino P, Jones G, Paragios N, Regazzoni C (eds) *Video-Based Surveill Syst*. Springer, Berlin, pp 135–144
- Lee DS (2005) Effective Gaussian mixture learning for video background subtraction. *IEEE Trans Pattern Anal Mach Intell* 27(5):827–832. doi:[10.1109/TPAMI.2005.102](https://doi.org/10.1109/TPAMI.2005.102)
- McFarlane N, Schofield C (1995) Segmentation and tracking of piglets in images. *Mach Vis Appl* 8(3):187–193. doi:[10.1007/BF01215814](https://doi.org/10.1007/BF01215814)
- Messelodi S, Modena C, Segata N, Zanin M (2005) A Kalman filter based background updating algorithm robust to sharp illumination changes. In: Roli F, Vitulano S (eds) *Image analysis and processing ICIAP 2005*, vol 3617., *Lecture Notes in Computer Science* Springer, Berlin, pp 163–170
- Piccardi M (2004) Background subtraction techniques: a review. In: *Systems, man and cybernetics, 2004 IEEE international conference on*, vol. 4, pp 3099–3104
- Power PW, Schoonees JA (2002) Understanding background mixture models for foreground segmentation. In: *Proceedings image and vision computing New Zealand*
- Roy A, Shinde S, don Kang K (2010) An approach for efficient real time moving object detection. In: *Embedded systems & applications (ESA), International Conference on (2010)*

- Shah M, Deng J, Woodford B (2010) Localized adaptive learning of mixture of Gaussians models for background extraction. In: *Image and Vision Computing New Zealand (IVCNZ)*, 25th international conference of, pp 1–8. doi:[10.1109/IVCNZ.2010.6148870](https://doi.org/10.1109/IVCNZ.2010.6148870)
- Shah M, Deng J, Woodford B (2014) Video background modeling: recent approaches, issues and our proposed techniques. *Mach Vis Appl* 25(5):1105–1119. doi:[10.1007/s00138-013-0552-7](https://doi.org/10.1007/s00138-013-0552-7)
- Sobral A, Vacavant A (2014) A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Comput Vis Image Underst* 122:4–21. doi:[10.1016/j.cviu.2013.12.005](https://doi.org/10.1016/j.cviu.2013.12.005)
- Stauffer C, Grimson W (1999) Adaptive background mixture models for real-time tracking. In: *Computer vision and pattern recognition, 1999. IEEE Computer Society Conference on.*, vol 2, pp 246–252. Los Alamitos. doi:[10.1109/CVPR.1999.784637](https://doi.org/10.1109/CVPR.1999.784637)
- Toyama K, Krumm J, Brumitt B, Meyers B (1999) Wallflower: principles and practice of background maintenance. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol 1, pp 255–261
- Vacavant A, Chateau T, Wilhelm A, Lequivre L (2013) A benchmark dataset for outdoor foreground/background extraction. In: Park JI, Kim J (eds) *Computer Vision—ACCV 2012 Workshops*, vol 7728., *Lecture Notes in Computer Science*. Springer, Berlin, pp 291–300
- Wang L, Pan C (2011) Effective multi-resolution background subtraction. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp 909–912
- White B, Shah M (2007) Automatically tuning background subtraction parameters using particle swarm optimization. In: *2007 IEEE International Conference on Multimedia and Expo*, pp 1826–1829
- Yu X, Chen X, Zhang H (2011) Accurate motion detection in dynamic scenes based on ego-motion estimation and optical flow segmentation combined method. In: *Photonics and Optoelectronics (SOPO), 2011 symposium on*, pp 1–4
- Zhang E, Chen F, Zhang W (2006) A novel particle filter based background subtraction method. In: *Computational intelligence and security, 2006 international conference on*, vol 2, pp 1837–1840
- Zheng J, Wang Y, Nihan N, Hallenbeck E (2006) Extracting roadway background image: a mode based approach. *J Transp Res Rep* 1944:82–88
- Zivkovic Z, van der Heijden F (2006) Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognit Lett* 27(7):773–780. doi:[10.1016/j.patrec.2005.11.005](https://doi.org/10.1016/j.patrec.2005.11.005)