Stochastic approximation for background modelling

Ezequiel López-Rubio, Rafael Marcos Luque-Baena


Abstract

Many background modelling approaches are based on mixtures of multivariate Gaussians with diagonal covariance matrices. This often yields good results, but complex backgrounds are not adequately captured, and postprocessing techniques are needed. Here we propose the use of mixtures of uniform distributions and multivariate Gaussians with full covariance matrices. These mixtures are able to cope with both dynamic backgrounds and complex patterns of foreground objects. A learning algorithm is derived from the stochastic approximation framework, which has a very reduced computational complexity. Hence, it is suited for real time applications. Experimental results show that our approach outperforms the classic procedure in several benchmark videos.

Keywords: background modelling, probabilistic mixture models, stochastic approximation, unsupervised learning

1. Introduction

Research in intelligent video surveillance systems is an area in which scientific community has made great efforts for the last years, conceiving new ideas, developing new theoretical frameworks and implementing new algorithms which have stimulated the increase of the number of quality contributions [1, 2, 3].

This kind of systems can be applied to a wide range of potential applications, such as security and safety systems in important buildings, traffic surveillance in cities and highways, or supervision of suspicious behaviour in supermarkets or in public means of transport. In fact, any area which requires applications with real-time monitoring of motion objects from a camera video stream is susceptible of using intelligent video surveillance systems. However, the increase of the research in video surveillance has not led to a huge rise of intelligent surveillance systems in operation. Although several commercial frameworks have already been defined to deal with motion objects both in outdoor and indoor scenes, nowadays, passive supervision systems keep being used in many areas, in which the number of cameras exceeds the capability of human operators to analyse and monitor them.

Several steps are required for a typical video surveillance system to reach its objective. At first, a video object segmentation method is required to obtain the objects in motion of the stream. Subsequently, a tracking algorithm is applied to identify the objects in several frames of the sequence. A matching between each blob (or set of blobs) and each tracked object must be done. Finally, an algorithm to detect the object behaviour is used to understand and analyse what it is happening in a scene. In general, the first stage is considered as the crucial part of the system since the rest of modules depend largely on it. In addition to

*Corresponding author

Email addresses: emeqr@lcc.uma.es (Ezequiel López-Rubio), rmlaque@lcc.uma.es (Rafael Marcos Luque-Baena)

Preprint submitted to Computer Vision & Image Understanding November 15, 2010
that, low time complexity is demanded at object detection stage in order to carry out the entire process in real time.

Every video surveillance system starts its activity by detecting moving objects in the scene. A comparison among different techniques can be found in some important works [4, 5, 6]. In essence, the aim of this task is to separate pixels corresponding to the foreground motion objects from those corresponding to the stationary background ones. Different kinds of approaches can be found in the literature to model this problem, such as techniques based on optical flow [7, 8, 9], whose main disadvantage is that it requires very high computational time; frame difference, which is efficient but inaccurate and unreliable; or background subtraction, which models the background by comparison with the frames of the sequence. In particular, the development of a background model has been widely used by a large number of papers as a way to detect the foreground objects in each frame. This model can be as complex as it is required, from the simplest ones which are just based on an image background, statistical parametric models which represent the colour signal of each pixel, to even more ambitious models which analyse and maintain information about a set of scene features. The complexity of these background models directly influence how complex or efficient their update process is.

However, it is not easy to find a low-complexity method which is able to handle all the unexpected situations, gets suitable and unbeatable results and whose efficiency and robustness is high. The simplest process, which involves the subtraction of the current analysed frame from the previously calculated background image, is not enough to deal with a great number of difficulties which make the object detection process rather complex. Unfavourable factors such as both abrupt and progressive illumination changes in the scene, cast foreground object shadows on the background, noisy frames because of a poor quality image source, or repetitive movements of stationary objects like waving tree branches must be taken into account by the developed object detection method.

Different approaches in the literature try to find solutions to the previous drawbacks. In [10], a combination between background subtraction and frame difference is presented, dealing with sudden illumination changes and noise reduction. The background update is performed using a moving $W$-window and the foreground mask is obtained by using an ad-hoc thresholding process. Both parameters, $W$ and the threshold $\sigma_T$ are empirically chosen depending on the analysed scene. Another method which uses an sliding window with some recent past $L$ frames is included in [11], where an estimation of the background image of the sequence is obtained by means of a temporal median background process over the window. A novel variation of frame difference is presented in [12], where a method based on double frame differences and edge encloses is described, which stands out mainly in its reduced time complexity. There are two different low-level segmentation modules depending on the lighting conditions, since the system is designed to operate both daytime and at night. It is considered as a competitive algorithm for detecting simple rigid objects only, like in traffic sequences.

A model based on the Kalman filter is developed in [13, 14], with the aim of compensating the illumination variability of each pixel. They replace the sliding window with an online adaptive computation of the background model, whose update is controlled by a learning rate $\alpha$. In [15], Wren et al. describes its Pfinder method by modelling the background for each pixel using a normal distribution. As a modified version of this method, Stauffer et al. in [16] and [17], which complete and improve the probabilistic approach outlined in [18], present a well-known theoretical framework for the upgrade background, based on a mixing process of Gaussian distributions for each pixel, using an expectation-maximisation (EM) algorithm to update them. This statistical approach is more robust in scenes with many moving objects and lighting changes, and it is one of the most cited techniques in the literature. Because of the fact that sometimes it is not necessary to use a fixed number of distributions, in [19] the number of Gaussians is estimated online, thus reducing slightly the time complexity of the algorithm.
In order to determine as fast as possible whether a pixel value \( t \) belongs to the background distribution in the methods which model pixels as probability density functions, a quick implementation of the probability \( p(t | \text{Back}) \) is performed for all the statistical approaches \([15, 16, 19]\). It is assumed that a background match is defined as a pixel value within \( \mathcal{N} \) standard deviations of a distribution. This assumption is stated in \([16]\) and \([20]\). This parameter \( \mathcal{N} \), which is considered as a threshold for Gaussian tiles in standard deviations, varies depending on the amount of noise which is observed in the scene.

Unlike the previous parametric methods, Elgammal et al. \([21]\) model the background using non-parametric methods, more robust and invariant especially in outdoor scenes with a huge background variability. Within the statistical methods, Haritaoglu et al. \([22]\) build a model named \( W4 \), to represent each pixel with three values: minimum and maximum values, and the maximum intensity difference between consecutive frames during the training period. These three values are updated periodically. McKenna et al.\([23]\) use an adaptive background model with colour and gradient information to reduce the influences of shadows and unreliable colour cues. Under a Bayes decision framework, Li et al. \([24]\) developed an approach which separates the analysis of the stationary pixels from the moving objects in scenes with complex backgrounds, by using different feature vectors for each purpose. A reference background image is also maintained, by performing a IIR filter in order to handle the gradual changes for stationary background objects.

Other kind of approaches which analyse the input data by using a phase-divided structure have been proposed in the literature \([25]\). These models combine different methods to solve some of the main problems associated to the segmentation task. Basically, they have a low-level layer which processes the data pixel by pixel, and then, several additional modules are applied in order to improve the segmentation quality. Morphological operations such as closing and opening as well as connected components algorithms and the elimination of small objects should be used for this purpose.

Our proposal departs from the classical algorithms based on the EM framework. We propose to use stochastic approximation methods \([26, 27, 28, 29, 30, 31, 32, 33, 34, 35]\) which are based on the Robbins-Monro algorithm \([36]\). These techniques have already been used to replace or modify typical EM strategies in a wide range of problems \([37, 38, 39, 40, 41]\), and they have been applied in image and video processing \([42, 43, 44, 29, 45]\). A key advantage of stochastic approximation is that it is conceived to process input samples online. On the other hand, EM is commonly associated with batch processing of a complete set of input data. In video segmentation, the online processing nature of stochastic approximation can be of paramount importance, since data is generated in real time. Moreover, this framework has an intrinsic tendency to assign more relevance to the newer data with respect to the older ones, which reinforces its suitability for this task. As we will see, it is also a robust strategy which needs a minimal number of tuneable parameters.

The structure of the paper is as follows. Section 2 presents the new video segmentation model, its learning rule and the initialization and implementation details. Section 3 is devoted to experiments. First we explain our choice of the competing models. Then an evaluation methodology is developed, with both qualitative and quantitative performance measures. And, of course, the experimental results are also presented. In Section 4 we discuss the novel features and the advantages of our method. Finally, conclusions are drawn in Section 5.

2. The model

2.1. Model definition

We propose to use the stochastic approximation framework to train a mixture which models the distribution of pixel values \( t(x) = (t_1(x), t_2(x), t_3(x)) \) at position \( x = (x_1, x_2) \). There is one Gaussian component for the background and one uniform component for the foreground. This leads to the following probabilistic
model for the distribution of pixel values at any given position, where we drop the position indices $x$ for the sake of simplicity:

$$ p(t) = \pi_{\text{Back}} p(t|\text{Back}) + \pi_{\text{Fore}} p(t|\text{Fore}) = $$

$$ \pi_{\text{Back}} G(t|\mu_{\text{Back}}, C_{\text{Back}} + \Psi) + \pi_{\text{Fore}} U(t) \quad (1) $$

We have:

$$ G(t|\mu, \Sigma) = (2\pi)^{-D/2} \det(\Sigma)^{-1/2} \exp \left( -\frac{(t - \mu)^T \Sigma^{-1} (t - \mu)}{2} \right) \quad (2) $$

$$ U(t) = \begin{cases} 
1/\text{Vol}(H) & \text{iff } t \in H \\
0 & \text{iff } t \notin H 
\end{cases} \quad (3) $$

$$ \mu_{\text{Back}} = E[t|\text{Back}] \quad (4) $$

$$ \mu_{\text{Fore}} = E[t|\text{Fore}] \quad (5) $$

$$ C_{\text{Back}} = E[(t - \mu_{\text{Back}})(t - \mu_{\text{Back}})^T|\text{Back}] \quad (6) $$

$$ \forall i \in \{\text{Back}, \text{Fore}\}, R_{ni} = P(i|t_n) = \frac{\pi_i p(t_n|i)}{\pi_{\text{Back}} p(t_n|\text{Back}) + \pi_{\text{Fore}} p(t_n|\text{Fore})} \quad (7) $$

where $A$ is the support of the uniform pdf, $\text{Vol}(H)$ is the $D$-dimensional volume of $H$, and $\Psi$ is a constant diagonal matrix which accounts for the quantization noise due to lossy compression (see Subsection 2.3). In background modelling, we typically have $D=3$ for tristimulus pixels, and $H$ spans all the colour range:

$$ H = \{(t_1,t_2,t_3) | t_1,t_2,t_3 \in [0,v]\} \quad (8) $$

with $\text{Vol}(H) = v^3$. In most cases the pixels are given in RGB, but $t$ could also be expressed in any other colour space, such as L*u*v*. On the other hand, if we have colour values with 8 bit precision then $v=255$. It must be highlighted that $\mu_{\text{Fore}}$ is not used to compute the probability densities in any way. It is only needed when a sudden background change is detected, as explained in Subsection 2.4.

The Gaussian component $G(t|\mu, \Sigma)$ is able to capture a wide range of dynamic backgrounds because $C_{\text{Back}}$ (and hence $C_{\text{Back}} + \Psi$) are not restricted to be diagonal matrices. On the other hand, the uniform component $U(t)$ is able to model foreground objects of any colour equally well.

### 2.2. Learning

When applied to the probabilistic model given in the previous subsection, the Robbins-Monro stochastic approximation algorithm yields the following update equations at time step $n$ for an observed pixel value $t_n$ (see Appendix A for details):

$$ \forall i \in \{\text{Back}, \text{Fore}\}, \pi_i(n) = (1 - \varepsilon) \pi_i(n-1) + \varepsilon R_{ni} \quad (9) $$

$$ \forall i \in \{\text{Back}, \text{Fore}\}, \mu_i(n) = (1 - \varepsilon) \mu_i(n-1) + \varepsilon R_{ni} t_n \quad (10) $$
∀i ∈ \{Back, Fore\}, µ_i(n) = \frac{m_i(n)}{π_i(n)} \tag{11}

M_{Back}(n) = (1 - ε)M_{Back}(n - 1) + εR_{n,Back}(t_n - µ_{Back}(n))(t_n - µ_{Back}(n))^T \tag{12}

C_{Back}(n) = M_{Back}(n)π_{Back}(n) \tag{13}

where ε is a constant step size, ε ≈ 0.01. The step size is constant (and not decaying) because the system is time varying, as considered in Section 3.2 of [26]. Please note that m_i and M_{Back} are auxiliary variables required to update the model parameters π_i, µ_i and C_{Back}.

2.3. Quantization noise estimation

The diagonal noise matrix Ψ models the effects of the quantization and the compression on the pixels. This is estimated separately because these effects appear and disappear suddenly from one frame to the next, so that they cannot be captured adequately by the stochastic approximation learning algorithm. Furthermore, we assume that these effects are approximately equal for all the pixels, so that an offline estimation of a global and constant value of Ψ is carried out. The matrix is given by

Ψ = \begin{pmatrix}
σ^2_R & 0 & 0 \\
0 & σ^2_G & 0 \\
0 & 0 & σ^2_B
\end{pmatrix} \tag{14}

The noise parameters may be estimated by comparing the original, uncompressed pixels τ(x) with the observed (compressed) pixels t(x) for a selection of videos:

∀i ∈ \{R, G, B\} \quad σ^2_i = E \left[ (τ_i(x) - t_i(x))^2 \right] \tag{15}

where the expectation is computed by averaging over all the pixels in the video selection. Please note that this estimation is carried out offline (that is, Ψ is a constant throughout the online learning). In practice, it is often found that the original values τ(x) cannot be obtained because the video is available in compressed format only. This problem is solved by estimating the original values by first order kernel regression of the observed values t(x). In this context the original image is regarded as a function of the position x:

τ : [1, NumRows] × [1, NumCols] → [0, v]^3 \tag{16}

τ(x) = (τ_1(x), τ_2(x), τ_3(x)) \tag{17}

Please note that τ(x) = τ(x_1, x_2) is defined even for fractional values of x_1 and x_2. We remove the quantization noise by means of the Nadaraya-Watson kernel smoother (first order kernel regression):

\hat{τ}(x) = \frac{\sum_{y ∈ W(x)} K_x(y)t(y)}{\sum_{y ∈ W(x)} K_x(y)} \tag{18}

where \hat{τ}(x) is the estimated original value at pixel position x, W(x) is a bidimensional window centered at position x, and K_x is a smoothing kernel function. The adaptation to the input data is enhanced if we choose the kernel to depend on the local gradient covariance matrix S_x [46, 47].
2.4 Sudden background change detection

\[ K_x(y) = \frac{\sqrt{\det(S_x)}}{2\pi h^2} \exp \left( -\frac{1}{h^2} (x - y)^T S_x (x - y) \right) \]  

(19)

Here \( h \) is a global smoothing parameter, and \( S_x \) is given by [48]:

\[ S_x = E_x \left[ (\nabla t)(\nabla t)^T \right] \]  

(20)

\[ E_x[\xi] = \int \int_{V(x)} \xi(y) dy \]  

(21)

where \( \nabla t \) is a \( 2 \times 3 \) Jacobian matrix and \( V(x) \) a local neighbourhood of pixel \( x \). In practice \( V(x) \) is a circle with fixed radius \( r \) centred in \( x \), and \( S_x \) is quickly estimated by Sobel or Prewitt edge detection filters applied to the luminance component of \( t \). That is, \( Vt \) is reduced to \( 2 \times 1 \) size.

The final step of this procedure is the approximation of the quantization noise by comparison of the estimated original values \( \hat{\tau}_i \) with the observed (compressed) values \( t_i \):

\[ \forall i \in \{R, G, B\} \quad \sigma_i^2 \approx E \left[ (\hat{\tau}_i(x) - t_i(x))^2 \right] \]  

(22)

where the expectation is computed by averaging over all the pixels in the video selection, as before.

2.4. Sudden background change detection

Sometimes an object integrates into the background, for example when a car is parked. This produces a sudden change in the background distributions of many pixels, i.e. those which are inside the integrating object. These changes must be dealt with separately [24]. As pointed out by Kushner & Yin in Section 3.2 of [26], stochastic approximation of time varying systems is only advisable when the rate of change of the estimated parameters is slow. Hence, these sudden changes must be tackled with a different procedure which detects these situations that the stochastic approximation framework is unable to cope with.

The detection procedure needs a previous offline study. Let \( \chi \) be the ground truth for a particular pixel:

\[ \chi = \begin{cases} 
1 & \text{iff the pixel belongs to the foreground} \\
0 & \text{iff the pixel belongs to the background} 
\end{cases} \]  

(23)

Then, let \( \tilde{\chi} \) be the estimation of \( \chi \) carried out by the basic algorithm, i.e. without sudden background change detection:

\[ \tilde{\chi} = \begin{cases} 
1 & \text{iff the pixel is classified as foreground by the basic algorithm} \\
0 & \text{iff the pixel is classified as background by the basic algorithm} 
\end{cases} \]  

(24)

The basic algorithm works correctly if and only if \( \tilde{\chi} = \chi \). We define three disjoint random events: undetected background change (Change), correct foreground detection (Good) and other situations that we are not interested in (Other):

\[ \text{Change} \equiv (\tilde{\chi} = 1) \land (\chi = 0) \]  

(25)

\[ \text{Good} \equiv (\tilde{\chi} = 1) \land (\chi = 1) \]  

(26)
2.5 Initialization

At frame $n$, the number of frames $z(n)$ that the pixel has been continuously classified as foreground is defined as follows:

$$z(n) = \max \{ N \mid \tilde{\chi}(n) = \tilde{\chi}(n-1) = \ldots = \tilde{\chi}(n-N) = 1 \}$$

(28)

Please note that $z(n)$ is readily computed by a running counter which is incremented each frame that $\tilde{\chi} = 1$ and reset to zero each time that $\tilde{\chi} = 0$. The offline task consists in estimating the probabilities $P(\text{Good} \mid z, \tilde{\chi} = 1)$ and $P(\text{Change} \mid z, \tilde{\chi} = 1)$. This is accomplished by counting the number of times that Change and Good are verified for a certain value of $z$, where we take as input samples the values of $z(n)$ for all frames $n$ and all pixels such that $\tilde{\chi} = 1$. These pixel counts are noisy functions of $z$ that are smoothed by kernel regression before estimating the probabilities.

Then we compute a threshold $Z$:

$$Z = \min \left\{ z \mid P(\text{Change} \mid z, \tilde{\chi} = 1) > \frac{1}{2} \right\}$$

(29)

The computation of $Z$ finishes the online study. When the online system is operating, in each pixel we maintain a running counter of the number of frames $z$. If we have at frame $n$ that a pixel is classified as foreground and $z(n) \geq Z$, then that pixel must be reset. The pixel reset procedure involves setting $C_{\text{Back}} = 0$. Additionally, we swap $\mu_{\text{Back}}$ with $\mu_{\text{Fore}}$, and $m_{\text{Back}}$ with $m_{\text{Fore}}$.

2.5 Initialization

Here we outline the initialization procedure to be carried out at the beginning of the execution of the method. The first $K$ frames of the analyzed video sequence are used for this purpose, with $K = 200$ in our experiments. The background model of every pixel is set up according to the $K$ pixel value samples obtained from those first frames. Hence we have:

$$\forall i \in \{ \text{Back, Fore} \} , \pi_i(0) = \frac{1}{2}$$

(30)

$$m_{\text{Back}}(0) = \frac{\pi_{\text{Back}}(0)}{K} \sum_{j=1}^{K} t_j$$

(31)

$$\mu_{\text{Back}}(0) = \frac{m_{\text{Back}}(0)}{\pi_{\text{Back}}(0)}$$

(32)

$$M_{\text{Back}}(0) = \frac{\pi_{\text{Back}}(0)}{K} \sum_{j=1}^{K} (t_j - \mu_{\text{Back}}(0))(t_j - \mu_{\text{Back}}(0))^T$$

(33)

$$C_{\text{Back}}(0) = \frac{M_{\text{Back}}(0)}{\pi_{\text{Back}}(0)}$$

(34)

On the other hand, the initialization of $\mu_{\text{Fore}}$ is not critical, since its value is not used for the first $Z$ frames at least. Hence we start with a point in the middle of the colour space:
2.6 Implementation

\[ \mathbf{m}_{\text{Fore}}(0) = \pi_{\text{Fore}}(0) \mu_{\text{Fore}}(0) \]  \hspace{1cm} (36)

Finally, the running counter is initialized with \( z(0) = 0 \).

2.6. Implementation

Since our system is designed to operate in real time, it is of paramount importance to reduce the number of required calculations as possible. In particular, the computation of the determinant and the inverse of \( \Sigma \), which are needed to obtain the responsibilities \( R_{ni} \) are the heaviest parts in terms of floating point operations. Next we outline how an optimized implementation should be built.

First of all, there is no point in using routines from standard numerical libraries, since they are tuned for high dimension \( D \). Best results are obtained by deriving specific equations for the \( D = 3 \) case, which is the only one to be considered. In addition to this, we should take advantage of the fact that \( \Sigma \) is symmetric, because it is a covariance matrix.

The computation of the determinant is carried out by imposing the symmetry constraint to the well known Sarrus’ scheme:

\[ \text{det} \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} = adf - d^2e + 2bce - e^2a - b^2f \]  \hspace{1cm} (37)

Once the determinant is known, we can compute the inverse from the cofactors. Again, we take into account the symmetry of \( \Sigma \), which implies that \( \Sigma^{-1} \) is also symmetric:

\[ \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}^{-1} = \left( \text{det} \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix} \right)^{-1} \begin{pmatrix} a' & b' & c' \\ b' & d' & e' \\ c' & e' & f' \end{pmatrix} \]  \hspace{1cm} (38)

\[ a' = df - e^2 \]  \hspace{1cm} (39)

\[ b' = ce - bf \]  \hspace{1cm} (40)

\[ c' = be - cd \]  \hspace{1cm} (41)

\[ d' = af - c^2 \]  \hspace{1cm} (42)

\[ e' = bc - ae \]  \hspace{1cm} (43)

\[ f' = ad - b^2 \]  \hspace{1cm} (44)

These equations provide a way to fulfil real time execution requirements.
3. Experimental results

In this section a principled evaluation methodology is developed in order to compare our stochastic approximation approach (AE)\(^1\) against several state-of-art video segmentation methods. The results of the segmentation were assessed both quantitatively and qualitatively using a dataset of diverse sequences with a wide range of complex backgrounds and different situations.

3.1. Methods

Considering that our approach is included among the so-called pixel-level methods, which analyse the scene at a low level pixel by pixel, the comparison is done with respect to several techniques of the same class. The Pfinder method (PF) proposed in \([15]\), the two Mixture of Gaussians approaches, GMM and AGMM, which are developed in \([16]\) and \([19]\) respectively, and the Li et al. approach \([24]\) called FGD, are also included in our performance analysis. The AGMM code is given by the author in its web page\(^2\), whereas an FGD implementation version is easily accessible in the free OpenCV Computer Vision library\(^3\).

The implementations for all of the above considered methods are based on C language MEX files for its use with Matlab, as is the case of the implementation of our method. MEX files carry out the most time consuming sections, while the rest is implemented with Matlab scripts. We have not used any GPU resources, so there is no need for special graphics hardware. All the experiments reported on this paper have been carried out on a 32-bit PC with a quad core 2.40GHz CPU and 3GB RAM.

It is important to notice that as our aim is to assess the outputs of the five compared methods as fairly as possible, no post processing technique is taken into account, due to the fact that these modules are independent of the background model and no feedback on the model is performed from their outputs.

3.2. Sequences

The comparison of the five previous algorithms is performed using a representative set of nine outdoor and indoor video sequences. Some valuable public repositories provide these datasets together with the ground truth information, which consists of a set of manually-segmented images where the sets of pixels belonging to the foreground and the background are defined. One of them was created by the International Conference VSSN’06, on the occasion of an algorithm competition in Foreground/Background Segmentation\(^4\). These datasets are considered as synthetic datasets since they are composed by motion objects generated by 3D software together with real background scenes. A notorious advantage of this kind of datasets is that the ground truth can be automatically defined for all the frames, because the foreground objects are added subsequently to the real scene. The sequences named Video2 (V2) and Video4 (V4) have been selected for our comparison.

Several sequences from the dataset developed by Li et al. \([24]\) and available in his website\(^5\) have been used in our study, because they are considered as complex scenes which contain some of the more common problems which the ideal object detection method should deal with. Thus, each one of the scenes has a wide variety of characteristics and presents a significant challenge because of its complexity. Sequences which contain moving objects as a part of the stationary background, sudden illumination changes, foreground

---

\(^1\)Further information about the source code and test videos is available in http://www.lcc.uma.es/%7Eezeqlr/backsa/backsa.html. The test videos are also available as electronic annexes to this paper.

\(^2\)http://staff.science.uva.nl/%7Ezivkovic/DOWNLOAD.html

\(^3\)http://sourceforge.net/projects/opencvlibrary.html

\(^4\)http://mmc36.informatik.uni-augsburg.de/VSSN06_OSAC

\(^5\)http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html
3.2 Sequences

Figure 1: Experimental results on scenes with high variability on the background. An outdoor campus scene (CAM) with waving tree branches corresponds to the first three rows with frames 1392, 1758, and 2348. A meeting room environment (MR) with wavering curtains because of air-conditioning is shown in the last rows with frames 2774 and 3266. From left to right, each column shows: frame, GT, AE, FGD, AGMM, GMM and PF results.

object’s pixel features which may be subsumed by the modelled background, cast shadows or foreground motionless objects appearing at the beginning of the process, are a significant sample of the problematic situations which arise in video surveillance. Several of the available segmented sequences have been included, namely: meeting room with moving curtain (MR), campus with plentiful vegetation moving continuously (CAM), water surface (WS), public fountain throwing water (FT) and moving escalators in a subway station (SS).

Another representation of the analysed sequences in the literature is obtained from CAVIAR dataset\(^6\). Only a sequence where people are walking on the corridor (OS) is included in our test data due to the difficulty with getting the ground truth. The assessment of an outdoor level crossing scene (LC) is also carried out.

In order to check the stability of our approach, we have added three new video sequences which have several thousand frames each. The BUS sequence shows a busy bus stop during the morning whereas the IS sequence is an example of a highway intersection\(^7\). Another traffic long scene (US) provided by NGSIM\(^8\) is used in our evaluation. Unfortunately, the ground truth is not available for these huge sequences. Because of this, the experimental results on these long scenes are only qualitative.

\(^6\)http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

\(^7\)http://limu.aist.kyushu-u.ac.jp/en/dataset/

\(^8\)http://ngsim-community.org/
3.3 Qualitative Results

The performance results of the five studied methods are observed in the figures 1, 2 and 3. Each row shows an comparison of the segmentation results for each method from a single frame. All the rows correspond to some frames from the selected datasets. The first column displays some frames from the different benchmark sequences whereas in the second column the manually-segmented foreground mask (GroundTruth) is depicted. From the third column to the seventh one, the segmentation results of the following methods are showed: our stochastic approximation method (AE), Li et al. approach (FGD), adaptive mixture of Gaussians (AGMM), standard mixture of Gaussians (GMM) and single Gaussian distribution (PF) respectively. Several consequences can be extracted from the images. In most cases, our approach (AE) achieves better results than the other alternatives, both indoor and outdoor scenes. Particularly, the amount of noise in the outdoor ones is drastically reduced in the AE and FGD methods in comparison to the rest of statistical approaches. The negative effect known as waving trees in this kind of sequences causes a continuous variation of the pixel tonality on the background, requiring models which can represent disjoint sets of these pixel values. This is rather significant in CAM (the first three rows in the figure 1) where the noise is so intense that the motion objects are difficult to separate. A similar situation happens in LC frames (figure 3; first, second and third row) although is even worse for the PF method, whose foreground results are unreliable and not suitable to find the motion objects exactly. Other examples of the noise problem for AGMM, GMM and PF are observed in the sequences WS (the last two rows in the figure 2) and V4 (the last rows in figure 3).

On the other hand, repetitive movements of the stationary background objects keep appearing in the evaluated indoor scenes, either caused by escalators in the SS sequence (the first row in figure 2) or by curtains in motion because of the air conditioning in a meeting room in the MR sequence (fourth and fifth rows in figure 1). It can be seen that only our approach has correctly handled this situation, by removing almost totally both the escalators and the curtains from the foreground mask in the two sequences. Furthermore, the problem
3.3 Qualitative Results

![Figure 3: Experimental results on level crossing environments (LC), which are in the first three rows with frames 324, 430 and 465, scenes from corridors in leisure centres (OC), the fourth row with frame 390, and synthetic sequences which model 3D objects over real backgrounds (V2 and V4), corresponding to the last four lines, with frames 332, 611, 748 and 804 respectively.](image)

- (a) Frame
- (b) GT
- (c) AE
- (d) FGD
- (e) AGMM
- (f) GMM
- (g) PF

of camouflage, in which there is a significant similarity between the pixel colours from the background and foreground, is also presented in MR sequence (last row, figure 1). The appearance of holes in the foreground mask is a clear example of it, and its solution should be reached by analysing globally each object in the post processing phase. With regard to the cast shadows, our approach together with the statistical ones (AGMM, GMM and PF) are more sensitive to them than the FGD method, incorporating these pixels into the foreground. This can be noticed in the OS sequence, where the people’s reflected image on the corridor floor is considered as foreground (fourth row in figure 3). As we consider this as a problem to be dealt with in a subsequent post processing phase, we do not attempt to solve it in the experiments, in spite of the fact that it affects the assessment of our method negatively. In stationary scenes like V2 (fifth and sixth row in figure 3), there is not much difference between the segmentation quality in all the assessed techniques as we can observe. However, a non desirable effect is produced after applying the FGD method in sequences with quick movements on the foreground, which leaves a trace of the motion objects in previous frames. This is particularly evident in sequences MR, V2 and V4.
3.4 Performance Measures

Many algorithms have been proposed for objective evaluation of motion detection methods. Since successful tracking relies heavily on accurate object detection, the evaluation of object detection algorithms within a surveillance system plays an important role in overall performance analysis of the whole system. Generally, the selection of a set of evaluation metrics depends largely on the kind of applied segmentation. Provided that the ground truth is supplied, evaluation techniques based on comparison with that ideal output can be further classified according to the type of metrics they propose. Typically, pixel-based metrics [49, 50, 51] consist of a combination of true positives \( tp \), false positives \( fp \), true negatives \( tn \) and false negatives \( fn \), computed over each frame and whose sum corresponds to the image size. It should be noted that \( fp \) and \( fn \) refer to pixels misclassified as foreground \( fp \) or background \( fn \), while \( tp \) and \( tn \) account for accurately classified pixels.

Other alternatives which use object-based metrics are found in the literature [52]. They analyse features as false alarms, detection failures, or merge and split regions but always from the perspective of an object as a whole, and not as a sum of a set of pixels. We consider this approach very interesting after applying post-processing techniques, in which the output is a set of clean blobs, far from noise and spurious objects. Additionally, these measures can help to select the best post-processed configuration in order to enhance the segmentation results [20]. However, and as we said before, the post-processing phase and its evaluation are out of the scope of this paper, so we use a subset of the pixel-based metrics as our performance measures.

In this paper a similarity standard measure based on the previous pixel-based metrics is applied. It was defined as a metric to check the MPEG Segmentation Quality [53] and used as the starting point of other developed metrics [54]. It has also been used to compare different object detection methods [24, 25]. This quantitative measure named spatial accuracy is defined as follows:

\[
AC = \frac{\text{card}(A \cap B)}{\text{card}(A \cup B)} \tag{45}
\]

where ‘card’ stands for the number of elements of a set, \( A \) is the set of all pixels which belong to the foreground, and \( B \) is the set of all pixels which are classified as foreground by the analyzed method:

\[
A = \{ x \mid \chi(x) = 1 \} \quad B = \{ x \mid \tilde{\chi}(x) = 1 \} \tag{46}
\]

In order to take into account some information about false negatives and false positives, the proportions of these metrics are also defined and normalised using the foreground mask:

\[
FN = \frac{\text{card}(A \cap \bar{B})}{\text{card}(A \cup B)} \quad FP = \frac{\text{card}(ar{A} \cap B)}{\text{card}(A \cup B)} \tag{47}
\]

From set theory we know that:

\[
AC, FN, FP \in [0, 1] \quad AC + FN + FP = 1 \tag{48}
\]

The optimal performance would be achieved for \( AC = 1, FN = 0, FP = 0 \). On the other hand, the worst possible performance corresponds to \( AC = 0, FN + FP = 1 \). Additionally, we utilise two information retrieval measurements, recall and precision, to quantify how well each algorithm matches the ground-truth, by taking into account both the background and the foreground. Like the previous accuracy measure, the new ones have also been rather applied in vision for comparing the performance of motion detection algorithms. Both are considered as a standard within this scope. Therefore, the precision \( PR \) and recall \( RC \) measures can be computed as follows (higher is better):
Table 1: Parameter settings tested during evaluation of the object detection algorithms. The combination of all the values of the parameters generate a set of configurations for each method.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Test Parameters</th>
</tr>
</thead>
</table>
| PF [15]   | Learning rate, $\alpha = \{1e^{-4}, 5e^{-4}, 0.001, 0.005, 0.01, 0.05, 0.1\}$  
Threshold for Gaussian tiles in standard deviations, $N = \{2, 2.5, 3\}$ |
| GMM [16], AGMM [19] | Learning rate, $\alpha = \{1e^{-4}, 5e^{-4}, 0.001, 0.005, 0.01, 0.05, 0.1\}$  
No. of Gaussian mixture components, $K = \{3, 5, 7\}$  
Threshold for Gaussian tiles in standard deviations, $N = \{2, 2.5, 3\}$  
Weight threshold, $T = \{0.6, 0.8\}$ |
| FGD [24]  | Learning rate, $\alpha_1 = \{0.01, 0.05, 0.1\}$  
$\alpha_2 = \{1e^{-4}, 5e^{-4}, 0.001, 0.005, 0.01, 0.05, 0.1\}$  
$\alpha_3 = \{0.01, 0.05, 0.1\}$  
Similarity threshold, $\delta = \{2, 2.5, 3\}$ |
| AE        | Step size, $\epsilon_0 = \{1e^{-4}, 5e^{-4}, 0.001, 0.005, 0.01, 0.05, 0.1\}$ |

\[
PR = \frac{\text{card}(A \cap B)}{\text{card}(B)} \quad RC = \frac{\text{card}(A \cap B)}{\text{card}(A)} \quad PR, RC \in [0, 1] \quad (49)
\]

Figure 4: Experimental results in three long sequences: BusStop (BUS) showing the frame 2780, Intersection (IS), frame 2090, and US Highway 101 (US) with frame 7125. From left to right, each column shows: frame, AE, FGD, AGMM, GMM and PF results.

$RC$ is defined as the percentage of the correctly detected pixels to the real moving objects whereas $PR$ is defined as the percentage of the detected pixels to all detected moving object pixels. High recall means less segmentation misses while high precision means less false segmentation. Often there is a trade-off between precision and recall, i.e. we can make one of them grow as desired at the expense of diminishing the other.

Finally, another standard measure that combines precision and recall, the traditional F-measure or balanced F-score, is computed:
3.5 Parameter Selection

One of the major drawbacks of motion detection algorithms is to find the correct value of the defined parameters. Many times it is rather complicated to tune them for a particular analysed scene. Although our model defines several parameters, we have empirically found that a fixed value for some of them is suitable for all the studied scenes. Specifically, the number of frames used in the initialisation phase (Section 2.5) is always assigned to $K = 200$, and the global smoothing parameter (Section 2.3) is held fixed to $h = 2$. Therefore, in our model only the step size parameter $\varepsilon_0$, which influences the learning process, must be tuned for each scene.

Accordingly, for each parameter of the studied methods a valid set of values is considered. Then multiple configurations of the methods are generated by combining the values of the parameters in every possible way. The considered parameter values for each method are shown in Table 1. The figure 5 shows the result of all the generated configurations for each method in each one of the nine sequences after applying the previously defined evaluation metrics. The horizontal axis corresponds to the average percentage of $FP$ on the scene, while the vertical axis is associated with $FN$ values. Each point of the plot represents the average $FP$ and $FN$ of a generated configuration when applied to that sequence. The closer the points are to the origin, the better the segmentation has been carried out. The optimum performance occurs if $FN = 0$ and $FP = 0$, which implies there is a perfect match between the output of the algorithm and the ground Truth ($AC = 1$), according to the relation between the three computed metrics. The results are always below the diagonal of the plot because we always have $FN+FP \leq 1$, as seen in equation 48. Those configurations which are closer to the diagonal correspond to obtained segmentations with poor quality. As shown in the accuracy and F-measure table (2 and 3 respectively), our approach ($AE$) gets the best results for six of the nine analysed sequences ($CAM$, $SS$, $LC$, $V2$, $V4$ and $WS$), while for two of the remaining ones our performance is quite competitive with respect to the other methods ($MR$ and $OS$ sequences). The results of the best performing configuration are shown for each sequence in the table 5, including three different quality measures and the parameter setting for each method.

Next we discuss the robustness of the methods with respect to the choice of their parameters. Many times a wrong choice of the parameters of the applied segmentation algorithm generates a deficient segmentation, where it is not possible to distinguish moving objects at first sight. Sometimes, it is not easy to determine in advance the appropriate parameters to get a good quality segmentation, which requires a tedious empirical assessment of the scene to assign these values. The Figure 5 shows the variability in the evaluation of each algorithm. This variability depends largely on the analysed sequence, but the robustness of the method has also an influence, i.e. more robust methods yield smaller values. If the configuration cloud is compact, it means that the results do not vary significantly after a change in its parameters. On the other hand, if several configurations are far from each other, it implies that the variation of a parameter causes abrupt changes in the results, which is a very undesirable property for a motion detection algorithm. As shown in Figure 5, the compactness for $AGMM$, $GMM$ and $PF$ methods is rather poor in most of the sequences, while our approach and $FGD$ model have their configurations very close together. In other words, the performance of the proposed method is not very sensitive to the parameter selection. For instance, in sequences as $LC$ and $V2$ our results are fairly clustered, while at the same time they provide the best value of accuracy. The experimental qualitative results on several long scenes (Figure 4) further demonstrate the robustness of our proposal.
### 3.6 Quantitative Results

The previous figures have provided a qualitative assessment of the selected methods’ performance. In order to show more clearly the improvements achieved by our AE approach, the corresponding quantitative evaluations are listed in table 5, where a comparison is provided for each sequence using a set of performance

<table>
<thead>
<tr>
<th>Sequence</th>
<th>FP</th>
<th>FN</th>
<th>WaterSurface</th>
<th>Video4</th>
<th>Video2</th>
<th>LevelCrossing</th>
<th>OneShopOneWait1cor</th>
<th>Fountain</th>
<th>Escalator</th>
<th>Curtain</th>
<th>Campus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: False Positives (FP) and False Negatives (FN) ratios after applying each method to the test sequences with all the parameter configurations. Each coloured point '*' is considered as a different configuration of that method. The closer the points are to the origin, the better the segmentation is. Additionally, the method is less sensible to a parameters’ change if the cloud of points keeps compact and grouped.
3.6 Quantitative Results

Table 2: Quantitative evaluation using the accuracy measure (equation 45) and comparison results over the test sequences. The results show the mean and standard deviation after analysing each dataset, where the best performance for each sequence is highlighted in bold. Our approach is the best method in six of nine sequences.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>PF</th>
<th>GMM</th>
<th>AGMM</th>
<th>FGD</th>
<th>AE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Campus (CAM)</td>
<td>0.22±0.12</td>
<td>0.27±0.13</td>
<td>0.26±0.13</td>
<td>0.50±0.22</td>
<td><strong>0.60±0.17</strong></td>
</tr>
<tr>
<td>Meeting Room (MR)</td>
<td>0.66±0.07</td>
<td>0.71±0.08</td>
<td>0.71±0.09</td>
<td><strong>0.83±0.05</strong></td>
<td>0.78±0.05</td>
</tr>
<tr>
<td>Subway Station (SS)</td>
<td>0.31±0.10</td>
<td>0.41±0.09</td>
<td>0.40±0.09</td>
<td>0.28±0.12</td>
<td><strong>0.46±0.12</strong></td>
</tr>
<tr>
<td>Fountain (FT)</td>
<td>0.56±0.08</td>
<td><strong>0.60±0.08</strong></td>
<td>0.59±0.08</td>
<td>0.46±0.16</td>
<td>0.43±0.17</td>
</tr>
<tr>
<td>Level Crossing (LC)</td>
<td>0.53±0.15</td>
<td>0.77±0.06</td>
<td>0.75±0.08</td>
<td>0.78±0.05</td>
<td><strong>0.88±0.03</strong></td>
</tr>
<tr>
<td>Corridor (OC)</td>
<td>0.71±0.03</td>
<td>0.70±0.04</td>
<td>0.70±0.05</td>
<td><strong>0.75±0.02</strong></td>
<td>0.72±0.04</td>
</tr>
<tr>
<td>Video2 (V2)</td>
<td>0.86±0.03</td>
<td>0.91±0.02</td>
<td>0.91±0.02</td>
<td>0.76±0.06</td>
<td><strong>0.92±0.02</strong></td>
</tr>
<tr>
<td>Video4 (V4)</td>
<td>0.38±0.07</td>
<td>0.49±0.07</td>
<td>0.49±0.07</td>
<td>0.66±0.09</td>
<td><strong>0.68±0.11</strong></td>
</tr>
<tr>
<td>WaterSurface (WS)</td>
<td>0.77±0.03</td>
<td>0.80±0.03</td>
<td>0.80±0.03</td>
<td>0.78±0.06</td>
<td><strong>0.87±0.02</strong></td>
</tr>
</tbody>
</table>

Table 3: Another quantitative evaluation using the F-measure, given by a combination of the precision and recall values and defined in the equation 50. The results show the mean and standard deviation after analysing each dataset, where the best performance for each sequence is highlighted in bold.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>PF</th>
<th>GMM</th>
<th>AGMM</th>
<th>FGD</th>
<th>AE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Campus (CAM)</td>
<td>0.34±0.16</td>
<td>0.40±0.17</td>
<td>0.40±0.17</td>
<td>0.64±0.23</td>
<td><strong>0.74±0.15</strong></td>
</tr>
<tr>
<td>Meeting Room (MR)</td>
<td>0.79±0.05</td>
<td>0.83±0.06</td>
<td>0.83±0.06</td>
<td><strong>0.90±0.03</strong></td>
<td>0.88±0.03</td>
</tr>
<tr>
<td>Subway Station (SS)</td>
<td>0.47±0.12</td>
<td>0.58±0.09</td>
<td>0.56±0.10</td>
<td>0.42±0.15</td>
<td><strong>0.62±0.12</strong></td>
</tr>
<tr>
<td>Fountain (FT)</td>
<td>0.72±0.07</td>
<td><strong>0.74±0.07</strong></td>
<td>0.74±0.07</td>
<td>0.62±0.16</td>
<td>0.58±0.17</td>
</tr>
<tr>
<td>Level Crossing (LC)</td>
<td>0.69±0.10</td>
<td>0.87±0.04</td>
<td>0.85±0.05</td>
<td>0.87±0.03</td>
<td><strong>0.94±0.02</strong></td>
</tr>
<tr>
<td>Corridor (OC)</td>
<td>0.83±0.02</td>
<td>0.82±0.03</td>
<td>0.82±0.03</td>
<td><strong>0.86±0.02</strong></td>
<td>0.83±0.03</td>
</tr>
<tr>
<td>Video2 (V2)</td>
<td>0.93±0.02</td>
<td>0.95±0.01</td>
<td>0.95±0.01</td>
<td>0.86±0.04</td>
<td><strong>0.96±0.01</strong></td>
</tr>
<tr>
<td>Video4 (V4)</td>
<td>0.55±0.07</td>
<td>0.65±0.06</td>
<td>0.65±0.06</td>
<td>0.79±0.07</td>
<td><strong>0.80±0.08</strong></td>
</tr>
<tr>
<td>WaterSurface (WS)</td>
<td>0.87±0.02</td>
<td>0.89±0.02</td>
<td>0.89±0.02</td>
<td>0.88±0.04</td>
<td><strong>0.93±0.01</strong></td>
</tr>
</tbody>
</table>

evaluation measures.

The tables 2 and 3 display a comparison among all the methods for each sequence using the performance measures, AC and F-measure respectively. As it can be noted, our method AE achieves the best results for six of the nine test sequences. We must say that the AC performance measure is faithful as long as there are foreground objects in the evaluated frame. Otherwise, the most probable configuration is \( FP = 1, FN = 0 \), AC = 0 because of the more than likely occurrence of false positives after the detection. Therefore, this configuration does not truly reflect the effectiveness of the methods since it is only based on the detected foreground and not in the background. Moreover, in outdoor scenes with great background variability, and where moving objects are quite small compared to the image size, the results of the assessment will significantly decrease due to this foreground importance. To handle this problem, only frames containing foreground objects are evaluated with the aim of not affecting the overall accuracy measure adversely.

The accuracy, precision and recall results after the evaluation are represented in several figures and tables. In the figure 6, three measure graphs (Precision, Recall and, Accuracy, respectively) are plotted, in which the signal of each measure is shown along the time for each analysed method. We must highlight...
3.6 Quantitative Results

Figure 6: Precision, recall and accuracy measures, which are shown frame by frame over the sequences LC, V2 and V4. It is remarkable the stability and low variability of our AE approach.

The improvement of our approach (AE) in the first row of the figure 6 (LC sequence), where it gets higher values both in recall and accuracy measures. Moreover, the three obtained signals for the method AE in the sequence V2 (second row of the same figure) are much more stable than those corresponding to the other techniques like FGD. The results for the sequence V4 are quite similar between AE and FGD methods, although the signals for the first one are less abrupt and slightly higher than for the last one. The quantitative results from the three sequences represented in the figure 6 are shown (formatted as mean±std) in the table 5.

In table 4 we show the results of the time complexity analysis for each method. From the third column on, each cell shows the average rate of Frames Per Second (FPS) which each technique achieves when analysing each sequence (higher is better). It must be noted that the frame size influences the computational complexity of the algorithms. As said before, the processing time is measured on a 32-bit PC with a quad core 2.40 GHz CPU and 3GB RAM. As seen, our approach outperforms all its competitors in every tested video sequence.
Table 4: Frames per second (FPS, higher is better) for each analysed sequence together with its frame size (first and second columns). The bigger the frame size, the fewer FPS ratio. Real time requirements are fulfilled if a method is over 15 fps. Each value is obtained by using the parameter configuration with the best background detection accuracy for each method and sequence.

<table>
<thead>
<tr>
<th>Size</th>
<th>AGMM</th>
<th>FGD</th>
<th>PF</th>
<th>GMM</th>
<th>AE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BusStop (BUS)</td>
<td>320x240</td>
<td>99.10±0.06</td>
<td>21.16±0.01</td>
<td>115.48±0.13</td>
<td>32.40±0.01</td>
</tr>
<tr>
<td>Campus (CAM)</td>
<td>160x128</td>
<td>338.80±0.60</td>
<td>64.38±0.51</td>
<td>413.81±0.36</td>
<td>134.30±0.11</td>
</tr>
<tr>
<td>Meeting Room (MR)</td>
<td>160x128</td>
<td>340.59±0.87</td>
<td>84.94±0.09</td>
<td>431.94±0.38</td>
<td>158.05±0.12</td>
</tr>
<tr>
<td>Subway Station (SS)</td>
<td>160x130</td>
<td>311.71±0.26</td>
<td>60.33±0.06</td>
<td>415.96±0.31</td>
<td>122.28±0.11</td>
</tr>
<tr>
<td>Fountain (FT)</td>
<td>160x128</td>
<td>363.85±1.40</td>
<td>89.83±0.17</td>
<td>427.95±0.58</td>
<td>154.38±0.19</td>
</tr>
<tr>
<td>Intersection (IS)</td>
<td>320x240</td>
<td>102.40±0.06</td>
<td>24.11±0.02</td>
<td>116.19±0.15</td>
<td>32.56±0.02</td>
</tr>
<tr>
<td>Level Crossing (LC)</td>
<td>360x240</td>
<td>74.51±0.03</td>
<td>20.59±0.01</td>
<td>101.09±0.32</td>
<td>34.67±0.03</td>
</tr>
<tr>
<td>Corridor (OC)</td>
<td>384x288</td>
<td>67.76±0.04</td>
<td>14.24±0.00</td>
<td>80.25±0.06</td>
<td>32.16±0.04</td>
</tr>
<tr>
<td>Video2 (V2)</td>
<td>384x240</td>
<td>84.93±0.11</td>
<td>27.61±0.01</td>
<td>96.96±0.20</td>
<td>45.84±0.03</td>
</tr>
<tr>
<td>Video4 (V4)</td>
<td>384x240</td>
<td>76.31±0.05</td>
<td>18.00±0.00</td>
<td>94.38±0.12</td>
<td>34.96±0.03</td>
</tr>
<tr>
<td>WaterSurface (WS)</td>
<td>160x128</td>
<td>376.46±0.79</td>
<td>86.35±0.17</td>
<td>430.72±0.61</td>
<td>188.06±0.22</td>
</tr>
<tr>
<td>US Highway 101 (US)</td>
<td>640x480</td>
<td>23.86±0.16</td>
<td>6.37±0.00</td>
<td>28.57±0.04</td>
<td>9.94±0.01</td>
</tr>
</tbody>
</table>

4. Discussion

The method we have just described departs in some fundamental ways from those commonly found in literature. These differences can be summarized as follows:

1. The fact that our method always uses full covariance matrices and Mahalanobis distances implies that it provides some equivariance features with respect to affine transformations on the input data [55, 56, 57] that proposals based on diagonal covariance matrices [16] do not have. This means that our method is more independent from the choice of the RGB, Y'CbCr, Y'PbPr, or Photo YCC colour spaces, since all of them are obtained from each other by affine transformations [58]. That is, the background detection results of our method are less dependent on the used colour space, as we can observe in the table 6, in which the accuracy of each method after analysing the Meeting Room sequence (MR) in different colour spaces is observed. In this example the difference between the better and worse accuracy measures is found in the last column. The lower difference value, the more invariant the method to the color space. As seen, our method is more stable with respect to the used color space than all the others. It should be noted that the invariance of a method with respect to the color space is relevant only if the method is good enough in terms of efficiency. In the table, this implies that the stability of the PF method is useless, since it yields the worst background detection accuracy results. Nevertheless, the equivariance of our method can not be complete, since the colour channels are encoded separately in the video files, so that the quantization errors do not transform in an equivariant way.

2. The stochastic approximation framework provides a formal justification of the learning algorithm. This is not the case for some learning rules which are heuristic modifications of the Expectation-Maximization (EM) method. These modifications are aimed to obtain an online scheme. However, it must be remarked that the EM method is originally designed for batch learning. Hence, any online version of EM should have a clear formalization, if the desirable properties of the original method are to be preserved [59].
Table 5: Best configuration for each method over all the analysed sequences. The sequence names are shown in the first column whereas the second column one indicates the applied methods. The accuracy, precision and recall measures (third, fourth and fifth column) are listed for each method in the second column. The test parameter configuration is in the last column. The row highlights in bold correspond to the best option for that sequence between all the alternatives.

<table>
<thead>
<tr>
<th>Sequence (CAM)</th>
<th>Method</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PF</td>
<td>0.22±0.12</td>
<td>0.27±0.17</td>
<td>0.61±0.12</td>
<td>0.5, α=0.1</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>0.27±0.13</td>
<td>0.35±0.19</td>
<td>0.62±0.13</td>
<td>0.5, α=0.05, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>AGMM</td>
<td>0.26±0.13</td>
<td>0.33±0.19</td>
<td>0.63±0.12</td>
<td>0.5, α=0.05, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>FGD</td>
<td>0.5±0.22</td>
<td>0.58±0.27</td>
<td>0.83±0.08</td>
<td>α=0.05, α=0.01, α=0.01, δ=3</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td><strong>0.60±0.17</strong></td>
<td>0.67±0.19</td>
<td>0.88±0.06</td>
<td>α=0.01</td>
</tr>
<tr>
<td>Meeting Room (MR)</td>
<td>PF</td>
<td>0.66±0.07</td>
<td>0.76±0.05</td>
<td>0.83±0.07</td>
<td>0.5, α=0.0005</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>0.71±0.08</td>
<td>0.95±0.04</td>
<td>0.74±0.08</td>
<td>0.5, α=0.0005, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>AGMM</td>
<td>0.71±0.09</td>
<td>0.95±0.04</td>
<td>0.74±0.08</td>
<td>0.5, α=0.0005, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>FGD</td>
<td><strong>0.83±0.05</strong></td>
<td>0.94±0.04</td>
<td>0.87±0.05</td>
<td>α=0.01, α=0.0001, α=0.05, δ=2.5</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td>0.78±0.05</td>
<td>0.95±0.03</td>
<td>0.82±0.06</td>
<td>α=0.0005</td>
</tr>
<tr>
<td>Subway Station (SS)</td>
<td>PF</td>
<td>0.31±0.10</td>
<td>0.37±0.13</td>
<td>0.71±0.11</td>
<td>0.5, α=0.005</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>0.41±0.09</td>
<td>0.52±0.12</td>
<td>0.68±0.10</td>
<td>0.5, α=0.001, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>AGMM</td>
<td>0.40±0.09</td>
<td>0.49±0.12</td>
<td>0.68±0.10</td>
<td>0.5, α=0.001, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>FGD</td>
<td>0.28±0.12</td>
<td>0.32±0.17</td>
<td>0.74±0.10</td>
<td>α=0.05, α=0.01, α=0.05, δ=2</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td><strong>0.46±0.12</strong></td>
<td>0.71±0.17</td>
<td>0.59±0.16</td>
<td>α=0.005</td>
</tr>
<tr>
<td>Fountain (FF)</td>
<td>PF</td>
<td>0.56±0.08</td>
<td>0.82±0.08</td>
<td>0.64±0.08</td>
<td>0.5, α=0.01</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td><strong>0.60±0.08</strong></td>
<td>0.77±0.07</td>
<td>0.72±0.08</td>
<td>0.5, α=0.01, 0.5, δ=0.6</td>
</tr>
<tr>
<td></td>
<td>AGMM</td>
<td>0.59±0.08</td>
<td>0.76±0.08</td>
<td>0.72±0.08</td>
<td>0.5, α=0.01, 0.5, δ=0.6</td>
</tr>
<tr>
<td></td>
<td>FGD</td>
<td>0.46±0.16</td>
<td>0.54±0.20</td>
<td>0.78±0.13</td>
<td>α=0.01, α=0.05, α=0.05, δ=3</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td>0.43±0.17</td>
<td>0.61±0.26</td>
<td>0.58±0.11</td>
<td>α=0.01</td>
</tr>
<tr>
<td>Level Crossing (LC)</td>
<td>PF</td>
<td>0.53±0.15</td>
<td>0.67±0.22</td>
<td>0.76±0.06</td>
<td>0.5, α=0.005</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>0.77±0.06</td>
<td>0.86±0.08</td>
<td>0.88±0.02</td>
<td>0.5, α=0.005, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>AGMM</td>
<td>0.75±0.08</td>
<td>0.84±0.09</td>
<td>0.88±0.02</td>
<td>0.5, α=0.005, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>FGD</td>
<td>0.78±0.05</td>
<td>0.92±0.05</td>
<td>0.84±0.04</td>
<td>α=0.01, α=0.005, α=0.05, δ=2.5</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td><strong>0.88±0.03</strong></td>
<td>0.91±0.03</td>
<td>0.96±0.01</td>
<td>α=0.01</td>
</tr>
<tr>
<td>Corridor (OC)</td>
<td>PF</td>
<td>0.71±0.03</td>
<td>0.86±0.02</td>
<td>0.80±0.03</td>
<td>0.5, α=0.005</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>0.70±0.04</td>
<td>0.79±0.03</td>
<td>0.86±0.04</td>
<td>0.5, α=0.005, 0.5, δ=0.6</td>
</tr>
<tr>
<td></td>
<td>AGMM</td>
<td>0.70±0.05</td>
<td>0.78±0.03</td>
<td>0.87±0.04</td>
<td>0.5, α=0.005, 0.5, δ=0.6</td>
</tr>
<tr>
<td></td>
<td>FGD</td>
<td><strong>0.75±0.02</strong></td>
<td>0.91±0.01</td>
<td>0.81±0.03</td>
<td>α=0.01, α=0.001, α=0.01, δ=2.5</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td>0.72±0.04</td>
<td>0.76±0.02</td>
<td>0.92±0.04</td>
<td>α=0.01</td>
</tr>
<tr>
<td>Video2 (V2)</td>
<td>PF</td>
<td>0.87±0.03</td>
<td>0.93±0.03</td>
<td>0.93±0.03</td>
<td>0.3, α=0.001</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>0.91±0.02</td>
<td>0.94±0.02</td>
<td>0.96±0.03</td>
<td>0.5, α=0.005, 0.5, δ=0.6</td>
</tr>
<tr>
<td></td>
<td>AGMM</td>
<td>0.91±0.02</td>
<td>0.94±0.02</td>
<td>0.96±0.03</td>
<td>0.5, α=0.005, 0.5, δ=0.6</td>
</tr>
<tr>
<td></td>
<td>FGD</td>
<td>0.76±0.06</td>
<td>0.84±0.07</td>
<td>0.89±0.05</td>
<td>α=0.01, α=0.005, α=0.01, δ=2</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td><strong>0.92±0.02</strong></td>
<td>0.94±0.01</td>
<td>0.98±0.02</td>
<td>α=0.0001</td>
</tr>
<tr>
<td>Video4 (V4)</td>
<td>PF</td>
<td>0.38±0.07</td>
<td>0.51±0.11</td>
<td>0.61±0.06</td>
<td>0.5, α=0.05</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>0.49±0.07</td>
<td>0.63±0.10</td>
<td>0.70±0.06</td>
<td>0.5, α=0.005, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>AGMM</td>
<td>0.49±0.07</td>
<td>0.61±0.10</td>
<td>0.72±0.05</td>
<td>0.5, α=0.005, 0.5, δ=0.8</td>
</tr>
<tr>
<td></td>
<td>FGD</td>
<td>0.66±0.09</td>
<td>0.74±0.11</td>
<td>0.86±0.04</td>
<td>α=0.01, α=0.0001, α=0.01, δ=3</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td><strong>0.68±0.11</strong></td>
<td>0.75±0.13</td>
<td>0.88±0.03</td>
<td>α=0.005</td>
</tr>
<tr>
<td>WaterSurface (WS)</td>
<td>PF</td>
<td>0.77±0.03</td>
<td>0.93±0.03</td>
<td>0.82±0.04</td>
<td>0.5, α=0.001</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>0.80±0.03</td>
<td>0.95±0.02</td>
<td>0.84±0.03</td>
<td>0.5, α=0.005, 0.5, δ=0.6</td>
</tr>
<tr>
<td></td>
<td>AGMM</td>
<td>0.80±0.03</td>
<td>0.95±0.02</td>
<td>0.84±0.03</td>
<td>0.5, α=0.005, 0.5, δ=0.6</td>
</tr>
<tr>
<td></td>
<td>FGD</td>
<td>0.78±0.06</td>
<td>0.95±0.04</td>
<td>0.82±0.05</td>
<td>α=0.01, α=0.01, α=0.1, δ=2</td>
</tr>
<tr>
<td></td>
<td>AE</td>
<td><strong>0.87±0.02</strong></td>
<td>0.95±0.01</td>
<td>0.91±0.02</td>
<td>α=0.0001</td>
</tr>
</tbody>
</table>
Table 6: Comparative performance with respect to the color space. The selected sequence is Meeting Room (MR).

<table>
<thead>
<tr>
<th></th>
<th>PhotoYCC</th>
<th>RGB</th>
<th>YCbCr</th>
<th>YPbPr</th>
<th>BestAcc − WorseAcc</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF</td>
<td>0.56±0.13</td>
<td>0.66±0.07</td>
<td>0.61±0.11</td>
<td>0.64±0.01</td>
<td>0.10</td>
</tr>
<tr>
<td>GMM</td>
<td>0.57±0.10</td>
<td>0.71±0.08</td>
<td>0.61±0.09</td>
<td>0.66±0.09</td>
<td>0.14</td>
</tr>
<tr>
<td>AGMM</td>
<td>0.57±0.10</td>
<td>0.71±0.09</td>
<td>0.61±0.09</td>
<td>0.66±0.09</td>
<td>0.14</td>
</tr>
<tr>
<td>FGD</td>
<td>0.57±0.11</td>
<td>0.83±0.05</td>
<td>0.63±0.09</td>
<td>0.63±0.08</td>
<td>0.26</td>
</tr>
<tr>
<td>AE</td>
<td>0.74±0.07</td>
<td>0.78±0.05</td>
<td>0.75±0.07</td>
<td>0.83±0.06</td>
<td>0.09</td>
</tr>
</tbody>
</table>

3. Our method models the foreground by means of a uniform distribution. This is of paramount importance in order to cope with objects which do not follow any colour pattern. Every object with a previously unseen colour is modelled by the foreground mixture component just as well as any other object. Previous proposals rely on mixtures of Gaussian distributions for the foreground [60], so that new colours are poorly modelled. In order to illustrate this crucial point, the figure 7 shows a comparison between our model with one Gaussian and one Uniform mixture components, and the same model with two Gaussian mixture components. This way, the advantages of the first version are revealed, since the ‘Two Gaussians’ approach misses completely the foreground pixels which are far from the peak, which leads to background detection failures. In addition to this, the table 7 compares the accuracy values of the two algorithm versions for the sequences CAM and MR. It is clear from the table that the results for the model with one Gaussian and one uniform distribution are quantitatively better than the other approach. Please note that the table also shows that our quantization error estimator provides significant performance improvements.

4. Since we only use two mixture components (a Gaussian and a uniform), the computational complexity of the method is reduced with respect to those with multiple components [19] (see the table 4). Moreover, there is no need for an additional decision procedure [16] to determine which component should become part of the background when a sudden background change is detected. Consequently, the possible decision errors in that procedure are avoided.

5. The comparison between different segmentation methods has concluded that our approach (AE) has been the best one in terms of efficiency in six of the nine sequences, with the additional advantage of being the second simplest applied method (after PF). Another outstanding feature is its robustness and stability, with little variability in the segmentation results between consecutive frames, so that it features a stable level of efficiency over the time.

6. Unlike the rest of the comparative methods in which they have to adjust between two (PF) and four parameters (GMM AGMM and FGD), only one should be adjusted in this model, which involves simplifying the fine-tuning parameter setup and avoiding some errors because of a bad parameter ad-

Table 7: Quantitative evaluation using the accuracy measure which shows the improvements of including the quantization error estimation (Ψ) and using one Gaussian and one Uniform distributions in the model.

<table>
<thead>
<tr>
<th></th>
<th>Without Ψ</th>
<th>With Ψ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Campus (CAM)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gauss+Unif</td>
<td>0.57±0.18</td>
<td><strong>0.60±0.17</strong></td>
</tr>
<tr>
<td>Gauss+Gauss</td>
<td>0.48±0.18</td>
<td>0.52±0.17</td>
</tr>
<tr>
<td>Curtain (MR)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gauss+Unif</td>
<td>0.71±0.14</td>
<td><strong>0.78±0.05</strong></td>
</tr>
<tr>
<td>Gauss+Gauss</td>
<td>0.69±0.11</td>
<td>0.76±0.06</td>
</tr>
</tbody>
</table>
justment. In fact, the initial choice of parameter values is a crucial issue for the suitable performance of an object detection algorithm. This choice is clearly critical in AGMM, GMM and PF methods, where the modification of some parameters have caused significant variations in the results. Although such variations also depend on the analysed datasets, our method minimises the impact of a poor choice of the parameter values, getting similar results by using different configurations of the algorithm, as seen in Figure 5.

5. Conclusions

This paper proposes a novel statistical approach for background modelling and foreground object detection, in which one Gaussian and one uniform distributions are used to model the background and foreground respectively. It features an update process of the model that is based on stochastic approximation, which has been proved to be suitable and effective as a learning method for real-time algorithms with discarding data.

Experiments have been carried on a variety of complex environments, from outdoor and indoor scenes to synthetic sequences which try to mimic real scenarios. These sequences have provided manually segmented images (GT) in order to apply a set of evaluation metrics which are used to compare the performance of different object detection algorithms.

As a result of this comparison between several of the most cited segmentation techniques, it is noteworthy that our approach has obtained the best results in six of the nine test sequences, while has been competitive
in the rest of them. Additionally to that, an exhaustive study about the influence of the initial values in the segmentation results for each analysed method has been performed. This study shows that both FGD as our method are less sensitive to non-optimal choice of parameter values, getting similar results with different parameter configurations.

We can conclude that the proposed method is a robust alternative to known background modelling algorithms. The usage of the stochastic approximation framework allows to learn from online streams of data in a natural way, so that it is particularly suited for this kind of application. Hence, we are confident that it will help to build robust and accurate intelligent video surveillance systems.

Appendix A. Stochastic Approximation

Let \( \Theta = (\pi_i, \mu_i, C_i) \) be a vector comprising the parameters for mixture component \( i \), where \( i \in \{ \text{Back, Fore} \} \). In what follows we are deriving the algorithm as if the covariance matrix \( C_{\text{Fore}} \) were to be estimated. This is done for notational simplicity purposes, although \( C_{\text{Fore}} \) is never computed in practice. The derivation relies in the methodology proposed in [40, 41].

Let \( \varphi(\Theta_i, t) \) be an arbitrary function of \( \Theta_i \) and the input sample \( t \). Then we define the weighted mean of \( \varphi(\Theta_i, t) \) as:

\[
< \varphi >_i := E[P(i \mid t) \varphi(\Theta_i, t)]
\]

(A.1)

This allows expressing the conditional expectation of \( \varphi(\Theta_i, t) \) as follows:

\[
< \varphi >_i < 1 >_i = \frac{E[P(i \mid t) \varphi(\Theta_i, t)]}{E[P(i \mid t)]} = \int \frac{p(t)P(i \mid t) \varphi(\Theta_i, t) dt}{\pi_i} = \int \frac{p(t)P(i \mid t) \varphi(\Theta_i, t) dt}{\pi_i} \varphi(\Theta_i, t) dt = \int p(t \mid i) \varphi(\Theta_i, t) dt = E [\varphi(\Theta_i, t) \mid i]
\]

(A.2)

Therefore we can rewrite the mixture parameters in terms of the weighted means \( < \varphi >_i \):

\[
\pi_i = < 1 >_i
\]

(A.3)

\[
\mu_i = \frac{< t >_i}{< 1 >_i}
\]

(A.4)

\[
C_i = \frac{< (t - \mu_i)(t - \mu_i)^T >_i}{< 1 >_i}
\]

(A.5)

If we have \( n \) samples (finite case), the linear least squares approximation for \( < \varphi >_i \) is:

\[
< \varphi >_i = \frac{1}{n} \sum_{j=1}^{n} P(i \mid t_j) \varphi(\Theta_i, t_j) = \frac{1}{n} \sum_{j=1}^{n} R_{ji} \varphi(\Theta_i, t_j)
\]

(A.6)

where we should remember that the posterior probability that mixture component \( i \) generated the sample \( t_j \) is noted

\[
R_{ji} = P(i \mid t_j)
\]

(A.7)
As \( n \to \infty \), the approximation of (A.10) converges to the exact value given by (A.1). Next we apply the Robbins-Monro stochastic approximation algorithm (see Section 1.1 of [26]) to estimate iteratively the value of the weighted means \(< \varphi >_i\):

\[
< \varphi >_i (0) = P(i | t_0) \varphi (\Theta_i, t_0)
\]

(A.8)

\[
< \varphi >_i (n) = < \varphi >_{i, n} + \varepsilon (P(i | t_n) \varphi (\Theta_i, t_n) - < \varphi >_{i, n}) - < \varphi >_{i, n-1})
\]

(A.9)

where \( \varepsilon \) is the step size, which is a constant because the system is time varying (see Section 3.2 of [26]). Equation (A.10) is more conveniently written as:

\[
< \varphi >_i (n) = (1 - \varepsilon) < \varphi >_{i, n-1} + \varepsilon R_n \varphi (\Theta_i, t_n)
\]

(A.10)

Now we derive an online stochastic approximation algorithm by applying equation (A.10) to equations (A.3)-(A.5). First we need to define two auxiliary variables:

\[
m_i = < t >_i
\]

(A.11)

\[
M_i = < (t - \mu_i)(t - \mu_i)^T >_i
\]

(A.12)

The corresponding update equations are:

\[
m(n) = (1 - \varepsilon) m(n-1) + \varepsilon R_n t_n
\]

(A.13)

\[
M_i(n) = (1 - \varepsilon) M_i(n-1) + \varepsilon R_n (t_n - \mu_i(n)) (t_n - \mu_i(n))^T
\]

(A.14)

Then we are ready to rewrite (A.3)-(A.5):

\[
\pi_i(n) = (1 - \varepsilon) \pi_i(n-1) + \varepsilon R_n
\]

(A.15)

\[
\mu_i(n) = \frac{m_i(n)}{\pi_i(n)}
\]

(A.16)

\[
C_i(n) = \frac{M_i(n)}{\pi_i(n)}
\]

(A.17)

Acknowledgements

This work has been partially supported by the Ministry of Science and Innovation of Spain under grant TIN2010-15351, project name ‘Probabilistic self organizing models for the restoration of lossy compressed images and video’, and by Junta de Andalucía (Spain) under contract TIC-01615, project name ‘Intelligent Remote Sensing Systems’.
References


