

Fast and Accurate Foreground Background Separation for Video Surveillance

Prashant Domadiya¹(✉), Pratik Shah², and Suman K. Mitra¹

¹ Dhirubhai Ambani Institute of Information and Communication Technology,
Gandhinagar 382007, Gujarat, India

pmdomadiya@gmail.com, suman_mitra@daiict.ac.in

² Indian Institute of Information Technology, Vadodara, Gujarat, India
pratik@iiitvadodara.ac.in

Abstract. Fast and accurate algorithms for background-foreground separation are essential part of any video surveillance system. GMM (Gaussian Mixture Models) based object segmentation methods give accurate results for background-foreground separation problems, but are computationally expensive. In contrast, modeling with only a single Gaussian improves the time complexity with a reduction in the accuracy due to variations in illumination and dynamic nature of the background. It is observed that these variations affect only a few pixels in an image. Most of the background pixels are unimodal. We propose a method to account for dynamic nature of the background and low lighting conditions. It is an adaptive approach where each pixel is modeled as either unimodal Gaussian or multimodal Gaussians. The flexibility in terms of number of Gaussians used to model each pixel, along with *learning when it is required* approach reduces the time complexity of the algorithm significantly. To resolve problems related to false negative due to homogeneity of color and texture in foreground and background, a spatial smoothing is carried out by K-means, which improves the overall accuracy of proposed algorithm.

Keywords: Foreground background separation · Adaptive Gaussian mixture model · Video surveillance

1 Introduction

For any video surveillance system foreground background separation is a crucial step. Foreground (object) detection is a first stage of any computer vision based system including, but not limited to, activity detection, activity recognition, behavioral understanding, surveillance, medicare, parenting etc. Accuracy of such systems depend heavily on accuracy of object detection. Moreover, error in detection can affect system performance adversely. Object background segmentation from video is a challenging task, main challenges being variations in illumination, low lighting conditions, shadow effect, moving background and occluded foreground.

A naive frame differencing approach for object segmentation from video suffers from difficulty of setting up an appropriate threshold. Many times, even if a proper threshold is set, method fails due to the dynamic nature of background. In this method, a background frame is stored and subtracted from upcoming frame to detect movements. Sometimes, mean of intensity values over a series of N frames is also used to generate the difference image. This method is fast and handles noise, shadow and trailing effect but again it's accuracy depends on threshold value. It is observed that threshold depends on the movement of foreground, fast movement require high threshold and slow movement require low threshold.

Method proposed by Wren et al. in [1] models every pixel with a Gaussian density function. The method initializes mean by the intensity of first frame pixel and variance by some arbitrary value. Subsequently, the mean and variances are learnt from series of N frames of video for each pixel. This method can not cope with dynamic background. Method uses 2.5 times standard deviation as the threshold. This method can not cope with dynamic property of background. Intensity values of background pixels change with time due to the change in illumination. This results in miss-classification of background.

Another method, based on GMM, models each pixel independently as a mixture of at most K Gaussian components. Each component of GMM is either background or different environmental factors (may be movement of leaves of trees, moving objects, shadow etc.). This adaptive method, proposed by Stauffer and Grimson in [2], can handle dynamic nature of the background. During learning if there is no significant variation in intensity of some background pixels, then the resultant background components of those pixels shrink. As a result, when we try to classify such pixels based on identified Gaussians, a sudden change in illumination can cause miss-classification.

2 Related Works

A statistical classification approach was proposed by Lee [3] that improves the convergence rate of learning parameters without affecting the stability in computation. This method improves learning of parameters of each model via replacing retention parameter by learning rate. In [4], Shimanda et al., proposed a method which reduces number of components by merging similar components to reduces computational time. It handles intensity variation by increasing number of components (Gaussians). A hybrid method proposed by M. Haque et al. in [5], is a combination of probabilistic subtraction method and basic subtraction method. This method reduces miss-classifications and number of components to be maintained at each pixel by combining thresholds used in basic background subtraction (BBS) and Wren et al. [1]. This method can handle variation in illumination without increasing number of components.

To learn the parameters of Gaussian components, sampling-resampling based Bayesian learning approach was proposed, by Singh et al. in [6]. This method gives better result than simple learning methods as suggested by Stauffer and Grimson in [2], and the method does not depend upon the learning rate. But learning is slow.

All previous methods modelled every pixel by a fixed number (K) of Gaussian components. But we believe that it is not necessary to model every pixel with a fixed number of components. Moreover, we can not predict the number of components for each pixel. Some pixel can be modeled by less than K components while other may require more than K components. We propose a method which is based on a fact that intensity variation, shadow effect, movement of object, movement of leaf etc., do not affect uniformly each pixel. As a result the unaffected pixels may have single component and affected pixels may have different number of components (some may have 2 or 3 or more, complex background pixels can be modeled by 3 to 7 components). This method dynamically assigns the number of models to each pixel while learning. Further, the parameter learning is adaptive, in the sense that the parameters corresponding to Gaussian models are updated only when it is required. Finally, for classification, we use method proposed by Stauffer and Grimson [2] with a small change in threshold.

3 Proposed Method

We model each pixel as a Mixture of Gaussians (MoG).

$$X_j(t = N) = \{x_{j1}, x_{j2}, x_{j3}, \dots, x_{jN}\} \quad (1)$$

denotes j^{th} pixel intensities for first N frames. Our algorithm works in stages: (a) learning, (b) background component identification, (c) classification and (d) smoothing. The block diagram of proposed method is shown in Fig. 1. The detailed description of each stage is presented next.

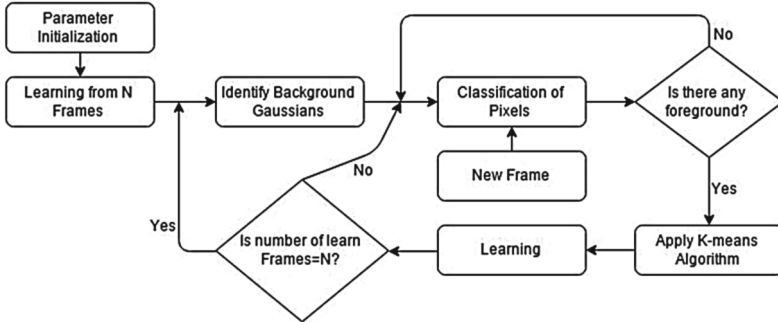


Fig. 1. Block diagram of the proposed method

3.1 Learning

In learning, we estimate parameters of Gaussians at every pixel from N frames. In Eq. (2), parameters $\mu_{ij}(t)$, $\sigma_{ij}(t)$, $W_{ij}(t)$ represent the mean, the standard deviation and weight of i^{th} Gaussian at time t for j^{th} pixel respectively. S is the

threshold used in BBS method, α and β are learning rates, B_j is the number of background components and K_j is the number of Gaussian components at j^{th} pixel. Moreover, $C_{ij}(t)$ denotes number of intensity matches accumulated at j^{th} pixel for i^{th} Gaussian up to time t .

Let's consider j^{th} pixel of first frame at time $t = 1$, $X_j(t = 1) = \{x_{j1}\}$ is first observation. We start with single Gaussian $K_j = 1$ (initially $i = 1$) and initialize mean $\mu_{ij}(t) = x_{j1}$, standard deviation $\sigma_{ij}(t) = V_0$, learning rate α , number of observation $C_{ij}(t) = 1$. With a new observation x_{jn} of n^{th} frame, we check whether it is matched to any of existing Gaussians as follows,

$$l = \arg \min_{\forall k: |x_{jn} - \mu_{kj}(t)| \leq \max(3\sigma_{kj}(t), S)} \{W_{kj}(t)\eta(x_{jn}; \mu_{kj}(t), \sigma_{kj}(t))\} \quad (2)$$

where, l is the component where a match is found and $\eta(x; \mu, \sigma)$ is the value of Gaussian function with mean μ and standard deviation σ at x .

After we find a match, we update parameter of l^{th} Gaussian component of j^{th} pixel according to given in method [3],

$$C_{lj}(t) \leftarrow C_{lj}(t) + 1 \quad (3)$$

$$\beta \leftarrow (1 - \alpha)/C_{lj}(t) + \alpha \quad (4)$$

$$\sigma_{lj}^2(t) \leftarrow (1 - \beta) * \sigma_{lj}^2(t) + \beta * (x_{jn} - \mu_{lj}(t))^2 \quad (5)$$

$$\mu_{lj}(t) \leftarrow (1 - \beta) * \mu_{lj}(t) + \beta * x_{jn} \quad (6)$$

$$W_{lj}(t) \leftarrow (1 - \alpha) * W_{lj}(t) + \alpha \quad (7)$$

where, β is learning rate. We do not put restriction on generation of new Gaussian components, number of Gaussians may increase if match is not found during learning. We control K_j after learning from N frames. We initialize new Gaussian components with $K_j = K_j + 1$, $C_{ij}(t) = 1$, $\sigma_{ij}(t) = V_0$, $\mu_{ij}(t) = x_{jn}$, $W_{ij}(t) = \alpha$. After generating new Gaussian(s) or updating existing Gaussian(s), we normalize weights according to following,

$$W_{kj}(t) \leftarrow W_{kj}(t) / \sum_{\forall k} W_{kj}(t), \quad \forall k. \quad (8)$$

We do not put any restriction on K_j , after learning from N frames each pixel might have different number of Gaussians. But, it is observed that most of the pixels have a single Gaussian (due to background). Sensitivity S is used as a threshold in basic background subtraction (BBS) method. Accuracy and speed of this method depends on α . Higher value of α shrinks background component and it results in more number of components and it further reduces speed. If the value of α is selected on lower side, more time is required to learn parameters. To make learning independent of α , M. Haque et al. [5] replaces threshold $3\sigma_{ij}(t)$ by $\max(3\sigma_{ij}(t), S)$. S is a minimum threshold that identifies pixel's membership with a component. Sensitivity parameter S prevents generation of unnecessary component due to shrinking of background components. Lower value of S gives better result as discussed in [7].

3.2 Identification of Background Gaussians

We model background pixels by mixture of Gaussians, 2 to 3 Gaussians are enough to model background but it is observed that pixels which are not affected by dynamic nature of background is modeled by only one Gaussian. Such single component is obviously due to background, there is no need to identify it with background components. To identify background components from multi-component models, we use the fact that, background components typically have higher weights and lower standard deviations, whereas foreground components have lower weights and higher standard deviations due to dynamic nature. Once learnt, we sort all components according to $W_{ij}(t)/\bar{\sigma}_{ij}(t)$ ratio (where, $\bar{\sigma}_{ij}(t) = 0.2989\sigma_{ij}^r(t) + 0.5870\sigma_{ij}^g(t) + 0.1140\sigma_{ij}^b(t)$ according to gray scale conversion from RGB). For background components this ratio is higher than foreground components. B_j is set of components which are identified as background. B_j is computed by,

$$B_j = \underset{b \in B_j}{\operatorname{argmin}} \sum_{i=1}^b W_{ij}(t) > T \quad (9)$$

here T is threshold (we choose $T = 0.7$) and its value depends on complexity of background. Higher value causes misidentification of foreground components as background components and lower value causes misclassification of some background components.

Controlling number of components (K_j): Since we have not put any restriction on number of components K_j during learning, the number of components keep on increasing. For classification, we need either background or foreground components. We identified B_j as set of background components so, we can set K_j as size of B_j . Discarding components other than B_j after learning from N frames, we are able to control the continuous increment in K_j .

3.3 Classification

To classify any new observation x_{jn} , we check whether it is part of any of B_j components. If it is part of any of background components then it will be considered as background otherwise foreground. To represent foreground and background in a frame we use an indicator function as suggested below,

$$x_{jn} = \begin{cases} 0, & \forall b \in B_j : |x_{jn} - \mu_{bj}(t)| \leq \max(3\sigma_{bj}(t), S) \\ 1, & \text{otherwise.} \end{cases}$$

Learning when it is required: To handle dynamic nature of background it is required to learn parameters of components continuously. Continuous learning increases complexity of algorithm which in turn increases time for foreground detection. To reduce the complexity, we do not learn parameters at every time instance because learning is required only when there is some change in background (i.e. foreground present, illumination change, insertion/deletion of background). During learning, we may remove previously identified background components B_j , we also allow modification of parameters of that components or

add extra components if required. After learning from N frames, during classification we allow generation of new components which might be due to change in background. Components having larger weights and smaller standard deviations are considered background components. This assumption reduces misclassification error.

Using K -means to reduce False Negatives: Pixel based MoG method can't detect foreground when foreground colors and textures are homogeneous and have low contrast with background. This situation results in false negative error. To improve on the results, we use the prior about the spatial smoothness of the foreground or the object in motion. We apply K -means algorithm to the portion of frame where foreground is detected. We choose $K_{sp} = 3$ (K_{sp} is number of classes in K -means) because there may be more than one object present in foreground or the object in foreground may have more than one color. Let's consider $Y = \{y_1, y_2, \dots, y_n\}$ as the intensities (in R,G,B format) in the frame where the foreground is detected (shown in Fig. 2(c)). We follow a stepwise procedure to correct errors as follows,

Step-1: Apply K-mean algorithm on data set Y and get centers $D_m (m = 1, 2, 3)$ of all three classes.

Step-2: Sort all three centers based on number of observations in each clusters and remove the center which has least number of observations. We also remove empty cluster centers. Now m is number of centers which are not removed.

Step-3: Identify rectangular blocks that cover every detected object. To reduce the time complexity, we apply the spatial smoothing, suggested in next step, only on identified object blocks.

Step-4: Consider the set of identified background pixels inside the object block, denoted by $Z = \{z_1, z_2, \dots, z_n\}$, we re-classify a pixel $z_k \in Z$ as foreground based on the following rule:

$$z_k = \begin{cases} 1, & \forall m, \|z_k - D_m\|^2 \leq th \\ 0, & \text{otherwise,} \end{cases}$$

where, th is threshold (for our experiment we have set $th = 10$). By using spatial information, we can also detect slow moving object whose some portion is wrongly classified by pixel based method. Results of proposed method on various datasets are presented in Fig. 2(e).

4 Experiments and Results

We have experimented with various datasets including *Wallflower* dataset [8] (frame size 160×120 24 Bits RGB) and compared results of our method qualitatively and quantitatively with state-of-the-art method proposed by Stauffer and Grimson [2] (with $K = 3$ and $\alpha = 0.01$). We have implemented proposed method and state-of-art methods in MATLAB-2009. For comparative timing analysis we run our codes on a system with Core-i3 1.80GHz processor and 4GB memory.



Fig. 2. (a) Original frames from video sequences, (b) Ground truth, (c) Results of method proposed by Stauffer and Grimson [2], (d) Results of hybrid method proposed by M. Haque et al. [5], and (e) Results of proposed method

4.1 Qualitative Analysis

To analyze effect of sensitivity S , we experimented with different value of S . It is observed that low S gives better results as suggested in [7]. We have selected $S = 15$, $N = 50$, $\alpha = 0.1$ and $V_0 = 10$ for simulations. Figure 2 shows a significant improvement in detection quality of our method as compared to methods suggested in [2, 5]. Higher value of α provides better adaption in case of illumination change but results in generation of spurious and redundant components due to shrinking of components. There is a significant reduction in the dependency of number of Gaussian components on α after using S as threshold in both learning and classification as suggested by [5]. With S , we may choose high learning rate $\alpha = 0.1$ which adapts to change in background quickly. In *Light Switch* video, the improvement due to high learning rate is evident. Results also show improvement

in detection for *Camouflage* and *Foreground Aperture* video data sets where the foreground intensities match background components. Spatial smoothing further improves the results by re-classifying misclassified background pixels.

4.2 Quantitative Analysis

Table 1 presents comparison of proposed method with method suggested in [2, 5]. For comparison, we have used precision, recall and accuracy. Recall is defined as fraction of number of detected foreground pixels from actual foreground ($Recall = TP/(TP+FN)$). Precision is defined as fraction of number of detected foreground pixels which are actually foreground ($Precision = TP/(TP+FP)$). Accuracy indicates how accurate the detection is ($Accuracy = (TP+TN)/(TN+TP+FN+FP)$). TP is number of truly detected foreground pixels. TN is number of truly detected background pixels. FP is number of falsely detected foreground pixels. FN is number of falsely detected background pixels. Occlusion of foreground and background occurs in *Camouflage* and *Foreground Aperture* videos that is corrected by K-means in proposed method. So, recall and accuracy are high for proposed method in comparison with the methods proposed in [2, 5]. Last row of Table 1 presents average precision, recall and accuracy. Compared to method in [2, 5] proposed method gives high average precision, recall and accuracy.

For timing analysis, we present our experiment results for all videos of *Wallflower* data set. Table 2, shows results of timing analysis of proposed method, and the others proposed in [2] and in [5]. Our method significantly reduces learning time by not assigning same number of Gaussian components (K) to each pixel. Using threshold as S , we have restricted generation of redundant components. Our method requires on an average 0.229 (we learn parameters when it is require and varable K_j) sec per frame in comparison to 2.6155s per frame by [2] and 2.9387s per frame by method in [5]. It is observed that learning requires most of the time for all the methods. With the reduction in number of redundant

Table 1. Video sequence in table appear in the same order in which the results are presented in Fig. 2.

Video sequence	MoG [2]			Hybrid method [5]			Proposed method		
	Preci	Rec	Acc	Preci	Rec	Acc	Preci	Rec	Acc
Waving trees	98.61	99.58	93.04	97.30	87.72	81.32	98.21	99.13	93.15
Camouflage	98.34	88.86	77.54	96.25	88.82	69.09	98.50	99.95	88.11
Bootstrap	97.67	38.09	81.65	75.68	46.28	78.84	81.21	46.36	79.31
Foreground aperture	88.69	46.66	77.03	86.03	32.57	74.40	96.55	99.40	89.52
Time of day	100	17.18	89.23	100	23.86	89.29	87.18	32.23	88.60
Light switch	3.82	26.13	12.96	7.37	70.13	8.90	76.76	45.55	81.94
Average	81.18	52.75	71.90	77.11	58.23	66.97	89.74	70.43	86.77

Gaussians and *learning when it is required* concept, we are able to reduce the average learning time significantly.

Table 2. Time analysis of all video of *Wallflower* dataset (in s per frame). Video sequence in table appear in the same order in which the results are presented in Fig. 2.

Video sequence	MoG [2]		Hybrid method [5]		Proposed method		
	Learn	classi.	Learn	classi.	Learn	classi.	correc.
Waving trees	2.2940	0.0359	1.7823	1.9035	1.1478	0.0866	0.0679
Camouflage	2.2503	0.0315	1.3642	1.4865	0.7921	0.0610	0.0247
Bootstrap	2.2840	0.0352	1.7726	1.7678	0.9469	0.0714	0.0528
Foreground aperture	2.5748	0.0352	1.4305	1.5082	0.7535	0.0446	0.0135
Time of day	2.2903	0.0377	1.3601	1.4565	0.8697	0.0507	0.0172
Light Switch	2.2790	0.0325	1.4027	1.4681	0.9280	0.0393	0.1844

5 Conclusion and Future Work

In this work, we have proposed a method for background-foreground separation which is real-time and shows better accuracy. Other methods discussed are either time efficient but lack accuracy or are accurate but not time efficient. Proposed method is efficient in time and gives better results (both qualitatively and quantitatively) in many adverse conditions. Using spatial smoothness as a prior, we have shown that K -means successfully reduces false negative error when foreground and background intensities are homogeneous. It is interesting to note that this method can detect occluded portion of foreground with background. Time saving is essentially because of removal of redundant components from bookkeeping. Additionally we have used the fact that most of the pixels in a frame are background and allowed every pixel to have variable K_j number of Gaussians. The other significant factor for time saving is due to *learning when it is required* approach.

As an extension to this work, we would like to address a few other issues related to shadow in the image and illumination changes. Shadow in video introduces error in detection of object and it also adds up to generation of redundant components which in turn increases computational time. We would like to incorporate illumination invariance in our approach to make detection scheme more robust.

References

1. Wren, C., Azarbayejani A., Darrell, T., Pentland, A.: Pfunder: real-time tracking of the human body. In: IEEE Proceedings of the Second International Conference on Automatic Face and Gesture Recognition (1996)

2. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2 (1999)
3. Lee, D.-S.: Effective mixture learning for video background subtraction. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 827–832 (2005)
4. Shimanda, A., Arita, D., Taniguchi, R.: Dynamic control of adaptive mixture of gaussians background models. In: IEEE International Conference on Video and Signal Based Surveillance (2006)
5. Haque, M., Murshed, M., Paul, M.: A hybrid object detection technique from dynamic background using Gaussian mixture models. In: IEEE 10th Workshop on Multimedia Signal Processing (2008)
6. Singh, A., Jaikumar, P., Mitra, S.K.: A sampling-resampling based Bayesian learning approach for object tracking. In: IEEE Sixth Indian Conference on Computer Vision, Graphics and Image Processing, ICVGIP (2008)
7. Nascimento, J.C., Marques, J.S.: Performance evaluation of object detection algorithms for video surveillance. In: Seventh IEEE Transactions on Multimedia, vol. 8, pp. 761–774 (2006)
8. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision (1999)