Recent Advanced Statistical Background Modeling for Foreground Detection - A Systematic Survey
Thierry Bouwmans

To cite this version:

HAL Id: hal-00644746
https://hal.archives-ouvertes.fr/hal-00644746
Submitted on 19 Nov 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Recent Advanced Statistical Background Modeling for Foreground Detection - A Systematic Survey

Thierry Bouwman

Laboratoire MIA, Université de La Rochelle, Avenue M. Crépeau, 17000 La Rochelle, France
Tel Phone: (33) 05.46.45.72.02
Email address: tbouwman@univ-lr.fr
Fax number: (33) 05.46.45.82.40

Short Running Title: Advanced Background Modeling: A Systematic Survey

Abstract: Background modeling is currently used to detect moving objects in video acquired from static cameras. Numerous statistical methods have been developed over the recent years. The aim of this paper is firstly to provide an extended and updated survey of the recent researches and patents which concern statistical background modeling and secondly to achieve a comparative evaluation. For this, we firstly classified the statistical methods in term of category. Then, the original methods are reminded and discussed following the challenges met in video sequences. We classified their respective improvements in term of strategies used. Furthermore, we discussed them in term of the critical situations they claim to handle. Finally, we conclude with several promising directions for future research. The survey also discussed relevant patents.

Keywords: Background modeling, Kernel Density Estimation, Mixture of Gaussians, Single Gaussian, Subspace Learning

1. INTRODUCTION

Different applications such as video surveillance [1], optical motion capture [2-4] and multimedia [5-7] need firstly to model the background and then to detect the moving objects. One way to obtain the background is to acquire a background image which doesn't include any moving object but in some environment the background is not available. Furthermore, it can always be changed under critical situations like illumination changes, objects being introduced or removed from the scene. To take into account these problems, many background modeling methods have been developed [8, 9] and these methods can be classified in the following categories:

- **Basic Background Modeling:** In this case, the background is modeled using the average [10] or the median [11] or the histogram analysis over time [12].
- **Statistical Background Modeling:** The background is modeled using a single Gaussian [13] or a Mixture of Gaussians [14] or a Kernel Density Estimation [15]. Statistical variables are used to classify the pixels as foreground or background.
- **Fuzzy Background Modeling:** The background is modeled using a fuzzy running average [16] or Type-2 fuzzy mixture of Gaussians [17]. Foreground detection is made using the Sugeno integral [18] or the Choquet integral [19]. The foreground detection can be performed by fuzzy inferences [335].
- **Background Clustering:** The background model supposes that each pixel in the frame can be represented temporally by clusters. Incoming pixels are matched against the corresponding cluster group and are classified according to whether the matching cluster is considered part of the background. The clustering approach consists in using K-mean algorithm [361] or using Codebook [362].
- **Neural Network Background Modeling:** The background is represented by mean of the weights of a neural network suitably trained on N clean frames. The network learns how to classify each pixel as background or foreground [332][333].
- **Wavelet Background Modeling:** The background model is defined in the temporal domain, utilizing the coefficients of discrete wavelet transform (DWT) [336].
- **Background Estimation:** The background is estimated using a filter. Any pixel of the current image that deviates significantly from its predicted value is declared foreground. This filter may be a Wiener filter [20], a Kalman filter [21] or a Tchebychev filter [22].

Table 1 shows an overview of this classification. The first column indicates the category and the second column the name of each method. The number of papers counted for each method is indicated in the parenthesis. The third column gives the name of the authors who have made the main publication for the corresponding method and the date of the related publication. Other classifications can be found in term of prediction [23], recursion [1], adaptation [24], or modality [25].
### Table 1. Background Modeling Methods: An Overview

<table>
<thead>
<tr>
<th>Category</th>
<th>Methods</th>
<th>Authors - Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Background Modeling</td>
<td>Mean (11)</td>
<td>Lee et al. (2002) [10]</td>
</tr>
<tr>
<td></td>
<td>Histogram over time (13)</td>
<td>Zheng et al. (2006) [12]</td>
</tr>
<tr>
<td>Statistical Background Modeling</td>
<td>Single Gaussian (33)</td>
<td>Wren et al. (1997) [13]</td>
</tr>
<tr>
<td></td>
<td>Mixture of Gaussians (217)</td>
<td>Stauffer and Grimson (1999)</td>
</tr>
<tr>
<td>Fuzzy Background Modeling</td>
<td>Fuzzy Running Average (5)</td>
<td>Sigari et al. (2008) [16]</td>
</tr>
<tr>
<td></td>
<td>Type-2 Fuzzy Mixture of Gaussians (3)</td>
<td>El Baf et al. (2008) [17]</td>
</tr>
<tr>
<td>Background Clustering</td>
<td>K-Means (11)</td>
<td>Butler et al. (2003) [36]</td>
</tr>
<tr>
<td></td>
<td>Codebook (35)</td>
<td>Kim et al. (2005) [36]</td>
</tr>
<tr>
<td>Neural Network Background Modeling</td>
<td>General Regression Neural Network (1)</td>
<td>Culibrk et al. (2006) [332]</td>
</tr>
<tr>
<td></td>
<td>Self Organizing Neural Network (9)</td>
<td>Maddalena and Petrosino (2007) [333]</td>
</tr>
<tr>
<td>Wavelet Background Modeling</td>
<td>Discrete Wavelet Transform</td>
<td>Biswas et al. [336]</td>
</tr>
<tr>
<td>Background Estimation</td>
<td>Wiener Filter (1)</td>
<td>Toyama et al. (1999) [20]</td>
</tr>
<tr>
<td></td>
<td>Kalman Filter (19)</td>
<td>Messelodi et al. (2005) [21]</td>
</tr>
<tr>
<td></td>
<td>Tchebychev Filter (3)</td>
<td>Chang et al. (2004) [22]</td>
</tr>
</tbody>
</table>

All these modeling approaches are used in background subtraction context which presents the following steps and issues: background modeling, background initialization, background maintenance, foreground detection, choice of the feature size (pixel, a block or a cluster), choice of the feature type (color features, edge features, stereo features, motion features and texture features). Developing a background subtraction method, all these choices determine the robustness of the method to the critical situations met in video sequence [5, 20]: Noise image due to a poor quality image source (NI), Camera jitter (CJ), Camera automatic adjustments (CA), Time of the day (TD), Light switch (LS), Bootstrapping (B), Camouflage (C), Foreground aperture (FA), Moved background objects (MO), Inserted background (IB), Waking foreground object (WFO), Sleeping foreground object (SFO) and Shadows (S). The main difficulties come from the dynamic backgrounds and illumination changes:

- **Dynamic backgrounds** often appear in outdoor scenes. Fig. (1). presents four typical examples: Camera jitter, waving trees, water rippling and water surface. The left column shows the original images and the right the foreground mask obtained by the MOG [14]. In each case, there is a big amount of false detections.

- **Illumination changes** appear in indoor and outdoor scenes. Fig. (2). shows an indoor scene in which we can observe a gradual illumination change. This causes false detections in several parts of the foreground mask obtained by the MOG [14]. Fig. (3). illustrates the case of sudden illumination change due to a light on/off. Every pixel in the images is affected by this change which generates a large amount of false detections (see Fig. 3c).
Different datasets benchmarks are available [26-31] to evaluate the robustness of the background subtraction methods against these critical situations which have different spatial and temporal characteristics which must be taken into account to obtain a good segmentation. This challenge must be made in the context of real-time application which runs on common PC and so two constraints are introduced: less computation time (CT) and less memory requirement (MR) as possible. The performance is evaluated using the ROC analysis [32] or the PDR Analysis [33] or the similarity measure [34]. Others performance evaluation methods are proposed and compared in [35, 36]. Reading the literature, two main remarks can be made: (1) The most frequently used models are the statistical ones due to their robustness to the critical situations. (2) There are many recent developments regarding statistical models as can be seen for the MOG model with the acronyms found like GMM [37], TLGMM [38], STGMM [39], SKMGM [40], TAPPMOG [41] and S-TAPPMOG [42]. The objective is then to categorize the statistical models in one paper and classify their recent improvements following the strategies used. We also discuss them following the challenges met in video sequences and evaluate some of them in term of false alarms using the Wallflower dataset [20].

This paper is an extended and updated paper of the surveys on Mixture of Gaussians for background modeling [48] and Subspace Learning for background modeling [334].

The rest of this paper is organized as follows: In Section 2, we firstly provide a background on the statistical background models and a classification of these models. In Section 3, we survey the first generation models and their respective improvements. In Section 4, we classify the second generation models. In Section 5, the third generation models are reviewed. In Section 6, we firstly investigated the performance in term of robustness on dynamic backgrounds and illumination changes and secondly in terms on per-pixel complexity. Then, a comparative evaluation is provided in Section 7. Finally, conclusion and future developments are given.

2. STATISTICAL BACKGROUND MODELING: AN OVERVIEW

The statistical tools provide a good framework to model the background and so many methods have been developed. We classified them in term of category as follows:

- First category: The first way to represent statistically the background is to assume that the history over time of intensity values of a pixel can be modeled by a single Gaussian (SG) [13]. However, a unimodal model cannot handle dynamic backgrounds when there are waving trees, water rippling or moving algae. To solve this problem, the Mixture of Gaussians (MOG) has been used to model dynamic backgrounds [14]. This model has some disadvantages. Background having fast variations cannot be accurately modeled with just a few Gaussians (usually 3 to 5), causing problems for sensitive detection. So, a non-parametric technique was developed for estimating background probabilities at each pixel from many recent samples over time using Kernel density estimation (KDE) [15] but it is time consuming. In [165], Subspace Learning using Principal Component Analysis (SL-PCA) is applied on N images to construct a background model, which is represented by the mean image and the projection matrix comprising the first p significant eigenvectors of PCA. In this way, foreground segmentation is accomplished by computing the difference between the input image and its reconstruction.

Fig. (2). From left to right: The first image presents an indoor scene with low illumination. The second image presents the same scene with a moderate illumination while the third image shows the scene with a high illumination. The fourth image shows the foreground mask obtained with MOG [14]. This sequence called “Time of Day” comes from the Wallflower dataset [20].

Fig. (3). From left to right: The first image presents an indoor scene with light-on. The second image shows the same scene with light-off. The third image shows the foreground mask obtained with MOG [14]. This sequence called “Light Switch” comes from the Wallflower dataset [20].
Table 2: Advanced Statistical Background Modeling: An Overview

<table>
<thead>
<tr>
<th>Category</th>
<th>Methods</th>
<th>Authors - Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mixture of Gaussians (MOG) (217)</td>
<td>Stauffer and Grimson (1999) [14]</td>
</tr>
<tr>
<td></td>
<td>Principal Components Analysis (SL-PCA) (25)</td>
<td>Oliver et al. (1999) [165]</td>
</tr>
<tr>
<td>Second Category</td>
<td>Support Vector Machine (SVM) (9)</td>
<td>Lin et al. (2002) [180]</td>
</tr>
<tr>
<td></td>
<td>Support Vector Regression (SVR) (3)</td>
<td>Wang et al. (2006) [183]</td>
</tr>
<tr>
<td></td>
<td>Support Vector Data Description (SVDD) (6)</td>
<td>Tavakkoli et al. (2006) [186]</td>
</tr>
<tr>
<td>Third Category</td>
<td>Single General Gaussian (SGG) (3)</td>
<td>Kim et al. (2007) [190]</td>
</tr>
<tr>
<td></td>
<td>Mixture of General Gaussians (MOGG) (3)</td>
<td>Allili et al. (2007) [194]</td>
</tr>
<tr>
<td></td>
<td>Independent Component Analysis (SL-ICA) (3)</td>
<td>Yamazaki et al. (2006) [198]</td>
</tr>
<tr>
<td></td>
<td>Incremental Non Negative Matrix Factorization (SL-INMF) (3)</td>
<td>Bucak et al. (2007) [202]</td>
</tr>
<tr>
<td></td>
<td>Incremental Rank-(R₁,R₂,R₃) Tensor (SL-IRT) (2)</td>
<td>Li et al. (2008) [204]</td>
</tr>
</tbody>
</table>

- **Second category**: This second category uses support vector models. The objective is different following the models used. Lin et al. [180] used a SVM algorithm to initialize the background in outdoor scene. Wang et al. [183, 184] modeled the background by using SVR in the case of traffic surveillance scene where illumination changes (TD) appear. Tavakkoli et al. [186-189] applied SVDD to deal with dynamic backgrounds (MB).

- **Third category**: These models generalize the first generation model as the single general Gaussian (SGG) [190-192], the mixture of general Gaussians (MOGG) [193-195] and subspace learning using Independent Component Analysis (SL-ICA) [198, 200], Incremental Non-negative Matrix Factorization (SL-INMF) [202, 203] or Incremental Rank-(R₁,R₂,R₃) Tensor (SL-IRT) [204, 205]. The single general Gaussian (SGG) alleviates the constraint of a strict Gaussian and then shows better performance in the case of illumination changes (TD) and shadow (S). The MOGG have been developed to be more robust to dynamic backgrounds (MB). Subspace learning methods are more robust to illumination changes (LS).

Table 2 shows an overview of the statistical background modeling. The first column indicates the generation and the second column the name of each method. Their corresponding acronym is indicated in the first parenthesis and the number of papers counted for each method in the second parenthesis. The third column gives the name of the authors who have made the main publication for the corresponding method and the date of the related publication. We can see that the MOG with 217 papers is the most modified and improved because it is the most used due to a good compromise between robustness.

In the following sections, we remind the original methods for each generation and we have classify their related improvements in the following way: intrinsic improvements which concern the modification made in the initialization, the maintenance and the foreground detection steps, and extrinsic improvements which consist in using external tools to perform the results.

### 3. FIRST CATEGORY

#### 3.1 Single Gaussian (SG)

Wren et al. [13] proposed to model the background independently at each pixel location (i,j). The model is based on ideally fitting a Gaussian probability density function on the last n pixel’s values. In order to avoid fitting the pdf from scratch at each new frame time t+1, the mean and the variance are updated as follows:

\[
\mu_{t+1} = (1-\alpha)\mu_t + \alpha X_{t+1}
\]

\[
\sigma^2_{t+1} = (1-\alpha)\sigma^2_t + \alpha(X_{t+1} - \mu_{t+1})(X_{t+1} - \mu_{t+1})^T
\]

where \(X_{t+1}\) is the pixel’s current value, \(\mu_t\) is the previous average, \(\sigma_t\) is the previous variance and \(\alpha\) is the learning rate. The foreground detection is made as follows:

\[
|\mu_{t+1} - X_{t+1}| < T\quad \text{the pixel is classified as background}
\]

\[
 otherwise the pixel is classified as foreground
\]

**Improvements**: Medioni et al. [43] operated in the Hue-Saturation-Value (HSV) color space instead of the RGB one. The advantage is that the HSV color model is more robust to gradual illumination changes (TD) because it separates the intensity and chromatic information. Furthermore, HSV permits to eliminate partially camouflage. Zhao et al. [44] used HSV too remarking that the respective distributions of H and S vary naturally a lot and that the distribution of V is the most stable. So, the component H and S are only used when they are stable. Results [44] show better performance in presence of gradual illumination changes (TD) and shadows (S).

**Discussion**: The single Gaussian (SG) is suited for indoor scenes where there are moderate illumination changes.
3.2 Mixture of Gaussians (MOG)

In the context of a traffic surveillance system, Friedman and Russel [45] proposed to model each background pixel using a mixture of three Gaussians corresponding to road, vehicle and shadows. This model is initialized using an EM algorithm. Then, the Gaussians are manually labeled in a heuristic manner as follows: the darkest component is labeled as shadow; in the remaining two components, the one with the largest variance is labeled as vehicle and the other one as road. This remains fixed for all the process giving lack of adaptation to changes over time. For the foreground detection, each pixel is compared with each Gaussian and is classified according to it corresponding Gaussian. The maintenance is made using an incremental EM algorithm for real time consideration. Staugaard and Grimson [14] generalized this idea by modeling the recent history of the color features of each pixel \( \{X_1, ..., X_t\} \) by a mixture of \( K \) Gaussians. We remind below the algorithm.

Principle

First, each pixel is characterized by its intensity in the RGB color space. Then, the probability of observing the current pixel value is considered given by the following formula in the multidimensional case:

\[
P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \eta(X_t, \mu_{i,t}, \Sigma_{i,t})
\]

(1)

where the parameters are \( K \) is the number of distributions, \( \omega_{i,t} \) is a weight associated to the \( i \)th Gaussian at time \( t \) with mean \( \mu_{i,t} \) and standard deviation \( \Sigma_{i,t} \). \( \eta \) is a Gaussian probability density function:

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

(2)

For computational reasons, Stauraard and Grimson [14] assumed that the RGB color components are independent and have the same variances. So, the covariance matrix is of the form:

\[
\Sigma_{i,t} = \sigma^2_{i,t} I
\]

(3)

So, each pixel is characterized by a mixture of \( K \) Gaussians. Once the background model is defined, the different parameters of the mixture of Gaussians must be initialized. The parameters of the MOG’s model are the number of Gaussians \( K \), the weight \( \omega_{i,t} \), associated to the \( i \)th Gaussian at time \( t \), the mean \( \mu_{i,t} \), and the covariance matrix \( \Sigma_{i,t} \).

Remarks:

- \( K \) determined the multimodality of the background and by the available memory and computational power. Stauraard and Grimson [14] proposed to set \( K \) from 3 to 5.
- The initialization of the weight, the mean and the covariance matrix is made using an EM algorithm. Stauraard and Grimson [14] used the K-mean algorithm for real time consideration.

Once the parameters initialization is made, a first foreground detection can be made and then the parameters are updated. Firstly, Stauraard and Grimson [14] used as criterion the ratio \( r_j = \omega_j / \sigma_j \) and ordered the \( K \) Gaussians following this ratio. This ordering supposes that a background pixel corresponds to a high weight with a weak variance due to the fact that the background is more present than moving objects and that its value is practically constant. The first \( B \) Gaussian distributions which exceed certain threshold \( T \) are retained for a background distribution:

\[
B = \arg \min_k \sum_i \omega_{i,j} > T
\]

(4)

The other distributions are considered to represent a foreground distribution. Then, when the new frame incomes at times \( t+1 \), a match test is made for each pixel. A pixel matches a Gaussian distribution if:

\[
sqrt{(X_{t+1} - \mu_{i,t})}^T \Sigma_{i,t}^{-1} (X_{t+1} - \mu_{i,t}) < k \sigma_{i,t}
\]

(5)

where \( k \) is a constant threshold equal to 2.5. Then, two cases can occur:

- Case 1: A match is found with one of the \( K \) Gaussians. In this case, if the Gaussian distribution is identified as a background one, the pixel is classified as background else the pixel is classified as foreground.
- Case 2: No match is found with any of the \( K \) Gaussians. In this case, the pixel is classified as foreground.

At this step, a binary mask is obtained. Then, to make the next foreground detection, the parameters must be updated. Using the match test (5), two cases can occur like in the foreground detection:

Case 1: A match is found with one of the \( K \) Gaussians.

- For the matched component, the update is done as follows:

\[
\omega_{i,t+1} = (1 - \alpha) \omega_{i,t} + \alpha
\]

(6)

where \( \alpha \) is a constant learning rate.

\[
\mu_{i,t+1} = (1 - \rho) \mu_{i,t} + \rho X_{t+1}
\]

(7)
\[ \sigma^2_{i,j,t} = (1-\rho)\sigma^2_{i,j} + \rho(X_{i,j,t} - \mu_{i,j,t})(X_{i,j,t} - \mu_{i,j,t})^T \] (8)

where \( \rho = \alpha \eta(X_{i,j,t}, \mu_{i,t}, \Sigma_{i,t}) \)

- For the unmatched components, \( \mu \) and \( \Sigma \) are unchanged, only the weight is replaced by:

\[ \omega_{i,j,t} = (1-\alpha)\omega_{i,j} \] (9)

Case 2: No match is found with any of the K Gaussians. In this case, the least probable distribution \( k \) is replaced with a new one with parameters:

\[ \omega_{k,j,t} = \text{Low Prior Weight} \] (10)
\[ \mu_{k,j,t} = X_{i,j,t} \] (11)
\[ \sigma^2_{k,j,t} = \text{Large Initial Variance} \] (12)

Once the parameters maintenance is made, foreground detection can be made and so on. Complete studies on the signification and the setting of the parameters can be found in [46, 47][218][289].

**Improvements:** The original MOG presents several advantages. Indeed, it can work without having to store an important set of input data in the running process. The multimodality of the model allows dealing with multimodal backgrounds and gradual illumination changes. Despite it, this model presents some disadvantages: the number of Gaussians must be predetermined, the need for good initializations, the dependence of the results on the true distribution law which can be non-Gaussian and slow recovery from failures. Others limitations are the needs for a complete survey over 100 papers in the period 1999-2007 [106]. Zheng et al. [267, 268] combined multiple features such as brightness, chromaticity and neighborhood information. Recent patents concern block wise approaches [352], texture features [353], motion features [354] and spatial features [355]. An overview of the different features used in the literature is shown in Table 5.

- **Extrinsic improvements:** Another way to improve the efficiency and robustness of the original GMM consist in using external strategies (Table 4). Some authors used Markov Random Fields [107-109], hierarchical approaches [110-113], multi-level approaches [100, 114-118], multiple backgrounds [119, 121], graph cuts [81], multi-layer approaches [122, 123], tracking feedback [128, 129] or specific post-processing [130-131]. Recent patents concern graph cuts approaches [3576, 357].

- **Reducing the computation time:** All the intrinsic and extrinsic improvements concern the quality of the foreground detection but there is another manner to improve the original MOG which consists in reducing the computation time. It achieved by using region of interest [132] [287], by using a variable adaption rate [133], by switching the background model [134] [271], by using space sampling strategies [135][216][238][272] or by using hardware implementation [136, 137][271].

- **Enhancing the foreground detection:** All the previous improvements concern directly the original MOG and the foreground detection results only from it. Another way to improve this method is to enhance the results of the foreground detection by using cooperation with another segmentation method. It achieved by cooperation with a statistical background disturbance technique [138], with color segmentation [139], and with a region based motion detection [140]. Other authors used a cooperation with optical flow [217], block matching [247-248], predictive models [249], texture models [251][303], consecutive frame difference [258][261-262][279-280][282] and basic background subtraction [304-305][330]. A recent patent concern the cooperation with histogram statistics [358].

Table 6 and Table 7 show respectively an overview of the critical situations and the real-time constraints for the different MOG versions that can tackle them better than the original one.
Table 3. Intrinsic improvements of the MOG

<table>
<thead>
<tr>
<th>Background Step</th>
<th>Parameters</th>
<th>Authors - References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background Initialization</td>
<td>Variable K</td>
<td>Zivkovic [49], Cheng et al. [50], Shimada et al. [51], Tan et al. [52], Carminati et al. [53], Klare and Sarka [230], Shimada et al. [237], Shahid et al. [240], Singh and Mitra [248], Wang et al. [278], Huang et al. [288], Wang et al. [307], Zhou et al. [317]</td>
</tr>
<tr>
<td>Variables $\mu$, $\sigma$, $\omega$</td>
<td>Another algorithm: Morellas et al. [54], Lee [55], Ju et al. [241], Singh et al. [245], Singh et al. [246], Wang and Dai [252], Hu et al. [259], Guo et al. [270], Molin [285], Qin et al. [286], Li et al. [315], Wang and Miller [331]</td>
<td></td>
</tr>
<tr>
<td>Background Maintenance</td>
<td>Variable K</td>
<td>Zivkovic [49], Cheng et al. [50], Shimada et al. [51], Tan et al. [52], Klare and Sarka [230], Shimada et al. [237], Singh and Mitra [248], Wang et al. [278], Zhou et al. [317]</td>
</tr>
<tr>
<td>Variables $\mu$, $\sigma$, $\omega$</td>
<td>Maintenance rules: Han and Li [59], Park and Buyn [266]</td>
<td></td>
</tr>
<tr>
<td>Learning rates $\alpha$, $\rho$</td>
<td>Better settings: Zang and Klette [60], White and Shah [67]</td>
<td></td>
</tr>
<tr>
<td>Foreground Detection</td>
<td>Different measure for the matching test</td>
<td>Carminati et al. [53], Ren at al. [79], Lee [80], Sun [81], Morellas et al. [82], Xuehua et al. [261], Rui et al. [262]</td>
</tr>
<tr>
<td></td>
<td>Pixel Persistence Map (PPM)</td>
<td>Pnevmatikakis et al. [75, 76], Landabaso and Pardas [83]</td>
</tr>
<tr>
<td>Probabilities</td>
<td>Yang and Hsu [84], Lee [85], Lien et al. [251], Zhang and Zhou [21]</td>
<td></td>
</tr>
<tr>
<td>Foreground model</td>
<td>Lindstrom et al. [61], Landabaso et al. [63], Withagen et al. [86], Landabaso et al. [263], Feldman et al. [313], Feldman [314], Tian and Wang [318]</td>
<td></td>
</tr>
<tr>
<td>Some matching tests</td>
<td>Zhang et al. [39], Wang and Suter [60]</td>
<td></td>
</tr>
<tr>
<td>Fusion rules</td>
<td>Lien et al. [251]</td>
<td></td>
</tr>
<tr>
<td>Most dominant background</td>
<td>Haque et al. [87, 88, 89]</td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Extrinsic improvements of the MOG

<table>
<thead>
<tr>
<th>Methods</th>
<th>Authors - References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Markov Random Fields</td>
<td>Kumar and Sengupta [107], Zhou and Zhang [108], Schindler and Wang [109], Landabaso et al. [263], Li et al. [291], Dickinson et al. [316], Zhang and Zhou [327], Wang et al. [328]</td>
</tr>
<tr>
<td>Hierarchical approaches</td>
<td>Sun and Yuan [110], Park et al. [111], Chen et al. [112], Zhou et al. [113], Zhong et al. [242], Zhong et al. [264], Li et al. [265]</td>
</tr>
<tr>
<td>Multi-level approaches</td>
<td>Javed et al. [100], Zang and Klette [114], Zhong et al. [115], Cristani et al. [116-118], Yang et al. [325]</td>
</tr>
<tr>
<td>Multiple backgrounds</td>
<td>Su and Hu [119, 120], Porikli [121], Qi et al. [310], Qi et al. [311]</td>
</tr>
<tr>
<td>Graph cuts</td>
<td>Sun [81], Chang and Hsu [257], Li et al. [269], Li et al. [291]</td>
</tr>
<tr>
<td>Multi-layer approaches</td>
<td>Yang et al. [122], Porikli and Tuzel [123], Park and Buyun [266], Huang and Wu [292]</td>
</tr>
<tr>
<td>Features-Cameras strategies</td>
<td>Xu and Ellis [124], Nadimi and Bhanu [125, 126], Concia et al. [127]</td>
</tr>
<tr>
<td>Tracking feedback</td>
<td>Harville [128], Taycher et al. [129], Wang et al. [275], He et al. [301], Yuan et al. [344], Shao et al. [326]</td>
</tr>
<tr>
<td>Post-processing</td>
<td>Turdu and Erdogan [130], Parks and Fels [131], Fazli et al. [306]</td>
</tr>
</tbody>
</table>
Table 5. Features improvements of the MOG

<table>
<thead>
<tr>
<th>Feature Size/Type</th>
<th>Authors - References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Size</td>
<td>Fang et al. [90], Pokrajac and Latecki [91], Wang et al. [275], Zhong et al. [281], Zhang et al. [294], Wang et al. [329]</td>
</tr>
<tr>
<td>Cluster</td>
<td>Bhaskar et al. [92], Cai et al. [243]</td>
</tr>
<tr>
<td>Feature Type</td>
<td></td>
</tr>
<tr>
<td>Color features</td>
<td></td>
</tr>
<tr>
<td>Normalized RGB</td>
<td>Stijman et al. [93], Xu et Ellis [94]</td>
</tr>
<tr>
<td>YUV</td>
<td>Harville et al. [72], Sun [81], Fang et al. [90], Guo et al. [270], Feldman et al. [313], Feldman [314]</td>
</tr>
<tr>
<td>HSV</td>
<td>Sun [81], Xuehua et al. [261], Rui et al. [262], Wang and Tang [274]</td>
</tr>
<tr>
<td>HSI</td>
<td>Wang and Wu [95]</td>
</tr>
<tr>
<td>Luv</td>
<td>Yang and Hsu [96]</td>
</tr>
<tr>
<td>Improved HLS</td>
<td>Setiawan et al. [97]</td>
</tr>
<tr>
<td>YCrCb</td>
<td>Kristensen et al. [98], Ribeiro et al. [99]</td>
</tr>
<tr>
<td>Edge features</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Javed et al. [100], Jain et al. [101], Klare and Sarka [203], Li et al. [253]</td>
</tr>
<tr>
<td>Texture features</td>
<td>Tian and Hampapur [102], Shimada and Taniguchi [250], Huang et al. [235]</td>
</tr>
<tr>
<td>Stereo features</td>
<td></td>
</tr>
<tr>
<td>Disparity</td>
<td>Gordon et al. [103]</td>
</tr>
<tr>
<td>Depth</td>
<td>Harville et al. [72], Silvestre [104]</td>
</tr>
<tr>
<td>Spatial features</td>
<td>Yang and Hsu [84], Dickinson et al. [105], Klare and Sarka [230], Wei et al. [231]</td>
</tr>
<tr>
<td>Motion features</td>
<td>Tang et al. [40]</td>
</tr>
<tr>
<td>Phase features</td>
<td>Xue et al. [312]</td>
</tr>
<tr>
<td>Video features</td>
<td>Wang et al. [106], Wang et al. [239]</td>
</tr>
<tr>
<td>Entropy features</td>
<td>Park et al. [295], Park et al. [296]</td>
</tr>
<tr>
<td>Bayer features</td>
<td>Suhr et al. [297]</td>
</tr>
<tr>
<td>HOG features</td>
<td>Fabian [299], Hu et al. [300]</td>
</tr>
</tbody>
</table>

Table 6. Challenges and MOG Versions

<table>
<thead>
<tr>
<th>Critical Situations</th>
<th>Authors - References</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1 - Noise Image</td>
<td>Xu [221], Teixeira et al. [222], Li et al. [265]</td>
</tr>
<tr>
<td>CS 2-1 - Camera jitter</td>
<td>Campbell-West et al. [219], Xu [221], Achkar and Amer [223], Rao et al. [224], Li et al. [265]</td>
</tr>
<tr>
<td>CS 2-2 - Camera Adjustments</td>
<td>Zen and Lai [225], Molin [285]</td>
</tr>
<tr>
<td>CS 3 - Gradual Illumination Changes</td>
<td>Tian et al. [234], Huang et al. [254], Wang et al. [277], Baloch [283], Huang et al. [288], Lin et al. [309]</td>
</tr>
<tr>
<td>CS 4 - Sudden Illumination Changes</td>
<td>Tian et al. [234], Li et al. [235], Baloch [283], Lin et al. [309], Xue et al. [312], Li et al. [323]</td>
</tr>
<tr>
<td>CS 5-1 - Bootstrapping during initialization</td>
<td>Gao et al. [220]</td>
</tr>
<tr>
<td>CS 5-2 - Bootstrapping during maintenance</td>
<td>Lindstrom et al. [61]</td>
</tr>
<tr>
<td>CS 6 - Camouflage</td>
<td>Guo et al. [270]</td>
</tr>
<tr>
<td>CS 7 - Foreground Aperture</td>
<td>Utasi and Czüni [226]</td>
</tr>
<tr>
<td>CS 8 - Moved background objects</td>
<td>Teixeira et al. [222]</td>
</tr>
<tr>
<td>CS 9 - Inserted background objects</td>
<td>Teixeira et al. [222]</td>
</tr>
<tr>
<td>CS 10 - Multimodal background</td>
<td>Dalley et al. [227], Li et al. [265]</td>
</tr>
<tr>
<td>CS 11 - Waking foreground object</td>
<td>Su and Hu [119], Hu and Su [120]</td>
</tr>
<tr>
<td>CS 12 - Sleeping foreground objects</td>
<td>Cheng et al. [229], Cai et al. [256], Hu et al. [259]</td>
</tr>
<tr>
<td>CS 13 - Shadows Detection</td>
<td>Xu [221], Huang and Chen [232], Zhang et al. [233], Tian et al. [234], Izadi et al. [235], Rahman [236], Chen et al. [260], Landabaso et al. [263], Li et al. [265], Quast et al. [284], Molin [285], Huang et al. [288], Forczmanski and Seweryn [293], Tian and Wang [318], Li and Xu [319], Bin and Liu [320], Liu and Bin [321], Lai et al. [324], Wang et al. [328]</td>
</tr>
</tbody>
</table>
Table 7. Real Time Constraints and MOG Versions

<table>
<thead>
<tr>
<th>Table 7. Real Time Constraints and MOG Versions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-Time Constraints</td>
</tr>
<tr>
<td>Computation Time</td>
</tr>
<tr>
<td>Memory Requirement</td>
</tr>
</tbody>
</table>

**Discussion:** The Mixture of Gaussians (MOG) is adapted for outdoor scene where there are slow multimodal variations in the backgrounds. For the dynamic backgrounds like camera jitter, waving trees and water rippling, this model causes false detections.

### 3.3 Kernel Density Estimation (KDE)

To deal with dynamic backgrounds like camera jitter, waving trees and water rippling, Elgammal et al. [15] proposed to estimate the probability density function for each pixel using the kernel estimator \( K \) for \( N \) recent sample of intensity values \( \{x_1, x_2, ..., x_N\} \) taken consecutively in a time size window \( W \) as follows:

\[
P(x_i) = \frac{1}{N} \sum_{i=1}^{N} K(x_i - x_j) \quad (13)
\]

where \( K() \) is the kernel estimator function which is taken as a Normal Gaussian function \( N(0, \Sigma) \). So, the probability density function is determined as follows:

\[
P(x_i) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x_i - \mu)^T \Sigma^{-1} (x_i - \mu)} \quad (14)
\]

Elgammal et al. [15] assumed that the different color channels are independent with different kernel bandwidths, then the kernel function bandwidth is as follows:

\[
\Sigma = \begin{pmatrix}
\sigma_1^2 & 0 & 0 \\
0 & \sigma_2^2 & 0 \\
0 & 0 & \sigma_3^2
\end{pmatrix} \quad (15)
\]

So, the probability density function can be written as follows:

\[
P(x_i) = \frac{1}{N} \sum_{i=1}^{N} \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2}(x_{i,j} - \mu_j)^2 / \sigma_j^2} \quad (16)
\]

Elgammal et al. [15] detected the foreground using the probabilities and a threshold \( T \) as follows:

If \( P(x_i) < T \) then the pixel classified as foreground else the pixel is classified as background \( (17) \)

At this step, a binary mask is obtained. Then, to make the next foreground detection, the parameters must be updated. For this, Elgammal et al. [15] used two background models: a short term one and a long term one. These two models achieve different objectives:

- The short term model adapts quickly to allow very sensitive detection. This model consists of the most recent \( N \) background sample values. The sample is updated using a selective maintenance mechanism, where the decision is based on the foreground classification.
- The long term model captures a more stable representation of the scene background and adapts to changes slowly. This model consists of \( N \) sample pixels taken from a much larger window in time. The sample is updated using a non selective maintenance mechanism.

So, to combine the advantage of each model and to eliminate their disadvantages, the next foreground detection is obtained by taking the intersection of the two foreground detection coming from the short term model and the long term model. This intersection eliminates the persistence false positives detection from the short term model and extra false positives detection that occur in the long term model results. The only false positives detection that will remain will be rare events not represented in either model. If this rare event persists over time in the scene then the long term model will adapt to it, and it will be suppressed from the result later. Taking the intersection will, unfortunately, suppress true positives in the first model result that are false negatives in the second, because the long term model adapts to foreground as well if they are stationary or moving slowly. To address this problem, all pixels detected by the short term model that are adjacent to pixels detected by the combination are included in the final foreground detection.

**Improvements:** The original KDE present several advantages. The multimodality of the model allows dealing with multimodal backgrounds particularly in fast changes (waving trees, water rippling, etc...). Despite it, this model
present some disadvantages: N frames need to be kept in memory during the entire detection process which is costly memory wise when N is large. The algorithm is time consuming too due the complexity in \(O(N^2)\). To solve these problems, different improvements have been proposed:

- **Intrinsic improvements:** These strategies consist in changing the kernel function \([141-149]\) as shown in Table 8. For the training, some authors propose to decrease the number of samples by determining a proper size of the frame buffer \([143]\), by using a diversity sampling scheme \([150,151]\) or by using a sequential Monte Carlo sampling scheme \([152]\). A recent patent concern the sequential kernel density approximation through mode propagation \([359]\). Furthermore, recursive maintenance \([143-145,153,154,155]\) can be adopted to reduce the computation time. For the foreground detection, different scheme can be used as in \([143,146,147,153-155]\). For the feature type, several features are used instead of the RGB space like the edge features \([156]\) and motion features \([157]\). To choose which features to use, Parag et al. \([158]\) proposed a framework for feature selection.

- **Extrinsic improvements:** Some authors (Table 9) used Markov Random Fields \([155,159]\), hierarchical approaches \([160]\), multiple backgrounds \([161]\) and graph cuts \([162]\).

Enhancing the foreground detection: Another way to improve this method is to enhance the results of the foreground detection by using cooperation with another segmentation method. It achieved by cooperation with the consecutive frame difference \([163]\) or using a subspace learning approach using PCA \([164]\).

The Table 8 and 9 give respectively an overview of the intrinsic and extrinsic improvements. Table 10 and Table 11 show respectively an overview of the critical situations and the real-time constraints for the different KDE versions that can tackle them better than the original one.

<table>
<thead>
<tr>
<th>Table 8. Intrinsic improvements of the KDE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Background Step</strong></td>
</tr>
<tr>
<td>Background Model</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Background Initialization</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Background Maintenance</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Foreground Detection</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 9. Extrinsic improvements of the KDE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Methods</strong></td>
</tr>
<tr>
<td>Markov Random Fields</td>
</tr>
<tr>
<td>Hierarchical approaches</td>
</tr>
<tr>
<td>Multiples backgrounds</td>
</tr>
<tr>
<td>Graph cuts</td>
</tr>
</tbody>
</table>
Table 10. Challenges and KDE Versions

<table>
<thead>
<tr>
<th>Critical Situations</th>
<th>Authors - References</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS 1 - Noise Image</td>
<td>Mao and Shi [150, 151]</td>
</tr>
<tr>
<td>CS 2-1 - Camera jitter</td>
<td>Sheliik and Shah [155]</td>
</tr>
<tr>
<td>CS 2-2 - Camera Adjustements</td>
<td>Cvetkovic et al. [147], Sung et al. [347], Hwang et al. [348]</td>
</tr>
<tr>
<td>CS 3 - Gradual Illumination Changes</td>
<td>Sheliik and Shah [155]</td>
</tr>
<tr>
<td>CS 4 - Sudden Illumination Changes</td>
<td>Sung et al. [48], Hwang et al. [49]</td>
</tr>
<tr>
<td>CS 5-1 - Bootstrapping during initialization</td>
<td>Martel-Brisson and Zaccarin [346]</td>
</tr>
<tr>
<td>CS 5-2 - Bootstrapping during maintenance</td>
<td>Sheliik and Shah [155]</td>
</tr>
<tr>
<td>CS 6 - Camouflage</td>
<td>Tavakkoli et al. [142], Gu et al. [345]</td>
</tr>
<tr>
<td>CS 7 - Foreground Aperture</td>
<td></td>
</tr>
<tr>
<td>CS 8 - Moved background objects</td>
<td>Elgammal et al. [15], Cvetkovic et al. [147]</td>
</tr>
<tr>
<td>CS 9 - Inserted background objects</td>
<td></td>
</tr>
<tr>
<td>CS 10 - Multimodal background</td>
<td></td>
</tr>
<tr>
<td>CS 11 - Waking foreground object</td>
<td></td>
</tr>
<tr>
<td>CS 12 - Sleeping foreground objects</td>
<td></td>
</tr>
<tr>
<td>CS 13 - Shadows Detection</td>
<td>Elgammal et al. [15], Cvetkovic et al. [147], Mao and Shi [150, 151]</td>
</tr>
</tbody>
</table>

Table 11. Real Time Constraints and KDE Versions

<table>
<thead>
<tr>
<th>Real-Time Constraints</th>
<th>Authors - References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation Time</td>
<td>Elgammal [349], Sadeghi et al. [350]</td>
</tr>
<tr>
<td>Memory Requirement</td>
<td>Elgammal [349], Sadeghi et al. [350]</td>
</tr>
</tbody>
</table>

Discussion: The KDE is more adapted for outdoor scene where dynamic backgrounds appear but less suited for illumination changes.

3.4 Subspace Learning using PCA (SL-PCA)

Subspace learning offer a good framework to deal with illumination changes as it allows taking into account spatial information. Oliver et al. [165] proposed to model each background pixel using an eigenbackground model.

This model consists in taking a sample of N images \(\{I_1, I_2, ..., I_n\}\) and computing the mean background image \(\mu_B\) and its covariance matrix \(C_B\).

This covariance matrix is then diagonalized using an eigenvalue decomposition as follows:

\[
L_B = \Phi_B C_B \Phi_B^T \quad (18)
\]

where \(\Phi_B\) is the eigenvector matrix of the covariance of the data and \(L_B\) is the corresponding diagonal matrix of its eigenvalues.

In order to reduce, the dimensionality of the space, only M eigenvectors (M<N) are kept in a principal component analysis (PCA). The M largest eigenvalues are contained in the matrix \(L_M\) and the M vectors correspond to these M largest eigenvalues in the matrix \(\Phi_M\).

Once the eigenbackground images stored in the matrix \(\Phi_M\) are obtained and the mean \(\mu_B\) too, the input image \(I_t\) can be approximated by the mean background and weighted sum of the eigenbackgrounds, \(\Phi_M\).

The coordinate in eigenbackground space of input image \(I_t\) can be computed as follows:

\[
w_t = (I_t - \mu_B)^T \Phi_M \quad (19)
\]

When \(w\) is back projected onto the image space, a reconstructed background image is created as follows:

\[
B_t = \Phi_M w_t^T + \mu_B \quad (20)
\]

Then, the foreground object detection is made as follows:

\[
|I_t - B_t| > T \quad (21)
\]

where T is a constant threshold.

Improvements: The eigenbackground model which we have called SL-PCA provides a robust model of the probability distribution function of the background, but not of the moving objects while they do not have a significant contribution to the model. So, the first limitation of this model is that the size of the foreground object must be small and don’t appear in the same location during a long period in the training sequence. The second limitation appears for the background maintenance. Indeed, it is computationally intensive to perform model updating using the batch mode.
PCA. Moreover, without a mechanism of robust analysis, the outliers or foreground objects may be absorbed into the background model. The third limitation is that the application of this model is mostly limited to the gray-scale images since the integration of multi-channel data is not straightforward. It involves much higher dimensional space and causes additional difficulty to manage data in general. Another limitation is that the representation is not multimodal so various illumination changes cannot be handled correctly. To alleviate these limitations, numerous improvements (25 papers) have been proposed over the recent years. A survey over 15 papers in the period 1999-2009 can be found in [334]. Thus, the different improvements which attempt to solve these four limitations are summarized in the following classification with the recent advances:

- **Alleviate the limitation of the size of the foreground object:** Xu et al. [166, 167] proposed to apply recursively an error compensation process which reduces the influence of foreground moving objects on the eigenbackground model. An adaptive threshold method is also introduced for background subtraction, where the threshold is determined by combining a fixed global threshold and a variable local threshold. Results show more robustness in presence of moving objects. Another approach developed by Kawabata et al. [168] consists in an iterative optimal projection method to estimate a varied background in real time from a dynamic scene with foreground. Firstly, background images are collected for a while and then the background images are compressed using eigenspace method to form a database. After this initialization, a new image is taken and projected onto the eigenspace to estimate the background. As the estimated image is much affected by the foreground, the foreground region is calculated by using background subtraction with former estimated background to exclude the region from the projection. Thus the image whose foreground region is replaced by the former background is projected to eigenspace and then the background is updated. Kawabata et al. [25] proved that the cycle converges to a correct background image. Recently, Quivy and Kumazawa [351] proposed to generate the background images using the Nelder-Mead Simplex algorithm and a dynamic masking procedure. This paper presents an original method that replaces the projection/reconstruction step of the SL-PCA by a direct background image generation. The experiments proved that the proposed method performs better then than the SL-PCA [165], SL-REC [166, 167], and SL-IOP [168] for large and fast moving objects.

- **Dealing with the time requirement and the robustness:** For the maintenance, some authors [169-177] proposed different algorithms of incremental PCA. The incremental PCA proposed by [169] need less computation but the background image is contaminated by the foreground object. To solve this, Li et al. [170, 171] proposed an incremental PCA which is robust in presence of outliers. However, when keeping the background model updated incrementally, it assigned the same weights to the different frames. Thus, clean frames and frames which contain foreground objects have the same contribution. The consequence is a relative pollution of the background model. In this context, Skocaj et al. [172, 173] used a weighted incremental and robust. The weights are different following the frame and this method achieved a better background model. However, the weights were applied to the whole frame without considering the contribution of different image parts to building the background model. To achieve a pixel-wise precision for the weights, Zhang and Zhuang [174] proposed an adaptive weighted selection for an incremental PCA. This method performs a better model by assigning a weight to each pixel at each new frame during the update. Experiments [174] show that this method achieves better results than the SL-IRPCA [170, 171]. Wang et al. [175, 176] used a similar approach using the sequential Karhunen-Loeve algorithm. Recently, Zhang et al. [209] improved this approach with an adaptive scheme. All these incremental methods avoid the eigen-decomposition of the high dimensional covariance matrix using approximation of it and so a low decomposition is allowed at the maintenance step with less computational load. However, these incremental methods maintain the whole eigenstructure including both the eigenvalues and the exact matrix $\Phi_M$. To address this problem, Li et al. [177] proposed a fast recursive and robust eigenbackground maintenance avoiding eigen-decomposition. This method achieves similar results than the SL-IPCA [169] and the SL-IRPCA [170, 171] at better frames rates. Fig. (4). shows a classification of these algorithms following their robustness and their adaptivity.

- **Dealing with the grey scale and the pixel-wise limitations:** Recently, Wu et al. [207] proposed to combine the PCA model with single gaussian model. PCA allow the robustness to illumination changes and the single gaussian to describe color information for each pixel. So, it can detect the chroma changes and remove shadow pixels. An adaptively strategy is used to integrate the two models. A binary graph cut is then used to perform the foreground/background segmentation. In another way, Han and Jain [178] proposed an efficient algorithm using a weighted incremental 2-Dimensional Principal Component Analysis. It is shown that the principal components in 2DPCA are computed efficiently by transformation to standard PCA. To perform the computational time, Han and Jain [178] used an incremental algorithm to update eigenvectors to handle temporal variations of background. The proposed algorithm was applied to 3-channel (RGB) and 4-channel (RGB+IR) data.
Results show noticeable improvements in presence of multimodal background (MB) and shadows (S). To solve the pixel-wise limitation, Zhao et al. [206] used spatio-temporal block instead of pixel. Furthermore, their method consist in applying the candid covariance free incremental principal components analysis algorithm (CCIPCA) which is fast in convergence rate and low in computational complexity than classical IPCA algorithms. Results show more robustness robust to noise and fast lighting changes.

- **Dealing with multimodal illumination changes:** Recently, Dong et al. [211] proposed to use a multi-subspace learning to handle different illumination changes. The feature space is organized into clusters which represent the different lighting conditions. A Local Principle Component Analysis (LPCA) transformation is used to learn separately an eigen-subspace for each cluster. When a current image arrives, the algorithm selects the learned subspace which shares the nearest lighting condition. The results [211] show that the LPCA algorithm outperforms the original PCA [165] algorithm and MOG [14] especially under sudden illumination changes. In a similar way, Kawanishi et al. [213-214] generated the background image which well expresses the weather and the lighting condition of the scene. This method collects a huge number of images by super long term surveillance, classifies them according to their time in the day, and applies the PCA so as to reconstruct the background image.

A recent patent concern a method based on space-time video block and online subspace learning [360]. This method allows a robust incremental update and alleviates the pixel-wise limitations.

The Table 12, Table 13, Table 14 and Table 15 group by type the different improvements of the SL-PCA.

### Table 12. Influence of the foreground objects

<table>
<thead>
<tr>
<th>Methods</th>
<th>Authors - Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursive Error Compensation (SL-REC)</td>
<td>Xu et al. (2006) [166, 167]</td>
</tr>
<tr>
<td>Iterative Optimal Projection (SL-IOP)</td>
<td>Kawabata et al. (2006) [168]</td>
</tr>
<tr>
<td>Simplex Algorithm (SL-SA)</td>
<td>Quivy and Kumazawa (2011) [351]</td>
</tr>
</tbody>
</table>

### Table 13. Time requirement and the robustness

<table>
<thead>
<tr>
<th>Methods</th>
<th>Authors - Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incremental PCA (SL-IPCA)</td>
<td>Rymel et al. (2004) [169]</td>
</tr>
<tr>
<td>Incremental and robust PCA (SL-IRPCA)</td>
<td>Li et al. (2003) [170, 171]</td>
</tr>
<tr>
<td>Weighted Incremental and Robust PCA (SL-WIRPCA)</td>
<td>Skocaj et al. (2003) [172, 173]</td>
</tr>
<tr>
<td>Adaptive Weight Selection for Incremental PCA (SL-AIPCA)</td>
<td>Zhang and Zhuang (2007) [174]</td>
</tr>
<tr>
<td>Sequential Karhunen-Loeve algorithm (SL-SKL)</td>
<td>Wang et al. (2006) [175, 176]</td>
</tr>
<tr>
<td>Adaptive Sequential Karhunen-Loeve algorithm (SL-ASKL)</td>
<td>Zhang et al. [209]</td>
</tr>
<tr>
<td>Fast Recursive Maintenance (SL-FRM)</td>
<td>Li et al. (2006) [177]</td>
</tr>
</tbody>
</table>

### Table 14. Dealing with the grey scale and the pixel-wise limitations

<table>
<thead>
<tr>
<th>Methods</th>
<th>Authors - Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA - Single Gaussian (SL-PCA-SG)</td>
<td>Wu et al. (2009) [207, 208]</td>
</tr>
<tr>
<td>Weighted Incremental 2PCA (SL-WI2DPCA)</td>
<td>Han and Jain (2007) [178]</td>
</tr>
<tr>
<td>Candid Covariance Incremental PCA (SL-CCIPCA)</td>
<td>Zhao et al. (2008) [206]</td>
</tr>
</tbody>
</table>

### Table 15. Dealing with multimodal illumination changes

<table>
<thead>
<tr>
<th>Methods</th>
<th>Authors - Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local Principle Component Analysis on Clusters (LPCA-C)</td>
<td>Dong et al. (2010) [211, 212]</td>
</tr>
<tr>
<td>Local Principle Component Analysis on Separated Sequences (LPCA-SS)</td>
<td>Kawanishi et al. (2009) [213-214]</td>
</tr>
</tbody>
</table>

### 3.5. Discussion

In Section 3, we surveyed the models of the first category and their related improvements. These improvements perform each original algorithm for specified critical situations. However, some authors have recently proposed to use more advanced statistical models as Support Vector models to deal more accurately with dynamics background.
4. SECOND CATEGORY

The second category models use more sophisticated statistical model as support vector machine (SVM), support vector regression (SVR) and support vector data description (SVDD).

4.1 Support Vector Machine (SVM)

Support Vector Machines were introduced by Vapnik et al. [179]. For classification, SVMs work by determining a hyperplane in a high dimensional feature space to separate the training data into two classes. The best hyperplane can be derived by minimizing the margin which represents the least distance from the hyperplane to the data. Using this classification aspect, Lin et al. [180] proposed to use the SVMs for background modeling. Particularly, Lin et al. [180] used a PSVM with probabilistic outputs because the SVM gives only binary outputs. A sigmoid model is used to convert binary SVM scores into posterior probabilities:

\[ p(y = 1 | f) = \frac{1}{1 + \exp(Af + B)} \]  \hspace{1cm} (22)

where \( y \) is binary class label and \( f \) is an output score of the SVM decision function. The two parameters \( A \) and \( B \) are fitted using maximum likelihood estimation from a training set \( \{ f_i, y_i \} \), and derived by minimizing the negative log-likelihood function:

\[ \min \sum t_i \log(p_i) + (1 - t_i) \log(1 - p_i) \]  \hspace{1cm} (23)

where

\[ t_i = \frac{y_i + 1}{2} \quad \text{and} \quad p_i = \frac{1}{1 + \exp(Af_i + B)} \]  \hspace{1cm} (24)

To avoid overfitting and to derive unbiased training for the minimization, a hold-out set is generated from the data by dividing each training set of 80% and 20% respectively. The large subset is used for SVM training, and the smaller one is used for the two parameter minimization. In this context, Lin et al. [180] used 100 images of size 160*120 with known background. Each image is divided into blocks of size 4*4 and considering two features for each block: optical flow and consecutive M frames. When the initialization is finished, the foreground detection is made by thresholding the difference between the background model and the current image.

4.2 Support Vector Regression (SVR)

Given a set of training data, SVR fits a function by specifying an upper bound on a fraction of training data allowed to lie outside of a distance \( \varepsilon \) from the regression estimate. This type of SVR is usually referred to as \( \varepsilon \)-insensitive SVR [181]. For each pixel belonging to the background, a separate SVR is used to model it as a function of intensity. To classify a given pixel as background or not, Wang et al. [183] [184] feed its intensity value to the SVR associated it and threshold the output of the SVR. Let assume a set of training data for some pixel \( p \) obtained from several frames \( \{ (x_i, y_i) \}_{i=1}^{N} \}, \) where \( x_i \) corresponds to the intensity value of pixel \( p \) at frame \( i, \) and \( y_i \) corresponds to the confidence of pixel \( p \) being a background pixel. Once the SVR has been trained, the confidence of the pixel \( p \) in a new frame \( f(x), \) is computed using the following linear regression function:

\[ f(x_i) = \sum_{j \neq i} (a_j - a^*)k(x_i, x_j) + \xi \]  \hspace{1cm} (27)

where \( k(x_i, x_j) \) is a kernel function. The parameters \( a, a^* \) and \( \xi \), called Lagrange multipliers, are obtained by solving an optimization problem using the method of the Lagrange multipliers. Given the SVR-based background model, the intensity of each pixel in a new frame forms the input to the SVR. The output of the SVR represents the confidence that a given pixel belongs to the background. Eventually, a pixel is labelled as background if its intensity distribution of a block.

When an image block \( p(b_i) > T \) is classified as background for M consecutive times, the Fisher linear distance is used:

\[ d(b_i, b_{back}) = \frac{(\mu_i - \mu_{back})^2}{(\sigma_i^2 - \sigma_{back}^2)} \]  \hspace{1cm} (26)

where \( \mu \) and \( \sigma^2 \) are the mean and the variance of the intensity distribution of a block.

When the distance between the two blocks is large, two possible conditions appear. The current block can be either part of a uniform region of a moving object or a new background just revealed. The averaging PSVM probability for the current block over the past M frames is compared with the PSVM probability of the background. If the new average PSVM probability is larger, then the background is replaced by the current block. Continuing this way, the initialization process will be terminated when replacement events do not occur for a consecutive M frames. When the initialization is finished, the foreground detection is made by thresholding the difference between the background model and the current image.
Confidence is between a low threshold $S_l$ and a high threshold $S_h$. Specifically, a binary foreground detection map is formed at frame $t$ as follows:

\[
M'_{x_i} = 0 \text{ if } S_i < f(x_i) < S_h \\
M'_{x_i} = 1 \text{ otherwise}
\] (28)

where $f(x_i)$ is the SVR output and $S = \{S_l, S_h\}$ are the initial thresholds. Then, for each region in the binary map, the SVR-based background model is updated using an online SVR learning algorithm [182].

### 4.3 Support Vector Data Description (SVDD)

Tavakkoli et al. [186] proposed to model the background using support vector data description (SVDD) in videos with quasi-stationary backgrounds. Data domain description concerns the characteristics of a data set [185]. The boundary of the dataset can be used to detect novel data or outliers. A normal data description gives a closed boundary around the data. The simplest boundary can be represented by a hyper-sphere. The volume of this hyper-sphere with center $a$ and radius $R$ should be minimized while containing all the training samples $x_i$. To allow the possibility of outliers in the training set, slack variables $\epsilon_i \geq 0$ are introduced. The error function to be minimized is defined as:

\[
F(R, a) = R^2 + C \sum \epsilon_i
\] (29)

Subjected to the contraints:

\[
\|x_i - a\| \leq R^2 + \epsilon_i \quad \forall i
\] (30)

In equation (1), $C$ is a trade-off between simplicity of the system and its error and is called confidence parameter. After incorporating the constraints (30) into the error function (29) by Lagrange multipliers we have:

\[
L(R, a, \alpha, \gamma, \epsilon_i) = R^2 + C \sum \epsilon_i - \sum \alpha_i \|x_i - a\|^2 - \sum \gamma_i \epsilon_i
\] (31)

$L$ should be maximized with respect to Lagrange multipliers $\alpha_i \geq 0$ and $\gamma_i \geq 0$ and minimized with respect to $R$, $a$ and $\epsilon_i$. Lagrange multipliers $\gamma_i$ can be removed if the constraint $0 \leq \alpha_i \leq C$ is imposed. After solving the optimization problem we have:

\[
L = \sum \alpha_i (x_i \cdot x_i) - \sum \alpha_i \alpha_j (x_i \cdot x_j)
\]

\[
\forall \alpha_i : 0 \leq \alpha_i \leq C
\] (32)

When a new sample satisfies the inequality in (30), then its corresponding Lagrange multipliers are $\alpha_i \geq 0$, otherwise they are zero.

Therefore we have:

\[
\|x_i - a\|^2 < R^2 \rightarrow \alpha_i = 0, \gamma_i = 0
\]

\[
\|x_i - a\|^2 > R^2 \rightarrow \alpha_i = C, \gamma_i > 0
\] (33)

From the above, we can remark that only samples with non-zero $\alpha_i$ are needed in the description of the data set, therefore they are called support vectors of the description. To test a new sample $y$, its distance to the center of the hyper-sphere is calculated and tested against $R$. Tavakkoli et al. [186] used this methodology to built a descriptive boundary for each pixel in the background training frames to generate its model for the background. Then, these boundaries are used to classify their corresponding pixels in new frames as background and foreground pixels. In practice, for each pixel in the scene a single class classifier is trained by using its values in the background training frames. This classifier consists of the description boundary and support vectors, as well as a threshold used to describe the data. For the foreground detection, each pixel in the new frames is classified as background or foreground using its value and its corresponding classifier from the training stage. Feature vectors $x_i$ used in the current implementation are $x_i = [C_i; C_y]$, where $C_i$ and $C_y$ are the red and green chrominance values for pixel $(i, j)$.

**Improvements:** This model presents several advantages: The accuracy is not bounded to the accuracy of the estimated probability density functions and the memory requirement is less than non-parametric techniques. Because support vector data description explicitly models the decision boundary of the known class, it is suitable for novelty detection without the need to use thresholds. Furthermore, the classifier performance in terms of false positive is controlled explicitly. The main disadvantage is that the training of SVDD requires a Lagrange optimization which is computationally intensive. For the maintenance, all the SVDD must be recomputed. To perform the training, Tavakkoli et al. [187] proposed to use a genetic approach to solve the Lagrange optimization problem. The Genetic Algorithm (GA) starts with the initial guess and solves the optimization problem iteratively. In [188][189], Tavakkoli et al. proposed to use an incremental SVDD. In this way, the maintenance is performed too.

### 4.4 Discussion

Support vector models offer a nice framework for background modeling specifically in presence of illumination changes and dynamic backgrounds. Another way to model the background is to perform the first category by using a more adaptive model.
5. THIRD CATEGORY

The third category models generalize the first category model as the single general Gaussian (SGG), the mixture of general Gaussians (MOGG) and subspace learning using Incremental Component Analysis (SL-ICA), Incremental Non-negative Matrix Factorization (SL-INMF) or Incremental Rank-(R₁,R₂,R₃) Tensor (SL-IRT).

5.1 Single General Gaussian (SGG)

Kim et al. [190-192] proposed to model the background using a generalised Gaussian family (GGF) model of distributions to cope with problems from various changes in background and shadows. The idea is that pixel variance fitted sometimes a Laplace one or a Gaussian one. Indeed, pixel variance in a static scene over time taken with the latest camera is closer to a Laplace distribution than a Gaussian, but the Laplace model has limitation for use in various environments. The pixel variance in a static scene over time is defined as:

\[
P(X) = \frac{\rho^\gamma}{2\Gamma(1/\rho)} e^{-(\gamma^\rho - \rho^{\gamma})} \quad \text{with}
\]

\[
\gamma = \frac{1}{\sigma} \left( \frac{\Gamma(3/\rho)}{\Gamma(1/\rho)} \right)
\]

where \(\Gamma(\cdot)\) is a gamma function and \(\sigma^2\) is a variance of the distribution. In Equation (1), \(\rho = 1\) represents a Laplace distribution while \(\rho = 2\) represents a Gaussian distribution.

The optimal parameters of the background model are estimated by the maximization of the likelihood of the observed value:

\[
g_2 = \frac{N}{\sum_{i=1}^{\text{num}}(x_i - \mu)^2} - 3
\]

In practice, Kim et al. [190-192] modelled the background in two parts: a luminance component obtained by a weighted mean of RGB channels and a hue component in HSI color space. The maintenance is made using a selective running average as in [13]. The foreground detection is firstly performed by subtracting the intensity components of the current frame from the background model:

\[
D(x, y) = |I(x, y) - B(x, y)|
\]

where \(I(x, y)\) and \(B(x, y)\) correspond respectively to the luminance of the current frame and the background model. Then, pixels are classified into three categories using two thresholds as follows:

background pixel if \(D(x, y) < T_1 k(x, y)\)
suspicious pixel if \(T_1 k(x, y) \leq D(x, y) \leq T_2 k(x, y)\)
foreground pixel if \(T_2 k(x, y) < D(x, y)\)

where \(k(x, y)\) is a scale parameter. The thresholds \(T_1\) and \(T_2\) are determined using the training frames. The SGG shows better performance than the MOG and the KDE in indoor and outdoor scene.

5.2. Mixture of General Gaussians (MOGG)

Allili et al. [193-195] proposed a finite mixture model of general Gaussians for robust segmentation in the presence of noise and outliers. This model has more flexibility to adapt the shape of data and less sensibility for over-fitting the number of classes than the mixture of Gaussians. Each pixel is characterized by its intensity in the RGB color space. Then, the probability of observing the current pixel value is considered given by the following formula in the multidimensional case:

\[
P(X) = \sum_{i=1}^{K} \omega_i \eta(X_i, \mu_i, \sigma_i, \lambda_i)
\]

where the parameters are \(K\) is the number of distributions, \(\omega_i\) is a weight associated to the \(i\)th Gaussian at time \(t\) with mean \(\mu_i\) and standard deviation \(\sigma_i\). \(\lambda_i = 0\) if the distribution is a Gaussian one and \(\lambda_i = 3\) if the distribution is a Laplace one. \(\eta\) is a Gaussian probability density function:

\[
\eta(X \mu, \sigma, \lambda, \lambda_i) = \prod_{j=t}^{K} A(\lambda_i) \exp \left\{ -B(\lambda_i, X_j - \mu_j, \sigma_j) \right\}
\]

where \(A(\lambda) = \left( \frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)} \right)^{1/\lambda} \) and \(B(\lambda) = \left( \frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)} \right)^{1/\lambda} \).

The optimal number of Gaussians is computed at each time \(t\) by minimizing the criterion Minimum Message Length (MML). If the number of Gaussians at time \(t+1\) is smaller than at time \(t\), the parameters are updated in a similar way than in [14]. The same matching test as in [14] is used to check if a pixel matches a Gaussian. For the labeling, the same scheme that Stauffer and Grimson [14] is used. The MOGG show better performance than the MOG in the presence of shadows (S).
5.3 Subspace Learning

Subspace learning can be made using PCA as seen in the Section 3.4. In the literature [196], there are other methods to reduce the space and these different methods have been classified by Skocaj and Leonardis [197] as reconstructive methods and discriminative methods:

- **Reconstructive subspace learning**: The reconstructive methods allow a well approximation of data and so provide a good reconstruction. Another advantage is that reconstructive methods are unsupervised techniques. Furthermore, reconstructive methods enable incremental updating which is very suitable for real-time application. These methods are task-independents. The most common reconstructive methods are the following: Principal Components Analysis (PCA) [51], Independent Component Analysis (ICA) [52] and Non-negative Matrix Factorization (NMF) [53]. PCA transforms a number of possibly correlated data into a smaller number of uncorrelated data called principal components. ICA is a variant of PCA in which the components are assumed to be mutually statistically independent instead of merely uncorrelated. The stronger condition allows remove the rotational invariance of PCA, i.e. ICA provides a meaningful unique bilinear decomposition of two-way data that can be considered as a linear mixture of a number of independent source signals. Non-negative matrix factorization (NMF) finds linear representations of non-negative data. Given a non-negative data matrix $V$, NMF finds an approximate factorization $V = WH$ into non-negative factors $W$ and $H$. The non-negativity constraints make the representation purely additive, i.e. allowing no subtractions, in contrast to principal component analysis (PCA) and independent component analysis (ICA).

- **Discriminative subspace learning**: The discriminative methods are supervised techniques and allow a well separation of data and so provide a good classification. Furthermore, discriminative methods are spatially and computationally efficient. These methods are task-dependents. The most common discriminative methods are the following: Linear Discriminant Analysis (LDA) [54] and Canonical Correlation Analysis (CCA) [55]. LDA projects the data onto a lower-dimensional vector space such that the ratio of the between-class distance to the within-class distance is maximized. The goal is to achieve maximum discrimination. Canonical correlation analysis (CCA) is a multivariate statistical model that facilitates the study of interrelationships among sets of multiple dependent variables and multiple independent variables. Canonical correlation simultaneously predicts multiple dependent variables from multiple independent variables.

All these methods are originally implemented with batch algorithms which require that the data must be available in advance and be given once altogether. However, this type of batch algorithms is not adapted for the application of background modeling in which the data are incrementally received from the camera. Furthermore, when the dimension of the dataset is high, both the computation and storage complexity grow dramatically. Thus, incremental methods are highly needed to compute in real-time the adaptive subspace for the data arriving sequentially. Following these constraints, the reconstructive methods are the most adapted for background modeling. Furthermore, their unsupervised aspect allows avoid a manual intervention in the learning step. In the following paragraphs, we survey the subspace learning methods applied recently to background modeling: Independent Component Analysis (ICA), Non-negative Matrix Factorization (NMF) and Incremental Rank-(R1,R2,R3) Tensor.

5.3.1 Subspace learning using ICA (SL-ICA)

ICA generalizes the technique of PCA. When some mixtures of probabilistically independent source signals are observed, ICA recovers the original source signals from the observed mixtures without knowing how the sources are mixed. The assumption made is that the observation vectors $X = (x_1,x_2,...,x_M)^T$ can be represented in terms of a linear superposition of unknown independent vectors $S = (s_1,s_2,...,s_M)^T$:

$$X = AS$$

where $A$ is an unknown mixing matrix $(M\times N)$. ICA finds a matrix $W$, so that the resulting vectors:

$$Y = WX$$

recovers the independent vectors $S$, probabilistically permuted and rescaled. $W$ is roughly the inverse matrix of $A$. Applying it to background modeling, the ICA model is given by:

$$Y = W\hat{X}_i$$

$X_i = (x_{i1},x_{i2})^T$ is the mixture data matrix of size $2*K$ in which $K=M*N$. $x_i = (x_{i1},x_{i2},...,x_{iK})$ is the first frame which can contain or not foreground objects and $x_2 = (x_{i1},x_{i2},...,x_{iK})$ is the second frame which contain foreground objects. $W = (w_i,w_j)^T$ is the de-mixing matrix, in which $w_i = (w_{i1},w_{i2})$ with $i=1,2$. $Y = (y_{i1},y_{i2})^T$ is the estimated source signals in which $y_i = (y_{i1},y_{i2},...,y_{ik})$. Several ICA algorithms can be used to determine $W$. Yamazaki et al. [198] used a neural learning algorithm [199]. In another way, Tsai and Lai [200] used a Particle Swarm Algorithm (PSO) [201]. Once $W$ is
determined, there are two ways in the literature to generate the background and the foreground mask images:

- The first case which \( x_i \) contains foreground object like in Yamazaki et al. [198]. Then, the foreground mask for the frames \( x_1 \) and \( x_2 \) is obtained by thresholding respectively \( y_j \) and \( y_2 \). The background image is obtained by replacing regions representing foreground objects in \( x_i \) by the corresponding regions representing background in \( x_2 \).

- The second case which \( x_1 \) contains no foreground object like in Tsai and Lai [200]. Then, the foreground mask for the frames \( x_2 \) is obtained by thresholding \( y_2 \).

The background image is \( y_j \).

The ICA model was tested on traffic scenes by Yamazaki et al. [198] and shown robustness in changing background like illumination changes. In [200], the algorithm was tested on indoor scenes where sudden illumination changes appear.

### 5.3.2 Subspace learning using INMF (SL-INMF)

The non-negative matrix factorization (NMF), with rank \( r \), decomposes the data matrix \( V \in \mathbb{R}^{p \times q} \) into two matrices which are \( W \in \mathbb{R}^{p \times r} \) called the mixing matrix, and \( H \in \mathbb{R}^{r \times q} \) named as the encoding matrix:

\[
V \approx WH \quad (43)
\]

So, NMF aims to find an approximate factorization that minimizes the reconstruction error. Different cost functions based on the reconstruction error have been defined in the literature, but because of its simplicity and effectiveness, the squared error is the most used:

\[
F = \|V - WH\|^2 = \sum_{i=1}^{p} \sum_{j=1}^{q} (V_{ij} - (WH)_{ij})^2 \quad (44)
\]

where subscription \( ij \) stands for the \( ij \)th matrix entity. Applying it to background modelling, Bucak et al. [202, 203] proposed an incremental NMF algorithm. The background initialization is made using \( N \) training frames. So, \( V \) is vector column corresponding to a matrix of size \( p \times q \times N \). The matrices \( W \) and \( H \) are updated incrementally. The foreground detection is made by thresholding the residual error which correspond to the deviations between the background model and the projection of the current frame onto the background model. The INMF has similar performance to dynamic background and illumination changes than the IRPCA proposed by Li et al. [170].

### 5.3.3 Subspace learning using Incremental Rank-(R1,R2, R3) Tensor (SL-IRT)

The different previous subspace learning considered image as a vector. So, the local spatial information is almost lost. Li et al. [204, 205] proposed to use a high-order tensor learning algorithm called incremental rank-(\( R_1, R_2, R_3 \)) tensor based subspace learning to take into account the spatial information. This online algorithm constructs a low-order tensor eigenspace model in which the sample mean and the eigenbasis are updated adaptively. Denote \( G = \{BM_q \in \mathbb{R}^{M \times N} \}_{q=1,2,...,d} \) as a scene’s background appearance sequence with the \( q \)-th frame being \( BM_q \).

Denote \( p_{xy} \) as the \( x \)-th and the \( y \)-th pixel of the scene. The tensor-based eigenspace model for an existing \( A = \{BM_{xy} \in \mathbb{R}^{I \times I} \}_{i=1,2,...,d} \) corresponding to a \( K \) neighborhood of \( \hat{p}_{xy} \) with \( K = I_1 I_2 I_3 = 24^3 \) consists of the maintained eigenspace dimensions \( (R_1, R_2, R_3) \) corresponding to the three tensor unfolding modes, the mode-\( n \) column projection matrices \( U_{(n)}^{(1)} \in \mathbb{R}^{I \times R_n} \), the mode-3 row projection matrix \( V_{(3)}^{(2)} \in \mathbb{R}^{I_1 I_2 I_3} \), the column means \( \bar{I}^{(1)} \) and \( \bar{I}^{(2)} \) of the mode-(1,2) unfolding matrices \( A_{(1)} \) and \( A_{(2)} \), and the row mean \( \bar{I}^{(3)} \) of the mode-3 unfolding matrix \( A_{(3)} \). Given the \( K \)-neighbor image region \( I_{xy} \in \mathbb{R}^{I \times I} \) centered at the \( x \)-th and \( y \)-th pixel \( p_{xy} \) of the current incoming frame \( I_{xy} \in \mathbb{R}^{I \times I} \), the distance \( RM_{xy} \) (determined by the three reconstruction error norms of the three modes) between \( I_{xy} \) and the learned tensor-based eigenspace model is computed. Then, the foreground detection is defined as follows:

\[
p_{xy} \text{ is classified as background if } \exp\left(-\frac{RM_{xy}^2}{2\sigma^2}\right) > T \quad (45)
\]

where \( \sigma \) is a scaling factor and \( T \) denotes a threshold. Thus, the new background model \( BM_{r+1}(x, y) \) at time \( t+1 \) is defined as:

\[
BM_{r+1}(x, y) = H_{xy} \quad \text{if } p_{xy} \text{ is classified as foreground}
\]

\[
BM_{r+1}(x, y) = I_{r+1}(x, y) \quad \text{otherwise} \quad (46)
\]

where \( H_{xy} = (1 - \alpha)MB_{xy} + \alpha I_{r+1}(x, y) \). \( MB_{xy} \) is the mean matrix of \( BM_{r,j} \) at time \( t \) and \( \alpha \) is a learning rate factor. Then, the tensor eigenspace model is updated incrementally and so on. The IRT shows more robustness to noise than the IRPCA proposed by Li et al. [170].
Table 16. Performance evaluation on dynamic backgrounds and illumination changes

<table>
<thead>
<tr>
<th>Method</th>
<th>Dynamic backgrounds</th>
<th>Illumination changes</th>
<th>Indoor/outdoor scene</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG [13]</td>
<td>-</td>
<td>Slow changes</td>
<td>Indoor scene</td>
<td>Motion Capture</td>
</tr>
<tr>
<td>MOG [14]</td>
<td>Slow movement</td>
<td>Slow changes</td>
<td>Outdoor scene</td>
<td>Video Surveillance</td>
</tr>
<tr>
<td>KDE [15]</td>
<td>Yes</td>
<td>Slow changes</td>
<td>Outdoor scene</td>
<td>Video Surveillance</td>
</tr>
<tr>
<td>SL-PCA [165]</td>
<td>-</td>
<td>Slow changes</td>
<td>Outdoor scene (small objects)</td>
<td>Video Surveillance</td>
</tr>
<tr>
<td>SVM [180]</td>
<td>-</td>
<td>Slow movement</td>
<td>Outdoor scene</td>
<td>Video Surveillance</td>
</tr>
<tr>
<td>SVR [183]</td>
<td>Slow changes</td>
<td>Slow changes</td>
<td>Outdoor scene</td>
<td>Video Surveillance</td>
</tr>
<tr>
<td>SVDD [189]</td>
<td>Yes</td>
<td>Slow changes</td>
<td>Indoor scene</td>
<td>Motion Capture</td>
</tr>
<tr>
<td>SGG [190]</td>
<td>-</td>
<td>Slow changes</td>
<td>Outdoor scene</td>
<td>Video Surveillance</td>
</tr>
<tr>
<td>MOGG [194]</td>
<td>Slow movement</td>
<td>Slow changes</td>
<td>Outdoor scene (small objects)</td>
<td>Video Surveillance</td>
</tr>
<tr>
<td>SL-ICA [200]</td>
<td>-</td>
<td>Yes</td>
<td>Yes</td>
<td>Outdoor scene</td>
</tr>
<tr>
<td>SL-INMF [202]</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>Outdoor scene</td>
</tr>
<tr>
<td>SL-IRT [205]</td>
<td>-</td>
<td>Yes</td>
<td>No</td>
<td>Outdoor scene</td>
</tr>
</tbody>
</table>

Table 17. Computational complexity

<table>
<thead>
<tr>
<th>Method</th>
<th>Background Initialization</th>
<th>Background Maintenance</th>
<th>Foreground Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG [13]</td>
<td>O(N)</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>MOG [14]</td>
<td>O(NK)</td>
<td>O(K)</td>
<td>O(K)</td>
</tr>
<tr>
<td>KDE [15]</td>
<td>O(N)</td>
<td>O(n)</td>
<td>O(1)</td>
</tr>
<tr>
<td>SL-PCA [165]</td>
<td>O(N)</td>
<td>O(N+M)</td>
<td>O(1)</td>
</tr>
<tr>
<td>SVM [180]</td>
<td>O(N)</td>
<td>O(N+1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>SVR [183]</td>
<td>O(N)</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>SVDD [189]</td>
<td>O(N)</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>SGG [190]</td>
<td>O(N)</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>MOGG [194]</td>
<td>O(NK)</td>
<td>O(K)</td>
<td>O(K)</td>
</tr>
<tr>
<td>SL-ICA [200]</td>
<td>O(N)</td>
<td>O(M)</td>
<td>O(P)</td>
</tr>
<tr>
<td>SL-INMF [202]</td>
<td>O(N)</td>
<td>O(M)</td>
<td>O(P)</td>
</tr>
<tr>
<td>SL-IRT [205]</td>
<td>O(N)</td>
<td>O(M)</td>
<td>O(P)</td>
</tr>
</tbody>
</table>

6. PERFORMANCE EVALUATION

We have firstly evaluated the ability of each method to deal with dynamics backgrounds and illumination changes. Then, the evaluation is conducted of per-pixel computational complexity and memory requirements.

6.1 Challenges

Table 16 groups the ability of each method to deal with dynamics backgrounds and illumination changes. The third column indicates in which type of scene the method is well suited. The related applications are indicated in the fourth column.

6.2 Computational complexity

The SG is the fastest method because the classification is just made using a threshold and the background maintenance just adapts the mean and the variance. Its complexity depends on N for the initialization. The MOG method has O(NK) complexity with N the number of Gaussian distributions used, typically between 3 and 5. For maintenance, the KDE computes its value in the Gaussian kernels centered on the past n frames, thus raising O(n) complexity, with n typically as high as 100. For the reconstructive subspace learning, their computational complexities are related to the operations needed to compute the elements stored and updated, i.e the principal matrix or the eigenstructures. For example, the incremental tensor subspace learning requires O(I_1R_1(R_1 + R_2 + R_3)) operations [205]. For the foreground detection, the reconstructive subspace learning methods have an estimated complexity per pixel of O(P), where P is the number of the best eigenvectors. For the background maintenance, their complexity is related to M which is the number of samples used to update the model. M=1 if the model is update every frame. Table 17 shows the per-pixel computational complexity of each algorithm at each stage. More details about the complexity of each algorithm can be found in their corresponding papers.

6.3 Memory requirements

For the statistical methods, the memory complexity per pixel is the same as the computational complexity. At classification time, reconstructive approaches require a memory complexity per pixel O(P), with P the number of the best eigenvectors. However, at training time these methods require allocation of all the N training images, with an O(N) complexity. For the reconstructive subspace learning, the memory requirements are related to the elements stored and updated, i.e the principal matrix or the eigenstructures. For example, the incremental tensor subspace learning requires O(I_1R_1(R_1 + R_2 + I_1R_2)) memory units [205].
7. COMPARISON

We have chosen to compare different improvements of the MOG for dynamic backgrounds and the subspace learning models (SL-PCA, SL-ICA, SL-INMF and SL-IRT) for illumination changes. Results on the Wallflower dataset provided by Toyama et al. [20] are presented. We collected these results because of how frequent its use is in this field. This frequency is due to its faithful representation of real-life situations typical of scenes susceptible to video surveillance. Moreover, it consists of seven video sequences in which each sequence presenting one of the difficulties a practical task is likely to encounter (i.e. illumination changes, dynamic backgrounds). The size of the images is 160*120 pixels. A brief description of the Wallflower image sequences can be made as follows:

- **Moved Object (MO):** A person enters into a room, makes a phone call, and leaves. The phone and the chair are left in a different position. This video contains 1747 images.
- **Time of Day (TOD):** The light in a room gradually changes from dark to bright. Then, a person enters the room and sits down. This video contains 5890 images.
- **Light Switch (LS):** A room scene begins with the lights on. Then a person enters the room and turns off the lights for a long period. Later, a person walks in the room, switches on the light, and moves the chair, while the door is closed. This video contains 2715 images.
- **Waving Trees (WT):** A tree is swaying and a person walks in front of the tree. This video contains 287 images.
- **Camouflage (C):** A person walks in front of a monitor, which has rolling interference bars on the screen. The bars include similar color to the person’s clothing. This video contains 353 images.
- **Bootstrapping (B):** The image sequence shows a busy cafeteria and each frame contains people. This video contains 3055 images.
- **Foreground Aperture (FA):** A person with uniformly colored shirt wakes up and begins to move slowly. This video contains 2113 images.

For each sequence, the ground truth is provided for one image when the algorithm has to show its robustness to a specific change in the scene. Thus, the performance is evaluated against hand-segmented ground truth. Three terms are used in the evaluation: False Positive (FP) is the number of background pixels that are wrongly marked as foreground; False Negative (FN) is the number of foreground pixels that are wrongly marked as background; Total Error (TE) is the sum of FP and FN.

7.1 MOG and its improvements

For the first category, we compare the MOG with its main improvements. Table 18 and Fig. (5) group the experimental results found in the literature for the algorithms chosen which are:

1. **The original algorithm:** Stauffer and Grimson [14].
2. **Three intrinsic improvements:** White et al. [67] which used a better setting for the learning rates using Particle Swarm Optimization, Wang et al. [60] which modified the foreground detection step using a mixed color space i.e. a normalized RGB color space for pixels with high intensities and in RGB color space for pixels with low intensities and Setiawan et al. [97] which used the IHLS space.
3. **Three extrinsic improvements:** Schindler et al. [109] which used the MRFs to smooth the results spatially, Cristani et al. [117] which proposed the Spatial-Time Adaptive Per Pixel Mixture Of Gaussian called S-TAPPMOG and Cristani et al. [118] which used an adaptive spatio-temporal neighborhood analysis called ASTNA. For these two last algorithms, the authors don’t give the result for the following image sequences: Moved Object, Time of Day and Light Switch. So, we have indicated for these the Total Error without these image sequences.

From Table 18, we can see that the original MOG gives the bigger total of error. A better setting of the learning rate and the threshold T using the PSO [67] divides approximately by 2 the number of total errors. The use of the IHLS color space [97] decreases a lot the number TE which becomes just under 10 000. The improvement proposed by Wang et al. [60] gives the better results for the intrinsic improvements. For the extrinsic improvements, the best results are obtained by MOG using MRF proposed by Schindler et al. [109] followed by S-TAPPMOG [117] and ASTNA [118]. For all the methods, the image sequences Light Switch (LS) gives the larger amount of false positive. Here, the best result is obtained by the method proposed by Schindler et al. [109]. The use of IHLS [97] gives it best improvement for the image sequences Camouflage (C) and for the method proposed by Wang et al. [30], it is the image sequences Waving Trees (WT). In conclusion, this performance evaluation shows that taking into account spatial and temporal consistency improves the results in a significant way. Fig. (6) presents the overall performance for the five first algorithms. It is not intended to be a definitive ranking of these algorithms. Such a ranking is necessarily task-, sequence-, and application dependent.
Fig. (5). Results on the Wallflower dataset [26] for the MOG and its improvements.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>MO</th>
<th>TD</th>
<th>LS</th>
<th>WT</th>
<th>C</th>
<th>B</th>
<th>FA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test image</td>
<td><img src="image1.jpg" alt="Image" /></td>
<td><img src="image2.jpg" alt="Image" /></td>
<td><img src="image3.jpg" alt="Image" /></td>
<td><img src="image4.jpg" alt="Image" /></td>
<td><img src="image5.jpg" alt="Image" /></td>
<td><img src="image6.jpg" alt="Image" /></td>
<td><img src="image7.jpg" alt="Image" /></td>
</tr>
<tr>
<td>Ground Truth</td>
<td><img src="image8.jpg" alt="Image" /></td>
<td><img src="image9.jpg" alt="Image" /></td>
<td><img src="image10.jpg" alt="Image" /></td>
<td><img src="image11.jpg" alt="Image" /></td>
<td><img src="image12.jpg" alt="Image" /></td>
<td><img src="image13.jpg" alt="Image" /></td>
<td><img src="image14.jpg" alt="Image" /></td>
</tr>
<tr>
<td>MOG</td>
<td>Stauffer et al. [14]</td>
<td><img src="image15.jpg" alt="Image" /></td>
<td><img src="image16.jpg" alt="Image" /></td>
<td><img src="image17.jpg" alt="Image" /></td>
<td><img src="image18.jpg" alt="Image" /></td>
<td><img src="image19.jpg" alt="Image" /></td>
<td><img src="image20.jpg" alt="Image" /></td>
</tr>
<tr>
<td>MOG with PSO</td>
<td>White et al. [67]</td>
<td><img src="image21.jpg" alt="Image" /></td>
<td><img src="image22.jpg" alt="Image" /></td>
<td><img src="image23.jpg" alt="Image" /></td>
<td><img src="image24.jpg" alt="Image" /></td>
<td><img src="image25.jpg" alt="Image" /></td>
<td><img src="image26.jpg" alt="Image" /></td>
</tr>
<tr>
<td>MOG using IHLS</td>
<td>Setiawan et al. [97]</td>
<td><img src="image27.jpg" alt="Image" /></td>
<td><img src="image28.jpg" alt="Image" /></td>
<td><img src="image29.jpg" alt="Image" /></td>
<td><img src="image30.jpg" alt="Image" /></td>
<td><img src="image31.jpg" alt="Image" /></td>
<td><img src="image32.jpg" alt="Image" /></td>
</tr>
<tr>
<td>Improved MOG</td>
<td>Wang et al. [60]</td>
<td><img src="image33.jpg" alt="Image" /></td>
<td><img src="image34.jpg" alt="Image" /></td>
<td><img src="image35.jpg" alt="Image" /></td>
<td><img src="image36.jpg" alt="Image" /></td>
<td><img src="image37.jpg" alt="Image" /></td>
<td><img src="image38.jpg" alt="Image" /></td>
</tr>
<tr>
<td>MOG with MRF</td>
<td>Schindler et al. [109]</td>
<td><img src="image39.jpg" alt="Image" /></td>
<td><img src="image40.jpg" alt="Image" /></td>
<td><img src="image41.jpg" alt="Image" /></td>
<td><img src="image42.jpg" alt="Image" /></td>
<td><img src="image43.jpg" alt="Image" /></td>
<td><img src="image44.jpg" alt="Image" /></td>
</tr>
<tr>
<td>S-TAPMOG</td>
<td>Cristani et al. [117]</td>
<td><img src="image45.jpg" alt="Image" /></td>
<td><img src="image46.jpg" alt="Image" /></td>
<td><img src="image47.jpg" alt="Image" /></td>
<td><img src="image48.jpg" alt="Image" /></td>
<td><img src="image49.jpg" alt="Image" /></td>
<td><img src="image50.jpg" alt="Image" /></td>
</tr>
<tr>
<td>ASTNA</td>
<td>Cristani et al. [118]</td>
<td><img src="image51.jpg" alt="Image" /></td>
<td><img src="image52.jpg" alt="Image" /></td>
<td><img src="image53.jpg" alt="Image" /></td>
<td><img src="image54.jpg" alt="Image" /></td>
<td><img src="image55.jpg" alt="Image" /></td>
<td><img src="image56.jpg" alt="Image" /></td>
</tr>
</tbody>
</table>

Table 18. Comparison on the Wallflower dataset [26] for the MOG and its improvements.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Problem Type</th>
<th>MO</th>
<th>TD</th>
<th>LS</th>
<th>WT</th>
<th>C</th>
<th>B</th>
<th>FA</th>
<th>TE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOG [14]</td>
<td>FN 0 1008 1633 1323 398 1874 2442 27053</td>
<td>FP 0 20 14169 341 3098 217 530</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOG with PSO [67]</td>
<td>FN 0 807 1716 43 2386 1551 2392</td>
<td>FP 0 6 772 1689 1463 519 572 13916</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOG-IHLS [97]</td>
<td>FN 0 379 1146 31 188 1647 2327</td>
<td>FP 0 99 2298 270 467 333 554 9739</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improved MOG – FD [60]</td>
<td>FN 0 597 1481 44 106 1176 1274</td>
<td>FP 0 358 669 288 413 134 541 7081</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOG with MRF [109]</td>
<td>FN 0 47 204 15 16 1060 34</td>
<td>FP 0 402 546 311 467 102 604 3808</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-TAPMOG [117]</td>
<td>FN - - - 153 643 1414 1912</td>
<td>FP - - - 1152 1382 811 377 7844</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASTNA [118]</td>
<td>FN - - - 253 823 2349 1900</td>
<td>FP - - - 100 1173 73 360 7031</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
INMF is the second in term of performance. For example, only SL-PCA gives FP in the sequence called “Moved Objects” (MO) due the fact that the model is not update over time. In the same way, SL-INMF gives the biggest total error due to its results on the sequence called “Camouflage” (C). This is confirmed by the Fig. (9), which shows the performance without this sequence. In this case, SL-INMF is the second in term of performance. SL-ICA has globally good performance except for the sequence called “Bootstrap” (B) by giving less true detection.

**7.2 Subspace learning models**

SL-PCA which is from the first category is compared with the subspace learning models from the third category: SL-IRT, SL-PCA and SL-INMF. Table 19 and Fig. (7) group the experimental results found in the literature for the subspace learning algorithms. From Table 19, we can see that SL-ICA gives the smallest TE followed by SL-IRT, SL-PCA and SL-INMF. Fig. (8) shows the overall performance. This ranking has to be taken with precaution because a poor performance on one video influences the TE and then modifies the rank. The main interpretation is that all the models are robust to illumination changes as can be seen on the sequence called “Time of Day” (TD) and “Light Switch” (LS). Otherwise, the subspace learning algorithms are more or less adapted for specific situations. For example, only SL-PCA gives FP in the sequence called “Moved Objects” (MO) due the fact that the model is not update over time. In the same way, SL-INMF gives the biggest total error due to its results on the sequence called “Camouflage” (C). This is confirmed by the Fig. (9), which shows the performance without this sequence. In this case, SL-INMF is the second in term of performance. SL-ICA has globally good performance except for the sequence called “Bootstrap” (B) by giving less true detection. SL-IRT seems to be more efficient in the case of camouflage. SL-PCA gives less FN than FP. For SL-ICA, SL-INMF and SL-IRT, it is the contrary. We can remark that SL-ICA provides very less FP than FN. It is interesting in video-surveillance because it decreases false alarms.

**Fig. (6). Overall performance on the Wallflower dataset** [26] for the MOG and its improvements.
8. CURRENT & FUTURE DEVELOPMENTS

This paper attempts to provide a comprehensive survey on statistical background modeling for foreground detection and to provide some structural categories for the strategies developed in 300 papers and 10 recent patents. Thus, we proposed a classification in term of category. For the MOG and KDE, we proposed a classification for their related improvements in two classes respectively called intrinsic and extrinsic improvements. Strategies adding spatial and temporal information in the different steps or in added process proved their abilities to improve the robustness of the original model to the critical situations. Cooperation with other segmentations has shown their interests too. Methods which reduce the computation time permit to deal with the constraints of real-time application. Although significant progress has been made, there is still work to be done and we believe that a systematic comparative evaluation must be made and thus determine the best combination of strategies. In this context, we encourage the evaluation using the Wallflower dataset like in [60, 67, 97, 109].

Furthermore, two main investigations seem to be very promising:
- For dynamic backgrounds, combination between SG, MOG and KDE [337-339] which allows to gives more robustness when there are waving trees, water surfaces and water rippling in the scene.
- For illumination changes, robust PCA [340-344] in which the background is modeled by a low rank subspace that can gradually change over time, while the moving foreground objects are considered as the correlated sparse outliers.

In conclusion, this paper allows the reader to survey recent advances on statistical background modeling and it can effectively guide him to select the best improvement for his specific application. Particularly, this survey paper allows: 1) Developers to choose the appropriate improvement to tackle the critical situations met in their application. 2) Researchers to have a recent state-of-the-art and so easily identify new ideas. 3) Reviewers to verify quickly the originality of a paper.

9. ACKNOWLEDGMENTS

The authors would like to thanks Serhat Selcuk BUCAK (Pattern Recognition and Image Processing Lab, Michigan State University, USA) who provided the results obtained by his algorithm SL-INMF, Du-Ming TSAI and Shia-Chih LAI (Department of Industrial Engineering and Management, Yuan-Ze University, Taiwan) who provided the results obtained by their algorithm SL-ICA.

CONFLICT OF INTEREST

The author declared no conflict of interest

REFERENCES

Multiple Cameras Towards Human Detection in Smart-Rooms, Landabaso J., Pardas M. Cooperative Background Modelling using Hsinchu, Taiwan, December 2006.


Incremental and Robust PCA, ICIP 2003, Barcelona, Spain, August 2006.


[301] He S., Guan Q., Xu S., Li Y., Wu Y. Improving mixture Gaussian background model by integrating trace information obtained from Kalman filter, ICCAS 2010, pages 378-382, Chengdu, China, July 2010.


[312] Xue G., Sun J., Song L. Background Subtraction based on Phase and Distance Transform under Sudden Illumination Change, ICIP 2010, pages 3465-3468, Hong Kong, China, September 2010.


Chang F., Hsu W., Chueh Method and system for foreground detection using multi-modality fusion graph cut, GB 2467804, 2010.

Xie D. Background model initializing and updating method based on video monitoring, CN 101489121, 2009.


Zhu S., Zhao Y. Background modeling method based on space-time video block and online subspace learning, CN 101645171, 2010
