# Spatial Mixture of Gaussians for dynamic background modelling

Sriram Varadarajan, Paul Miller and Huiyu Zhou
The Centre for Secure Information Technologies (CSIT)
Queen's University Belfast
{svaradarajan01,p.miller,h.zhou}@qub.ac.uk

## Abstract

*Modelling pixels using mixture of Gaussian distributions is a popular approach for removing background in video sequences. This approach works well for static backgrounds because the pixels are assumed to be independent of each other. However, when the background is dynamic, this is not very effective. In this paper, we propose a generalisation of the algorithm where the spatial relationship between pixels is taken into account. In essence, we model regions as mixture distributions rather than individual pixels. Using experimental verification on various video sequences, we show that our method is able to model and subtract backgrounds effectively in scenes with complex dynamic textures.*

## 1. Introduction

Probabilistic modelling of backgrounds is used extensively in scene change detection. Of all the approaches used in the probabilistic modelling of pixels, mixture of Gaussians (MoG) is one of the most popular algorithms. Even though there was prior research on modelling pixels based on mixture models [1], the MoG modelling technique proposed by Stauffer and Grimson [2] provides the basis for almost all recent research in this topic. They modelled each pixel in an image as an online MoG model which was learned using a recursive filter. While this works well for a static background subject to gradual lighting changes in the scene, it fails to handle any dynamic changes in the background such as leaves swaying, or water waves.

One way to address this issue is to extend the modelling of a pixel's temporal variation to include its spatial neighbourhood. This is because these type of changes (waves rippling, leaves swaying) in the background can be viewed as dynamic textures [3] which exhibit certain spatio-temporal stationarity properties. In this paper, we propose a generalisation of the MoG approach that incorporates spatio temporal data to handle dynamic textures in the scene. Our model is allowed to update from pixels within a spatial neighbourhood thus incorporating small changes in the background from frame to frame such as the motion of waves or leaves in a scene.

In the following sections, we review previous work in this field, and derive novel on-line update equations, using expectation maximisation (EM), for modelling scenes containing dynamic textures. Finally, we provide experimental results to verify the working of our algorithm. We also compare the performance of our algorithm with other well-known background subtraction algorithms.

## 2. Related Work

The system proposed by Stauffer and Grimson [2] is the de facto standard for probabilistic modelling of background pixels based on Gaussian mixtures. They used an online k-means approximation to incrementally learn new observations with a learning parameter that ranges from zero to one. The model is thus learned with an exponential decay of the past history. This approach is a per-pixel approach, with the pixels assumed to be independent of each other. There have also been various other pixel level methods [4] that have improved upon this method; however, all these methods fail to address the issue of dynamic backgrounds effectively. Whilst the original k-means updating equations, and variants thereof, were heuristic, [5] presented a rigorous theoretical derivation for them using an EM framework.

Dynamic textures are defined as sequences of images in a scene that exhibit certain stationarity properties in time [3]. Examples range from waves in the sea, movement of branches in the wind, smoke, fire etc. All these phenomena could be termed periodic because they tend to repeat within a small neighbourhood of the image. By using spatial data along with the pixel temporal values, these dynamic textures can be modelled in a better manner. Many researchers have indeed tried to use spatial data to model the dynamic background in scenes. Some notable examples include Sheikh and Shah [6], who used a kernel-based approach that took into account neighbouring pixel locations, whilst modelling the background of a particular pixel. However, in their approach, they had to maintain a history of frames based on their learning rate. This results in a high dimensionality problem as it requires storage of a lot of data. Jodoin et al. [7] proposed a combined spatial and temporal framework, with the assumption that a pixel's spatial and temporal variations can be modelled in a similar manner. However, the authors themselves state that this assumption does not always hold true, as for example, in the case of a temporally periodic event such as a blinking light.

A background model based on spatio-temporal texture was proposed by Yumiba et al. in [8]. They handled local and global changes in the background by using a spatio-temporal texture, called a "Space-Time Patch", which uses gradients to describe the motion and appearance of objects. However, their method is just used for detecting the presence of motion in a scene and not for actual segmentation. Dalley et al. [9] proposed a heuristic generalisation of the MoG model to handle dynamic textures. They allowed models to be generated from any/all pixels in a neighbourhood window by using multiple simultaneous measurements because different Gaussians could independently affect the same pixel at the same time. These calculations increase the computational complexity of the algorithm. Our method is motivated by their approach; however, instead of providing heuristic equations, we provide a theoretical justification using EM theory, which is computationally less burdensome as it removes the need for simultaneous measurements.

## 3. Spatial MoG

As shown in [5], the original MoG algorithm proposed by Stauffer and Grimson is an online EM algorithm which estimates the maximum likelihood of the observed data. The pixel data is grouped into clusters or mixtures that model this data over time in a k-means like manner. We extend the theory in [5] to produce a generalisation of the algorithm, which includes spatial data taken from a pixel's neighbourhood region. We first look at the algorithm from a batch EM viewpoint, to provide a theoretical justification of our work, and then we extend this into a generalised online version of pixel mixture modelling based on spatio-temporal relationships.

### 3.1. Batch mode updates

Consider the data $X = \{x_1, x_2, ..., x_i, ..., x_N\}$ where $N$ is the number of data samples and $i = 1, 2, ..., N$ denotes the index of the data samples. The neighbourhood space of $i$, $\mathcal{R}_i$ is defined at any instant as a collection of $r$ data values in the neighbourhood centered around $x_i$. Let $k = 1, 2, ..., K$ be the number of classes the data samples could belong to, at each position $i$. Let $z_{ik}$ denote the membership of any data sample to a cluster at position $i$, where $z_{ik} = [0, 1]$. Furthermore, let $\theta = \{\mu_{ik}, {\sigma^2}_{ik}, \omega_{ik}\}$ be the parameters of each class $z_{ik}$ that the data samples can belong to. Now, any particular sample can be modelled as a mixture distribution which is a combination of both colour and space. Over time, the mixtures are updated from pixels that are close in colour and fall within the region of the mixtures.

In batch mode, the EM algorithm is used to find the maximum likelihood estimate of the log likelihood function of the observed data [10]. The expectation over the posterior distribution of the latent variable $z_{ik}$ given the data and the parameters is

$$E_{p(z_{ik}|x_q,\theta^{old})}(z_{ik}) = \sum_{q \epsilon \mathcal{R}_i} \gamma_q(z_{ik}) \tag{1}$$

where

$$\gamma_q(z_{ik}) = \frac{\omega_{ik} * \mathcal{N}(x_q|\mu_{ik}, {\sigma^2}_{ik})}{\sum_{i \in \mathcal{R}_i} \sum_{j=1}^{k} \omega_j * \mathcal{N}(x_q|\mu_{ij}, {\sigma^2}_{ij})} \tag{2}$$

Now, this is different from the traditional batch MoG in that this expected value depends on the neighbouring locations whereas in the traditional batch approach it only depends on the current location. The subscript $q$ for $\gamma$ here indicates that the expectation of the mixture, based on its neighbourhood data, $r$, is maintained separately. Outside of the cluster's neighbourhood, the data does not influence the cluster, therefore, the expectation values are assumed to be zero.

The update parameters in the maximisation step can be derived by maximising the expected value of the log likelihood function with respect to the different parameters (the derivation of which is beyond the scope of this paper). The final update equations are obtained as follows:

$$\mu_{ik} = \frac{1}{N_{ik}} \sum_{q \epsilon \mathcal{R}_i} \gamma_q(z_{ik}) * x_q \tag{3}$$

$${\sigma^2}_{ik} = \frac{1}{N_{ik}} \sum_{q \epsilon \mathcal{R}_i} \gamma_q(z_{ik}) * (x_q - \mu_{ik})^2 \tag{4}$$

$$\omega_{ik} = \frac{N_{ik}}{N} \tag{5}$$

where $N_{ik}$ is given by

$$N_{ik} = \sum_{n} \sum_{q \epsilon R_i} \gamma_q(z_{ikn}) \tag{6}$$

### 3.2. Online mode updates

In order to extend the batch algorithm to the online situation, stochastic gradient descent technique is used [5] resulting in the following equations:

$$\mu_{qk}^{t} = (1 - \rho)\mu_{qk}^{t-1} + \rho(x_i^t) \tag{7}$$

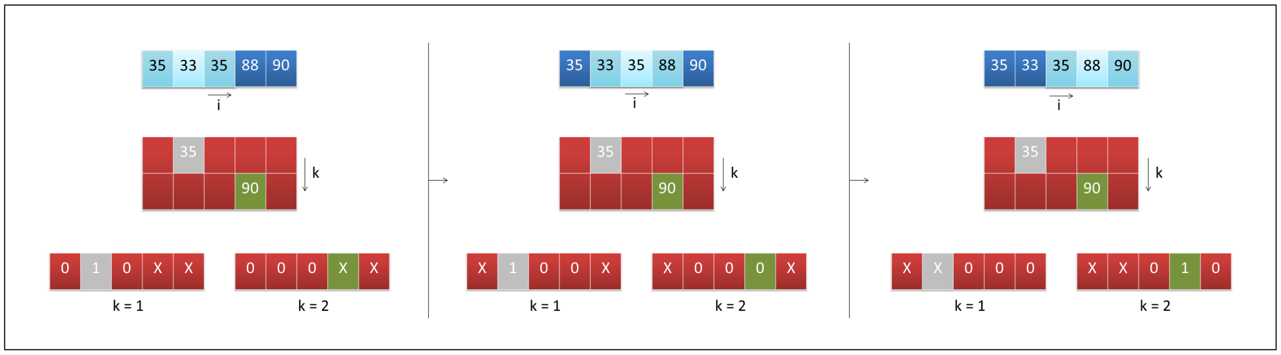

Figure 1. An illustration of the model update. The light boxes in the top row indicate the current pixel and its neighbourhood. The middle row indicates the possible clusters the data could belong to. The bottom row shows the cluster updated for the current pixel index (X denotes that the clusters are not considered as they fall outside the neighbourhood of the pixel). The mixture updates are performed using equations 7-9

$$\sigma^{2}{}^{t}_{qk} = (1-\rho)\sigma^{2}{}^{t-1}_{qk} + \rho({x_i}^{t} - \mu^{t-1}_{qk})^2 \tag{8}$$

$$\omega^{t}_{qk} = (1-\alpha)\omega^{t-1}_{qk} + \alpha \tag{9}$$

We now explain the Spatial MoG algorithm incorporating the above equations in detail. This is followed by the explanation of the model update example shown in fig. 1.

Algorithm: Spatial Mixture of Gaussians

1. Given an observation space $\{x_1, x_2, ..., x_n\}$, initialise the mixture parameters $\{\mu_k, {\sigma^2}_k, \omega_k\}$ by assigning random values from the neighbourhood for the means, high variances and equal weights. For each observation $x_t$, all related observations in its neighbourhood defined by a window size $r$ are also taken into consideration.

2. Calculate the most likely mixture for the current observation (over the entire neighbourhood). This is usually approximated by a distance measure between the mean of the mixture and the pixel intensity of the current observation. In our experiments, Euclidean distance was used.

3. Compare the distance of the above mixture with a threshold value which is usually a scaled factor of the standard deviation of the mixture. This indicates whether the pixel matches the mixture model or falls outside the model.

4. If a match is found, update the mixture parameters of the above mixture by using the equations 7-9

5. If no match is found, replace the mixture with the lowest weight (at the current observation location) with a new mixture by initialising it again.

6. Normalise the weights in the current region appropriately so that the weights are balanced even if there are multiple updates to a mixture from pixels around its location.

7. The background model is then built with Gaussians having high weights in the region. If the current observation falls within this model, it is classified as a background pixel; otherwise it is classified as a foreground pixel.

In fig. 1, we illustrate how the online model is updated for a 1-d case with a neighbourhood size $r = 3$. The top row represent the $N$ data samples for one time instant, with $i$ representing the data indices. The light coloured boxes in this row indicate the current pixel and its neighbourhood. The left column shows the situation for the update for $i = 2$. The pixel value is 33 and the neighbourhood pixel values to either side are 35. The middle row of boxes represent the possible clusters that have been learnt by the model thus far. For illustrative purposes, only the relevant clusters that are updated are shown here but other clusters that may be present are assumed to be unimportant in this example. For this reason, the cluster 90 at $i = 4$ is shown at $k = 2$ and not $k = 1$. Note here that the total number of clusters at each pixel location is, say $K = 2$, as in normal MoG, but these clusters can be updated from any pixel in their neighbourhood space unlike normal MoG where the clusters are located at the same location as the pixel that influence them. Assume that the models have been updating for a certain time period and the figure shows the model at the current time instant. The bottom row shows the cluster allocation based on a hard choice from the cluster likelihoods. Thus for the pixel value 33 at $i = 2$, a region wide search provides the cluster at $i = 2$ and $k = 1$ as the most probable cluster, which is denoted by a 1 at that location and the other clusters in the region are denoted by zeros, i.e. at $i = 1$ and 3 and also at $i = 2$ and $k = 2$. The $X$ marks at $i = 4$ and 5 indicate that these clusters are not considered as they fall outside the neighbourhood space of $i = 2$. Based on this information, the parameters of the cluster at $i = 2$ and $k = 1$ are updated by the equations 7-9. We then move the window one step to the right and we obtain the situation shown in the middle column. In this case, the most probable cluster in the neighbourhood of the pixel value at $i = 3$ is found to be located at $i = 2$ and $k = 1$ and hence that cluster is again updated using the update equations given above. This is the key difference between our method and the original approach in that a cluster at a different pixel location can be updated provided it falls under the neighbourhood of the current observation. The weights are normalised at each step over the region that is currently under consideration. Because the cluster at $i = 2$ and $k = 1$ is updated twice, when the weights are normalised in that region, it will result in a higher weight for that cluster and lower weights for other clusters that may be present in the same neighbourhood. Then we move the window again one step to the right and this case is shown in the right column. Here, the pixel value is 88 and from all the clusters in the neighbourhood of $i = 4$, the cluster at $i = 4$ and $k = 2$ is found to be the most probable cluster and hence, it is updated as above. Now, if a new pixel is encountered that does not belong to the clusters in its region, a new cluster is created at the same location of the pixel with a low weight and high variance and the new pixel intensity as the mean.

In our model, it can be seen that the dynamic background information is taken into account by updating the mixtures from pixels that lie within a spatial region. The learning rate controls the rate at which the mixtures are learnt over time. Since there can be multiple updates in a single time frame, this model can update faster than the conventional approach. Although the modelling uses block level properties, it is still a pixel level approach and hence provides a well defined output unlike other block based techniques using spatial relationship that provide block level results.

Another advantage of this model is that it can be extended to general probabilistic models like kernel density estimates for non-parametric modelling and also to other families of probability distributions like mixture of Poissons for low lighting conditions. It can also be generalised further to different feature sizes and block based approaches.

## 4. Experiments

To evaluate the proposed spatial MoG algorithm, we tested its performance on four video sequences containing dynamic texture backgrounds. The first three are well-known sequences that have been extensively used for testing by the video analytics research community. These are a bottle sequence (240x320) [11], beach sequence (128x160) [12], and waving trees sequence (120x160) [13]. In addition, we have evaluated the spatial MoG performance against a sequence acquired from a CCTV camera on board a moving bus. This is the most complex sequence of all and it involves a moving bus with a passenger in it. The complexity of the scene comes from the window region of the

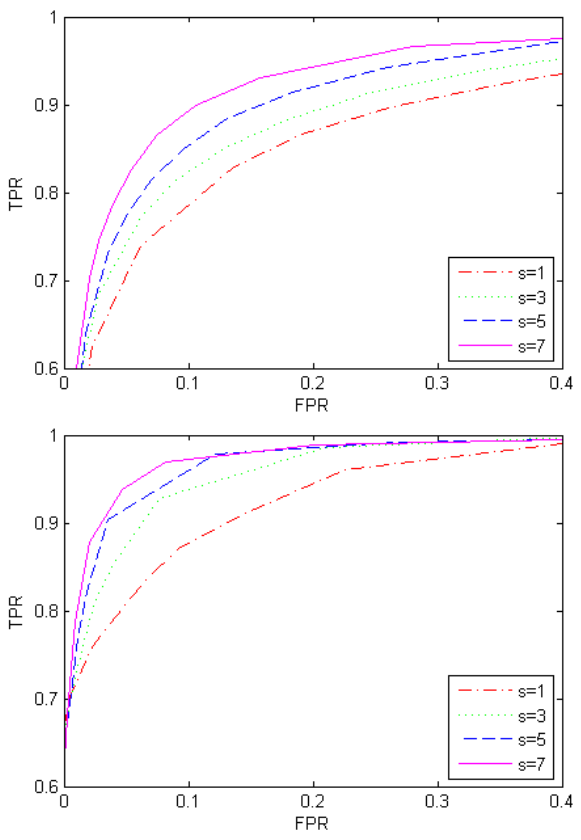

Figure 2. ROC Curves for the water sequence and the beach sequence

bus where the outside scene varies dynamically. Also, there is a huge illumination change in the bus due to shadows falling in the bus as it changes direction. These scenarios together make this sequence a challenging one.

We first investigated the effect of the neighbourhood window parameter size on the algorithm performance. Fig. 2 shows ROC curves for both the water and beach sequences for window sizes of 1, 3, 5 and 7 (a window size of 1 corresponds to Stauffer and Grimson's original algorithm). The ROC curves for the other two sequences were found to follow a similar pattern. Analysis of Fig. 2 confirms that as the neighbourhood window size increases the performance of the algorithm improves, indicating that the spatial modelling can cope with dynamic background textures. Fig. 3 shows representative output frames from each sequence for different window sizes. Qualitative analysis reveals that as the neighbourhood size increases, there is a decrease in false positives. This is because smaller neighbourhood models cannot accurately learn the dynamic textures that move out of the neighbourhood. However, larger neighbourhood models are able to suitably adapt to this change and include them in their model. This results in fewer false positives with larger windows. In the bus sequence, it should be noted that the algorithm is able to adapt to significant illumination changes. This is visible in the aisle of the bus, where the shadow falling in the bus causes a drastic change in illumination. The key reason for including the bus sequence is because of its highly dynamic window region. This region has a wide range of colours; green from vegetation, brown/gray due to buildings, white due to the pavement etc. Now, the original approach fails to work here because, as the pixels are assumed to be independent, it models only the colour that first appears in the video sequence, and fails to handle any further variation. But in our approach, because the model is maintained over a spatial neighbourhood, any movement in the window region will be modelled appropriately, thus enabling it to extract foreground only.

Fig. 4 shows a comparison of the results from different well known background subtraction approaches like Eigen background [14], ViBe [15] and MGM-UM [16]. A simple median filtering has been applied uniformly to all the images to remove noise in them. The number of pixel errors (False Positives and False Negatives) for each algorithm/sequence combination is shown in Table 1. The total number of errors from all sequences (TE) for each algorithm is also shown. It can be seen that our algorithm performs better than most algorithms and also produce comparable results to ViBe which is one of the best background subtraction algorithms currently in literature. The pixel models of ViBe are updated exclusively with observed pixel values compared to our approach where we approximate the model with a weighting factor. Further, their background and foreground models are mutually exclusive whereas there is a possibility of outliers present in our models. However, our results are still similar to their results and we believe our approach can be further improved by adaptively varying the neighbourhood size and the classification threshold. Fig. 5 shows the output of a few key frames from the waving trees sequence for all the above mentioned algorithms. Here, the branches of the tree are wildly waving in the background and towards the end of the video sequence, a person enters the scene. The results show that our method is able to perform both dynamic background subtraction as well as foreground segmentation very well.

Finally, the optimal choice of neighbourhood size depends on the dynamics of the background texture. To illustrate this, consider the water sequence, which is twice the size of the beach sequence. Here, the wave motion covers a larger region than the wave motion in the beach sequence. This indicates that the larger neighbourhood is required to subtract the background in the water sequence compared to the one required for the beach sequence. This can be seen in the ROC results, where the results with neighbourhood sizes of 5 and 7 for the beach sequence are almost the same, whereas there is a marked difference between them for the water sequence. This indicates that there is a cut-off size that is dependent on the magnitude of motion of the background texture. Any window size above this cut-off will not show a noticeable difference in the results. Also, we noticed in our experiments that the learning rate helps learn the model faster than the conventional algorithm because multiple updates can be done in one iteration. However, this can be suitably adjusted if we would like the model to update slower, i.e. foreground objects should not move to background immediately.

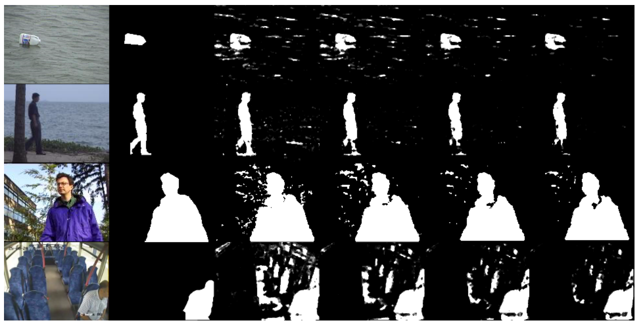
Figure 3. Background subtraction results for the different video sequences. First Column - Original images; Second Column - Ground Truth; Third Column - OriginalMoG (s = 1); Fourth, Firth and Sixth Columns - SpatialMoG (s = 3, 5 and 7)

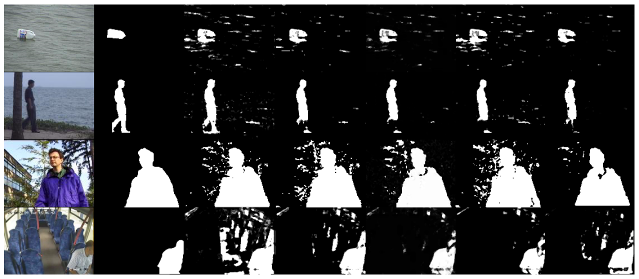
Figure 4. Comparison of results from different algorithms. First Column - Original images; Second Column - Ground Truth; Third Column - OriginalMoG; Fourth Column - Eigenbackground [14]; Fifth Column - ViBe [15]; Sixth Column - MGMUM [16]; Seventh Column - SpatialMoG

| Algorithm | Errors | Water | Beach | Trees | Bus | Total Errors |
|---|---|---|---|---|---|---|
| MoG | FN | 351 | 75 | 302 | 212 | 26496 |
| | FP | 7001 | 1553 | 1694 | 15308 | |
| SMoG | FN | 423 | 336 | 199 | 1020 | 8767 |
| | FP | 2093 | 231 | 688 | 3777 | |
| Eigen | FN | 388 | 270 | 48 | 468 | 9809 |
| | FP | 2100 | 222 | 1942 | 4371 | |
| MGMUM | FN | 564 | 334 | 153 | 1436 | 10972 |
| | FP | 4296 | 422 | 2182 | 1585 | |
| ViBe | FN | 592 | 333 | 186 | 1201 | 7757 |
| | FP | 2160 | 168 | 1424 | 1693 | |

Table 1. Quantitative Analysis

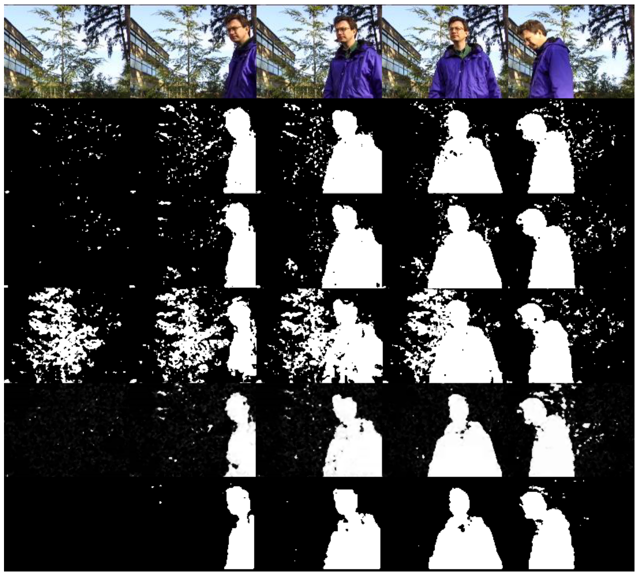

Figure 5. Comparison of results from different algorithms for the waving trees sequence. First Row - Original images; Second Row - OriginalMoG; Third Row - Eigenbackground [14]; Fourth Row - MGMUM [16]; Fifth Row - ViBe[15]; Sixth Row - SpatialMoG

## 5. Conclusions

In this paper, we proposed a region based Mixture of Gaussians approach by deriving update equations from EM theory to handle dynamic background in video sequences. By experimental evaluation, we showed that this approach works well in modelling dynamic backgrounds in well known sequences. Future work will include extending the algorithm to work with different feature sizes instead of individual pixels to improve the background subtraction further without compromising much on the foreground definition. As mentioned earlier, we also look to combine the motion information of the texture to adapt the neighbourhood of the model.

## References

[1] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," in *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pp. 175–181, 1997.

[2] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2246–2252, 1999.

[3] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[4] T. Bouwmans, F. E. Baf, and B. Vachon, "Background modeling using mixture of gaussians for foreground detection - a survey," *Recent Patents on Computer Science*, vol. 1, no. 3, pp. 219–237, 2008.

[5] H. Wang and P. Miller, "Regularized online mixture of gaussians for background subtraction," in *8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 249–254, 2011.

[6] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1778–1792, November 2005.

[7] P.-M. Jodoin, M. Mignotte, and J. Konrad, "Statistical background subtraction using spatial cues," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, pp. 1758–1763, December 2007.

[8] R. Yumiba, M. Miyoshi, and H. Fujiyoshi, "Moving object detection with background model based on spatio-temporal texture," in *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 352–359, 2011.

[9] G. Dalley, J. Migdal, and W. E. L. Grimson, "Background subtraction for temporally irregular dynamic textures," in *IEEE Workshop on Applications of Computer Vision*, pp. 1–7, 2008.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood for incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1–38, 1977.

[11] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust kalman filter," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 44–50, October 2003.

[12] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proceedings of the eleventh ACM international conference on Multimedia*, pp. 2–10, 2003.

[13] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Seventh International Conference on Computer Vision*, pp. 255–261, 1999.

[14] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 831–843, 2000.

[15] O. Barnich and M. Van Droogenbroeck, "ViBe: a powerful random technique to estimate the background in video sequences," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009)*, pp. 945–948, 2009.

[16] F. Baf, T. Bouwmans, and B. Vachon, "Type-2 fuzzy mixture of gaussians model: Application to background modeling," in *Proceedings of the 4th International Symposium on Advances in Visual Computing*, ISVC '08, pp. 772–781, Springer-Verlag, 2008.

[17] A. Sobral, "BGSLibrary: A opencv c++ background subtraction library," 2012. Software available at `http://code.google.com/p/bgslibrary/`.