Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey

Thierry Bouwmans, Fida El Baf, Bertrand Vachon

To cite this version:

HAL Id: hal-00338206
https://hal.archives-ouvertes.fr/hal-00338206
Submitted on 12 Nov 2008
Background Modeling using Mixture of Gaussians for Foreground Detection - A Survey

T. Bouwmans, F. El Baf, B. Vachon

Laboratoire MIA, Université de La Rochelle, Avenue M. Crépeau, 17000 La Rochelle, France

Abstract: Mixture of Gaussians is a widely used approach for background modeling to detect moving objects from static cameras. Numerous improvements of the original method developed by Stauffer and Grimson [1] have been proposed over the recent years and the purpose of this paper is to provide a survey and an original classification of these improvements. We also discuss relevant issues to reduce the computation time. Firstly, the original MOG are reminded and discussed following the challenges met in video sequences. Then, we categorize the different improvements found in the literature. We have classified them in term of strategies used to improve the original MOG and we have discussed them in term of the critical situations they claim to handle. After analyzing the strategies and identifying their limitations, we conclude with several promising directions for future research.

Keywords: Background modeling, foreground detection, mixture of gaussians.

INTRODUCTION

Background modeling is often used in different applications to model the background and then detect the moving objects in the scene like in video surveillance [2, 3], optical motion capture [4, 5-6] and multimedia [7, 8-10]. The simplest way to model the background is to acquire a background image which doesn't include any moving object. In some environments, the background isn't available and can always be changed under critical situations like illumination changes, objects being introduced or removed from the scene. To take into account these problems of robustness and adaptation, many background modeling approaches have been developed and the most recent surveys can be found in [2, 11, 12]. These background modeling methods can be classified in the following categories: Basic Background Modeling [13-15], Statistical Background Modeling [1, 16, 17], Fuzzy Background Modeling [18, 19, 20] and Background Estimation [3, 21, 22]. Other classifications can be found in term of prediction [23], recursion [2], adaptation [24], or modality [25]. 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 [3, 26]: 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 objects (IBO), Multimodal background (MB), Waking foreground object (WFO), Sleeping foreground object (SFO) and Shadows (S). Different datasets benchmarks are available [27-32] to evaluate the robustness of the background subtraction methods against these critical situations which have different spatial and temporal characteristics which must be take 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 [33] or the PDR Analysis [34] or the similarity measure [35]. Others performance evaluation methods are proposed and compared in [36, 37]. Reading the literature, two remarks can be made: (1) The most used model is certainly the pixel-wise MOG one proposed by Stauffer and Grimson [1] due to a good compromise between robustness to the critical situations and the constraints (CT, MR). (2) There are many improvements of this MOG model as shown by the different acronyms found like GMM [38], TLGMM [39], STGMM [40], SKMGM [41], TAPPMOG [42] and S-TAPPMOG [43]. All the developed strategies attempt to be more rigorous statistically or to introduce spatial and/or temporal constraints. The motivation of this paper concerns these two remarks. The objective is to group these different strategies in one paper and classify them following the different steps and issues of the MOG model. We also discuss them following the challenges met in video sequences. The idea is not to present a numeric comparative evaluation using ROC or PDR analysis of the different algorithms due their large number and because this evaluation can be generally found in the related paper. Furthermore, the purpose of this paper is limited to temporal pixel-wise MOG model. So, the spatial MOG model versions [44-49] are not discussed here.

The rest of this paper is organized as follow: In the Section 2, we firstly remind the original pixel-wise MOG model [1]. Then, we provide a classification of the modified versions found in the literature. In the Section 3, we survey the intrinsic model improvements which concern the initialization and the maintenance of the parameters, the foreground detection and by extension the features used. In the Section 4, we have classified the extrinsic model improvements which use the temporal and spatial constraints in their external strategies in the sense that how the MOG is applied. In the Section 5, the methods which reduce compu-
2. BACKGROUND MODELING USING MIXTURE OF GAUSSIANS

In the context of a traffic surveillance system, Friedman and Russel [50] 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 [51]. 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. Stauffer and Grimson [1] generalized this idea by modeling the recent history of the color features of each pixel \( \{X_r,...,X_t\} \) by a mixture of K Gaussians. We remind below the algorithm.

2.1. 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:

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

(2)

For computational reasons, Stauffer and Grimson [1] assumed that the RGB color components are independent and have the same variances. So, the covariance matrix is of the form:

\[
\Sigma_{i,t} = \sigma_{i,t} I
\]  

(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. Stauffer and Grimson [1] 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. Stauffer and Grimson [1] 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. First, Stauffer and Grimson [1] used as criterion the ratio \( r_j = \omega_j / \sigma_{j,t} \) 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:

\[
B = \arg \min_i \left( \sum_{j=1}^{B} \omega_{i,j} > T \right)
\]  

(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 the Mahalanobis distance

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

(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 foreground one, the pixel is classified as background else this case, if the Gaussian distribution is identified as 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.

Case 2: No match is found with any of the K Gaussians.

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)

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

(8)

where \( \rho = \alpha \eta (X_{t+1}, \mu_{i,t}, \Sigma_{i,t}) \).

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

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

(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:

\[ \pi_{k,t} = \text{Low Prior Weight} \]  
\[ \mu_{k,t+1} = X_{t+1} \]  
\[ \sigma^2_{k,t} = \text{Large Initial Variance} \]

Once the parameters maintenance is made, foreground detection can be made and so on. Complete studies on the significance and the setting of the parameters can be found in [52, 53].

2.2. Discussion

Firstly, we discussed the relevance of the pixel-wise MOG model and the approximation made at each step by grouping the different remarks found about it in the literature. Then, we have identified the critical situations that can handle this model.

Relevance and approximations made: Modeling the background using the MOG implies the assumption that the background distributions and the foreground ones are Gaussians but it isn’t always the case. For example, Kitahara et al. [54] show that the distribution in indoor scene is much closer to a Laplace model than a Gaussian one. In another way, Wang et al. [55] remark that the intensity doesn’t conform to the Gaussian distributions when the intensity varies abruptly like in the case of flickering trees (MB) in outdoor scene. For the initialization, the MOG needs that the number of Gaussians K is fixed and is the same for all pixels. It isn’t optimal because the multimodality is variable spatially and temporally. For the initialization of the mean, the variance and the weight, a series of training frames absent of moving objects is needed but in some environment, it isn’t possible to obtain frames without moving objects. Furthermore, this storage causes a amount of memory required in this step. For the maintenance phase, Greifenhagen et al. [56] characterizes it statistical behavior making different parameters initialization using real data and simulated data. The experiment shows that only the means are estimated and tracked correctly. The variance and the weights are unstable and unreliable but Greifenhagen et al. [56] remark that this is not really a problem because the weights and the variances are not used in a subsequent processing step. For the foreground detection, the main drawback is mainly due to the matching test as explained in [57]. Indeed, the maintenance is made according to the classification using this matching test which is an approximation of the MAP. The consequence is that the tail of the distribution is not updated when the maintenance is only made when the new value is between this interval given by the mean and standard deviation as in the Equation (5). When only the part of the distribution defined by this interval is used, a new Gaussian kernel is estimated which are a lower standard deviation. So, the standard deviation becomes underestimated and values in the tail are wrongly classified as foreground. This cause problem when the background is not updated due to the error in the classification. This gives more pixels classified as foreground causing false positive detections. This problem increased over time. For the feature size, Stauffer and Grimson [1] have chosen the pixel but this pixel-wise aspect has the main disadvantage that the temporal and spatial constraints aren’t tackled. For the feature type, Stauffer and Grimson [1] used the RGB space but these color components aren’t independent and so the simplification made in Equation (3) for the covariance matrix isn’t right. This simplification conducts to false positive and false negative detections as explained in [58]. More complete analysis of the assumption made in the MOG model is available in [59]. In this paper, Gao et al. [59] have presented an error analysis in the different steps providing both theoretical analysis and experimental validation.

Dealing with the challenges: The MOG model deals with the movement in the background (MB) due to the multimodality in the representation step. The maintenance step permits to cope up with the gradual illumination changes (TD) and the learning rate \( \alpha \) determines the speed of adaptation to illumination changes (TD) but also the speed of the incorporation of background objects moved or inserted (MBO, IBO) and the speed of incorporation of a moving object which stopped (SFO) as explained in [60]. This is one of the disadvantages of the MOG and generally in the literature the authors make a compromise between the two processes. Another disadvantage is that the pixel-wise aspect prevents to handle some critical situations (LS, B) which can be only detected spatially and temporally. Furthermore, some critical situations need pre-processing or post-processing ([NI, CJ, CA]). For these two types of critical situations, Stauffer and Grimson [1] proposed nothing to deal with it. Another disadvantage is the use of the RGB which can permit to make well shadows detection (S). In resume, the original pixel-wise MOG model is design well for (TD, MB), is medium for the (MBO, IBO, SFO), and isn’t design for the (NI, CJ, CA, LS, B, C, FA, WFO, S).

To solve these different limitations, many improvements can be found in the literature and consist in being more rigorous in the statistical sense and in introducing the spatial and temporal constraints. We have classified them as intrinsic and extrinsic model improvements which are presented respectively in the Section 2 and Section 3. Table 1 and Table 2 show respectively an overview of the critical situations and the real-time constraints versus the different MOG versions that can tackle them better than the original one.

3. INTRINSIC MODEL IMPROVEMENTS

Intrinsic model improvements concern directly the MOG model like the initialization and the maintenance of the parameters, the foreground detection and by extension the features used.

3.1. Number of Components - K

K is fixed experimentally and constant over time in Stauffer and Grimson [1]. K fixed and the same for each pixel is not optimal in term of detection and computational
gradual illumination changes (TD), Han and Lin [38] update the MOG via adaptive ISODATA algorithm. This method is less adaptive than the others because K isn’t updated after the training period.

3.2. Initialization of the Weight, the Mean and the Variance

Stauffer and Grimson [1] initialized the weight, the mean and the variance of each Gaussian using a K-means algorithm. A training sequence without foreground is needed. This initialization scheme is improved as follows:

**By using another algorithm for the initialization:** Pavlidis et al. [99] show that an EM algorithm [51] is a superior initialization method that provides fast learning and exceptional stability to the foreground detection. This is especially true when initialization happens during challenging weather conditions like fast moving clouds or other cases of multimodal background (MB). The disadvantage is that the EM algorithm is computationally intensive. In the continuity, Lee [84] proposes an approximation of the EM algorithm to avoid unnecessary computation or storage. His results on both synthetic data and surveillance videos show better learning efficiency and robustness in case of (B) and (MB) than the algorithm used by Friedman and Russel [50], Stauffer and Grimson [1], and Bowden et al. [132].

**By allowing presence of foreground objects in the training sequence:** Following the assumption that the background’s pixels appear in the image sequence with the maximum frequency, Zhang et al. [60] propose a background reconstruction algorithm to initialize the MOG even in presence of foreground in the scene. Another approach proposed by Amintoosi et al. [82] consists in a QR-decomposition based algorithm. To be more robust when large parts of the background are occluded by moving objects and parts of the background are never seen, Lepiski [83] proposes to use the optic flow to reason about if the background has been seen or not. This method is more robust in the case of bootstrapping (B).

3.3. Maintenance of the Weight, the Mean and the Variance

Stauffer and Grimson [1] updated the weight, the mean and the variance of each Gaussian with an IIR filter using a K-means algorithm. This method is less adaptive than the others because K isn’t updated after the training period.

### Table 1. Challenges and MOG Versions

<table>
<thead>
<tr>
<th>Critical Situations</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Image (NI)</td>
<td>[61-64]</td>
</tr>
<tr>
<td>Camera jitter (CJ)</td>
<td>[58, 62, 65, 66]</td>
</tr>
<tr>
<td>Camera Adjustements (CA)</td>
<td>[67]</td>
</tr>
<tr>
<td>Auto Gain Control</td>
<td>[68]</td>
</tr>
<tr>
<td>Auto White Balance</td>
<td>[69]</td>
</tr>
<tr>
<td>Automatic Exposure Correction</td>
<td>[70]</td>
</tr>
<tr>
<td>Gradual Illumination Changes (TD)</td>
<td>[1, 24, 38, 63, 70-73, 74, 75]</td>
</tr>
<tr>
<td>Sudden Illumination Changes (LS)</td>
<td>[24, 61, 63, 65, 67, 68, 70, 71, 74-81]</td>
</tr>
<tr>
<td>Bootstrapping during initialization (B)</td>
<td>[59, 82, 83]</td>
</tr>
<tr>
<td>Bootstrapping during running (B)</td>
<td>[84-88]</td>
</tr>
<tr>
<td>Camouflage (C)</td>
<td>[42, 72, 73, 88-92]</td>
</tr>
<tr>
<td>Foreground Aperture (FA)</td>
<td>[93]</td>
</tr>
<tr>
<td>Moved background objects (MBO)</td>
<td>[60, 63, 70, 74, 75, 80, 85, 87, 88]</td>
</tr>
<tr>
<td>Inserted background objects (IBO)</td>
<td>[60, 63, 70, 74, 75, 85, 87, 88]</td>
</tr>
<tr>
<td>Multimodal background (MB)</td>
<td>[1, 61, 64, 84, 86, 90, 94-107]</td>
</tr>
<tr>
<td>Waking foreground object (WFO)</td>
<td>[74-75, 80, 85, 87, 88]</td>
</tr>
<tr>
<td>Sleeping foreground objects (SFO)</td>
<td>[1, 42, 60, 74, 75, 78, 79, 85, 87, 88, 108-114]</td>
</tr>
<tr>
<td>Shadows and highlights (S)</td>
<td>[61, 62, 68-70, 81, 101, 109, 115-123]</td>
</tr>
</tbody>
</table>

### Table 2. Real Time Constraints and MOG Versions

<table>
<thead>
<tr>
<th>Real-Time Constraints</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation Time (CT)</td>
<td>[24, 43, 92, 124-131]</td>
</tr>
<tr>
<td>Memory Requirement (MR)</td>
<td>[127, 128]</td>
</tr>
</tbody>
</table>

Time. To solve this problem, Zivkovic [94] proposes an online algorithm that estimates the parameters of the MOG and simultaneously selects the number of Gaussians using the Dirichlet prior. The consequence is that K is dynamically adapted to the multimodality of each pixel. In the same idea, Cheng et al. [95] propose a stochastic approximation procedure which is used to recursively estimate the parameters of MOG and obtains the asymptotically optimal number of Gaussians. Another approach proposed by Shimada et al. [96] consists in a dynamic control of the number of Gaussians. This approach automatically changes the number of Gaussians in each pixel. The number of Gaussians increases when pixel values often change. On the other hand, when pixel values are constant in a while, some Gaussians are eliminated or integrated. Another idea proposed by Tan et al. [97] consists in a modified online EM procedure to construct an adaptive MOG in which the number K can adaptively reflect the complexity of pattern at the pixel. Carminati et al. [98] estimate the optimal number of K Gaussians for each pixel in a training set using an ISODATA algorithm. This method is less adaptive than the others because K isn’t updated after the training period.
correlation of the innovation to accurately update gradually the parameters without empirical parameter selection.

(2) Maintenance Mechanisms: The maintenance mechanism used in the original MOG model [1] has the advantage to adapt robustly to gradual illumination change (TD) and the learning rate $\alpha$ determines the speed of adaptation to illumination changes (TD) but also the speed of the incorporation of background objects moved or inserted (MBO, IBO) and the speed of incorporation of a moving object which stopped (SFO). So, one mechanism deals with different critical situations which have different temporal characteristics. To decouple the adaptation mechanism and the incorporation mechanism, Wang et al. [70] and Zhang et al. [60] propose to use a set of counter which represents the number of times a pixel is classified as a foreground pixel. When this number is larger than a threshold, the pixel is considered as background. This gives a time limit on how long a pixel can be considered as a static foreground pixel. Another approach developed by Lindstrom et al. [85] uses a CUSUM detector to determine when components should be transferred from a foreground component to a background component. The CUSUM detector utilizes local counters for the foreground components. If the counter exceeds a threshold, the corresponding foreground component is added to the background model. If the maximum number of background components has been reached, it replaces the least likely background component. The foreground component remains in the model because other pixels might still be explained by it.

(3) Selective Maintenance: The selective maintenance is based on the idea that the maintenance can be performed by using a different rule following some temporal or spatial conditions. For example, some authors [86,108, 109, 114] propose to select the rule for each pixel following it previous classification and other ones [71] propose to select the rule for all the pixels following the illumination change detected in the scene.

Table 3 shows an overview of the different methods for the initialization and maintenance of the parameters.

Another way to perform the maintenance is to adjust the learning rates using spatial and/or temporal information.

### 3.4. Learning rates $\alpha$ and $\rho$

The learning rates control the adaptation rate with a compromise between being fast enough to adapt to changes and slow enough to store a useful temporal history. At fast adaptation rates, the distribution quickly becomes dominated by a single Gaussian. In [1], $\alpha$ is a constant and $\rho = \alpha \eta \left( \bar{X}_i, \mu_i, \Sigma_i \right)$. However, a constant learning rate $\alpha$ causes problems with slow initial adaptation giving an

<table>
<thead>
<tr>
<th>Method</th>
<th>Training Sequence</th>
<th>Parameters Initialization</th>
<th>Parameters Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stuaffer and Grimson [1]</td>
<td>Clean sequence</td>
<td>manually set 3 to 5</td>
<td>K-means no</td>
</tr>
<tr>
<td>Zivkovic et al. [94]</td>
<td>-</td>
<td>Estimation at each pixel</td>
<td>-</td>
</tr>
<tr>
<td>Cheng et al. [95]</td>
<td>-</td>
<td>Dirichlet prior</td>
<td>EM algorithm yes</td>
</tr>
<tr>
<td>Shimada et al. [96]</td>
<td>-</td>
<td>Asymptotically optimal</td>
<td>IIR Filter</td>
</tr>
<tr>
<td>Tan et al. [97]</td>
<td>-</td>
<td>Incrementation/decrementation</td>
<td>EM algorithm yes</td>
</tr>
<tr>
<td>Cuevas et al. [130]</td>
<td>-</td>
<td>Estimation at each pixel</td>
<td>-</td>
</tr>
<tr>
<td>Tian et al. [133]</td>
<td>-</td>
<td>Automatic strategy</td>
<td>-</td>
</tr>
<tr>
<td>Carminati et al. [98]</td>
<td>-</td>
<td>Frequency of pixel value</td>
<td>-</td>
</tr>
<tr>
<td>Pavlidis et al. [99]</td>
<td>-</td>
<td>Estimation at each pixel</td>
<td>ISODATA algorithm no</td>
</tr>
<tr>
<td>Lee [84]</td>
<td>-</td>
<td>ISODATA algorithm</td>
<td>IIR Filter</td>
</tr>
<tr>
<td>Zhang et al. [60]</td>
<td>Allow presence MO</td>
<td>manually set 3 to 5</td>
<td>Sequential K-means no</td>
</tr>
<tr>
<td>Amintoosi et al. [82]</td>
<td>Allow presence MO</td>
<td>manually set 3 to 5</td>
<td>QR-Decomposition no</td>
</tr>
</tbody>
</table>
exaggerated importance to the initial values and gives no adaptation to the environment changes. For the learning rate $\rho$, the problem is that $\eta(X, \mu, \Sigma, r)$ is usually very small, this makes convergence tolerably slow. The different approaches can be classified as follows:

**By using better settings:** In a simple way, Zang and Klette [137] use a constant $\alpha$ and propose to use a reasonable constant value for the learning rate $\rho$ by setting this value according to specific situations in the captured scene to control the speed of adaptation. This setting improves the convergence. Sensitivity $\alpha$ is also closely tied with $T$. In this context, White et al. [138] propose to automatically tune the learning rate $\alpha$ and the background threshold parameter $T$ using Particle Swarm Optimization [134]. For this, a measure of quality called the F-measure quantifies in one scalar value for a frame how similar a resulting foreground detection image is to the ground-truth. Then, the fitness space is explored automatically to locate the couple $(\alpha, T)$ which maximize the fitness. This method has the advantage to tune $\alpha$ and $T$ following the sequence. Their results show that this method reduces the total pixel error of the MOG significantly on the seven critical situations represent in the Wallflower dataset [27] particularly for the camouflage (C) and the movement in the background (MB).

**By adapting the learning rates:** Bowden et al [87, 132, 135] propose to use a different learning rate between the steps of the parameters initialization and the parameters maintenance. During the initialization, the expected sufficient statistics update equations are used. When the first $L$ samples are processed, the algorithm changes to the $L$-recent window version. The expected sufficient statistics update equations provide a good estimate at the beginning before all $L$ samples can be collected. This initial estimate improves the accuracy of the estimate allowing fast convergence on a stable background model. The $L$-recent window update equations give priority to recent data to adapt to changes in the environment. A more adaptive approach is proposed by Lee [136] and used a different learning rate for each Gaussian. The convergence speed and approximation results are significantly improved. Harville et al. [88] adapted the learning rate following the pixel activity and so used temporal information in the objective to deal with (MBO, IBO, SFO). Lindstrom [85] proposed a progressive learning rate where the relative update speed of each Gaussian component depends on how often the component has been observed and so integrates temporal information in the maintenance scheme. Thus, in a natural way, giving varying learning rates, where current data is allowed greater relative influence on the parameter estimates of newer and seldomly observed components. Wang et al [70] adapt the learning rate using spatial information to be adaptive to the illumination context. If the pixel number of detected foreground pixels is larger than a threshold (e.g., 70% of the whole image pixels), Wang et al. adjust the learning rate to a high value because LS is detected. Otherwise, the learning rate is set to a low value. In the same idea, Porikli [24] adjusts the learning rate according to the illumination change of random pixels that do not correspond at an object. An illumination change score is computed. If an illumination change is detected in the pixels (TD, LS) i.e. the illuminations score is higher than a threshold, the learning rate is higher and then adapt quickly. Furthermore, Porikli [24] adapts the time period of the maintenance mechanism following the illumination change score. The idea is that no maintenance is needed if no illumination change is detected and a quick maintenance is necessary otherwise. Another adaptive approach is to use tracking feedback. Liu et al [110] divide foreground pixels into moving or static according to whether they are classified moving or static objects by the tracking process. So, background pixels are updated slightly on moving objects in order to preserve the original background, while greatly on stationary objects and background to make any changes in scenes quickly incorporated into the background model. In this way, the background model will be adapted quickly to the changes in the environment while keeping it from damaged by moving objects at the same time. Pnevmatikakis et al. [111] propose a feedback tracking architecture too. Indeed, the learning rate is modified on elliptical regions around the targets. Thus, instead of a constant value, a spatiotemporal adaptation of the learning rate is used. $\alpha(x, y, t)$ is large if the pixel $(x, y)$ isn’t near a target at time $t$ and $\alpha(x, y, t)$ is small if the pixel $(x, y)$ is near a target at time $t$. This delays the incorporation of the targets and depending on the selection of the small learning rate and the motion of the targets can be sufficient. In some cases, when targets stay immobile for every long periods even the small learning rate will gradually incorporate these targets into the background. If this case occurs, the original weight update mechanism is replaced by the following mechanism: The weight of the current Gaussian is decreased and the other ones are increased with a rate that is inversely proportional to the mobility of the target, as this is estimated from the state of the Kalman tracker for this particular target. This incorporation prevention mechanism is not always in effect and is only activated when targets are small and rather immobile. This tracking feedback is improved in Pnevmatikakis et al. [112] by modified the learning rate following the speed and the size of the targets and is useful for large vehicles, where their speed can be large, but their uniform colors can lead to fading into the background. Some authors use a combination of the previous adaptive strategies. For example, Schoonees et al. [139] propose a strategy which combined [87,136]. During the initialization period, $\alpha$ is variable and $\alpha = 0/t$ while $\alpha \geq \alpha_{\text{min}}$. After this condition isn’t respected, $\alpha = \alpha_{\text{min}}$. The learning rate $\rho$ is variable for each Gaussian and is computed as follows: $\rho_{i, t} = \alpha_i / \omega_{i, t}$ for the component matched and $\rho_{i, t} = 0$ otherwise. In the same way, Leotta et al. [140] use the same strategy for the initialization and the maintenance than [87] for the learning rate $\alpha$ but it is like [136] for the learning rate $\rho_{i, t}$. 
Table 4. Settings and Adaptation of the Learning Rates

<table>
<thead>
<tr>
<th>Method</th>
<th>Learning Rate $\alpha$</th>
<th>Learning Rate $\rho$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stauffer and Grimson [1]</td>
<td>Constant - set manually</td>
<td>Constant - set manually</td>
</tr>
<tr>
<td>Zang et al. [137]</td>
<td>Constant - set manually</td>
<td>Constant - Reasonable value – set manually</td>
</tr>
<tr>
<td>White et al. [138]</td>
<td>Constant - Particle Swarm Parameter [134]</td>
<td>-</td>
</tr>
<tr>
<td>Bowden et al. [87] [132]</td>
<td>Variable during the initialization Constant during the maintenance</td>
<td>Variable during the initialization Constant during the maintenance</td>
</tr>
<tr>
<td>Bowden et al. [135]</td>
<td>Variable during the initialization Constant during the maintenance</td>
<td>$\rho = \alpha$</td>
</tr>
<tr>
<td>Lee [136]</td>
<td>Constant - set manually</td>
<td>Variable and different for each Gaussian</td>
</tr>
<tr>
<td>Harville et al. [88]</td>
<td>Variable - pixel activity</td>
<td>$\rho = \alpha$</td>
</tr>
<tr>
<td>Lindstrom et al. [85]</td>
<td>Variable - occurrence of the components</td>
<td>$\rho = \alpha$</td>
</tr>
<tr>
<td>Wang et al. [70]</td>
<td>Variable - illumination activity</td>
<td>$\rho = \alpha$</td>
</tr>
<tr>
<td>Porikli [24]</td>
<td>Variable - illumination activity</td>
<td>$\rho = \alpha$</td>
</tr>
<tr>
<td>Liu et al. [110]</td>
<td>Variable - tracking feedback</td>
<td>Variable following the tracking feedback</td>
</tr>
<tr>
<td>Pnevmatikakis et al. [111]</td>
<td>Two values - small near target and large otherwise</td>
<td>$\rho = \alpha$</td>
</tr>
<tr>
<td>Pnevmatikakis et al. [112]</td>
<td>Two values - constant small value near target and variable large value otherwise</td>
<td>$\rho = \alpha$</td>
</tr>
<tr>
<td>Schoonees and Power [139]</td>
<td>Variable during the initialization Constant during the maintenance</td>
<td>Variable and different for each Gaussian</td>
</tr>
<tr>
<td>Leotta et al. [140]</td>
<td>Variable during the initialization Constant during the maintenance</td>
<td>Variable and different for each Gaussian [136]</td>
</tr>
</tbody>
</table>

Table 4 shows an overview of the different methods for the settings and adaptation of the learning rates.

These different adaptive approaches introduce spatial and temporal information in the learning rate and so improve the behavior of the maintenance to the critical situations. Another way to enhance the maintenance is to use a variable adaptation rate to optimize the computation time as developed in the Section 4.

3.5. Threshold $T$

$T$ is highly sensitive to the scene environment and is tied to $\alpha$. As seen in the Section 3.4, White et al. [138] propose to automatically tune $\alpha$ and $T$ using Particle Swarm Optimization [134] and results show that this method reduces the total pixel error of the MOG significantly. Another approach developed by Haque et al. [104-106] replace $T$ with another less sensitive parameter and so improve the robustness. Instead of $T$, this approach uses the background-foreground separation threshold $S$ used in the basic background subtraction. Low $S$ gives high quality detection independent of $\alpha$. Results show better performance than the original MOG and the modified version presented in [141].

3.6. Foreground Detection

Stauffer and Grimson [1] made the foreground detection in three phases: ordering following the ratio $r_j$, labelling the Gaussian using the background threshold $T$ and classifying the current pixel using the matching test. This scheme is improved in the literature in five different ways:

**By using a different measure for the matching test:** In this case, the ordering and labeling phases are conserved and only the matching test is changed to be more exact statistically. Indeed, Stauffer and Grimson [1] checked every new pixel against the $K$ existing distribution using the Equation (5) to classify it in background pixel or foreground one. This test gives a binary mask and was chosen in an approximation to the true Maximum A Priori (MAP) solution to permit a real-time implementation. This approximation causes false positive detections due the use of the interval defined by the mean and the standard deviation in Equation (5). To solve this problem, Ren et al. [142], Carminati et al. [98] and Lee [136] suggest to use a likelyhood maximization instead of the approximation MAP. Another approach is proposed by Pavlidis et al. [99] which use the Jeffreys divergence measure to check if the incoming pixel value can be ascribed to any of the existing $K$ Gaussians. Results presented by Pavlidis et al. [99] show that the false positives are reduced.

**By using a Pixel Persistence Map (PPM):** Pardas et al. [143] use a Pixel Persistence Map (PPM) which is a map of the same dimension as the frames containing at each location $(x,y)$ the weight of the Gaussian matching the current color of the pixel $(x,y)$. Small PPM values indicate foreground
pixels, while large indicate background ones. So, the foreground detection is made using a decision test with a threshold on the PPM. The disadvantage is that when there is camera jitter (CJ) or movement in the background (MB), the PPM needs to be bounded by a very low threshold in order not to consider flickering pixels as foreground but this threshold tends to discard true foreground pixels. To solve this problem, Pnevmatikakis et al. [111] propose to adapt the threshold on the PPM in a spatiotemporal fashion using a tracking feedback. This scheme presents the advantage that flickering pixels are avoided far from the targets, while the targets themselves are not affected. The drawback of this strategy is the delayed detection of new very small targets. Then, Pnevmatikakis et al. [112] improved this tracking feedback scheme by using the covariance matrix of the target. We can note that the ordering and labeling phases aren’t used and only a matching test is used to ascribe a pixel to a Gaussian.

By using the probabilities: Another way to make the foreground detection consists in using the MOG background likelihood but it is very sensitive. Instead of using directly the learned MOG background likelihood to determine whether a pixel belongs to background or not, Yang and Hsu [101] propose to include the spatial and color features to construct the background model. Considering the neighborhood around a pixel, Hsu models the prior probability of this pixel:

\[
P(B) = \frac{1}{Z} \exp(-(n_{hv} H + n_D D))
\]

where \(Z\) is a normalization term, \(n_{hv}\) and \(n_D\) indicate the number of foreground pixels in the horizontal/vertical and diagonal neighborhood respectively, and \(H\) and \(D\) are the corresponding penalties. A pixel is classified as background if:

\[
P(X_i/B)P(B) > T \quad \text{where } T \text{ is a fixed threshold.}
\]

Lee [141] uses the posterior probability in the foreground detection. This probability can be expressed in terms of the mixture components \(P(G_k)\) and \(P(X_i/G_k)\) as follows:

\[
P(B \mid X_i) = \frac{\sum_{k=1}^{K} P(X_i \mid G_k)P(G_k)P(B \mid G_k)}{\sum_{k=1}^{K} P(X_i \mid G_k)P(G_k)}
\]

The mixture components are obtained through the learning step. The estimation of \(P(B \mid G_k)\) involves heuristics encapsulating the domain knowledge of the background process. In the first model presented by Friedman and Russel [50], the Gaussians are manually labeled and remain fixed. In the original MOG [1], the probability is equal to 1 for Gaussians with the highest ratio \(r = \varrho / \sigma\) covering a certain percentage of observations, and 0 for all others. In this approach, Lee [141] trains a sigmoid function on \(r\) to approximate \(P(B \mid G_k)\) using logistic regression:

\[
\hat{P}(B \mid G_k) = 1 / (1 + e^{-\alpha_0 \varphi / \sigma_0 + b})
\]

Once \(P(X_i)\) and \(P(B \mid G_k)\) are estimated, the foreground is composed of pixels where

\[
P(B \mid X_i) < 0.5.
\]

By using a foreground model: In these approaches, the authors [57, 85, 109] use one model for the foreground and one for the background. The background model is the Mixture of Gaussians per pixel, as the background will be different per pixel. Oh the other hand, foreground objects are not static and move over all pixels. So, the foreground model is spatially shared and only one model is used for describing all foreground objects depicted in all pixels. Withagen et al. [57] and Pardas et al. [109] model the foreground by a uniform distribution. Lindstrom et al. [85] use a global Mixture of Gaussians for the foreground. Once the foreground model is determined, Withagen et al. [57] and Pardas et al. [109] propose to compare the probabilities that the pixels belongs to the background or the foreground. Respectively noted \(P(\bar{x} / B)\) and \(P(\bar{x} / F)\). Following the Bayes criterion, a pixel is assigned to be background when:

\[
P(\bar{x} / F) < C_f P(B) / C_f P(B) + C_b P(F)
\]

where the \(C_f\) and \(C_b\) are the costs of false positive detection and false negative detection, and \(P(F)\) and \(P(B)\) are the a priori foreground and background probabilities. In practice, Withagen et al. [57] set these parameters as follows:

\[
P(\bar{x} / F) = 1 / 256^4 . \quad P(B) = 1 - P(F) \quad \text{and} \quad C_b / C_f = 0.05 .
\]

In this approach, there is no need to order and label kernels as in the scheme proposed by Stauffer and Grimson [1]. Classification is based on the highest probability, meaning that if the models are accurate, classification results will be optimal i.e. the total cost of misclassification will be minimal. So, the classification is performed by choosing the class which is most probable. Similarly, Pardas et al. [109] classify a pixel as foreground if \(P(F)P(\bar{x} / F) > P(B)P(\bar{x} / B)\). This test is the same one than in the Equation 17 proposed by Withagen et al. [57] with \(C_b / C_f = 1\).

By using some matching tests: Wang et al. [70] proposed to classify pixels as foreground when the pixel doesn’t match any Gaussian. The matching tests differ following the intensity value of the pixel to be more robust when the intensity is low and in the case of shadows and highlights (S). Results presented by Wang et al. [70] show that the foreground detection is improved especially in the case of dark scenes. Another approach developed by Miller et al. [58] consists in using the cylindrical color model defined by Kim et al. [144] instead of the spherical color model used by Stauffer and Grimson [1]. A pixel is classified as background if two matching tests are verified otherwise the pixel is classified as foreground. In these two approaches, the ordering and labeling phases aren’t used. To incorporate spatial information, Zhang et al. [40] defined a new matching test in a spatiotemporal Gaussian mixture model (STGMM). This test permits to handle the complex motion of the background by considering every background pixel to
be fluctuating both in intensity and in its neighboring region. Quantitative evaluations demonstrate that the proposed STGMM performs better than MOG.

**By using the most dominant background model:** The ordering and labeling phases are conserved and the matching test is the same but instead of using a percentage of the Gaussian only the most dominant is considered to represent the background. Furthermore, instead of using the corresponding $\mu$ as the reference value, Haque et al. [104-106] proposed to use the most recent pixel value $m$ represented by this Gaussian. The idea is to avoid any artificial value as the representation. Then, the foreground detection is made using $S$ as explained in Section 3.5.

All the previous foreground detection used the pixel as element of comparison. To enhance the robustness, other feature sizes are used.

### 3.6 Feature Size

Stauffer and Grimson [1] used the pixel as element of the image but the pixel-wise aspect has some disadvantages. To alleviate this, some authors use block [145, 146] or cluster [147] instead of the pixel.

**Block-wise approaches:** Fang et al. [145] apply a block-wise MOG model which consists on a vector of 3x3 neighbors of the current pixel. The advantage is to take into account the spatial dimension to imp rove the robustness. The foreground detection is obtained with pixel-wise precision due to the cooperation with a color segmentation, and so Fang et al. [145] eliminate the disadvantage of the original MOG which can’t detect foreground completely when the foreground’s texture and color are homogeneous and have low contrast with background (C, FA). Another approach proposed by Latecki et al. [146] decomposes the video into spatiotemporal blocks. A dimensionality reduction technique is applied to obtain a compact vector representation of each block. The block vectors provide a joint representation of texture and motion patterns in videos. These block vectors are used as features. The foreground detection is improved decreasing the processing time due to the reduction of the number of input vectors per frame. The advantage to use the spatiotemporal blocks is the robustness to noise (NI) and to movement in the background (MB). The disadvantage is that the detection is less precise because only blocks are detected.

**Cluster-wise approaches:** Bhaskar et al. [147] divide first the image in clusters which are generated using a color clustering mechanism of the nearest neighbor approach. So, each cluster contains pixels than have similar features in the HSV space color. Then, the MOG is applied on these clusters to obtain cluster of pixels classified as background or foreground. Results presented by Bhaskar et al. [147] show that the cluster-wise MOG gives less false alarms. Instead of the block-wise approach, the foreground detection is obtained with a pixel-wise precision.

### 3.8. Feature Type

There are several features that are used in the literature rather than the RGB space used by Stauffer and Grimson [1]. These features can be classified in seven categories: color features, edge features, texture features, stereo features, spatial features, motion features, and video features.

**Color features:** In the original MOG, Stauffer and Grimson [1] used the RGB components but these components are very sensible to illumination changes (TD, LS) and shadowing (S). Furthermore, these components aren’t independent. So, some authors prefer to use other color space which are more robust to these critical situations and have independent components. These color spaces are the following ones: Normalized RGB [115, 116], YUV [88, 100, 117], HSV [100], HSI [148], Luv [101], Improved HLS Color Space [118]. Kristensen et al. [119] have studied the influence of the seven color spaces. The YCbCr is found to be best in terms of noise, due to numeric stability and an independent brightness channel. Ribeiro et al. [120] have made a similar study which confirms it.

**Edge features:** In addition of the color features, Javed et al. [80] use the gradient as feature to deal with local sudden illumination changes (LS) and the ghosts lefted in (MBO) and (WFO). Another example can be found in Lindstrom et al. [85] which uses only the gradient feature. The comparison with the intensity, the RGB space and the log-transformed output from a Prewitt edge detector show that the results are better in gray and RGB space because these features have less problems with false positives resulting in less noisy segmentation. It confirms the conclusion made in [149] that the gradient feature must be used in addition of the color features but not alone. In [67], Jain et al. use subpixel edges comparing this feature with the intensity, the gradient and the edges. Their results show that the subpixel edges are more robust in the case of camera adjustments like automatic gain control (CA) and sudden illumination change (MB).

**Texture features:** Hampapur et al. [81] use texture features to handle sudden illumination changes (LS). The original MOG method generates large areas of false positive foreground when there are LS. The basic idea is that the texture in the false positive foreground areas which is caused by lighting changes should be similar to the texture in the background. The results show robustness to LS.

**Stereo features:** In [89], Harville et al. propose to use the color space RGB and the disparity. In the same idea, Harville et al. [88] use the color space RGB but replace the disparity by the depth. The interest is that the two features can help each other in case of camouflage (C). One disadvantage of the stereo feature is the need of at least two cameras. Furthermore, the stereo features are not available in background areas which do not have sufficient texture to measure disparity with high confidence. In this case, only the color feature is used. Recently, Silvestre [91] alleviates the constraints of two cameras using a Time-Of-Flight (TOF) camera which is a recent technology for the acquisition in real-time of depth and intensity images.

**Spatial Features:** In addition to the color features (Luv), Yang and Hsu [101] use the spatial coordinate $(x, y)$ as features. As seen in the Section 3.5, this method is more robust in the case of MB. In the same idea, Dickinson et al. [90] use the spatial coordinate features $(x, y)$ too but with the color features (YUV). The interest of spatial features is developed in [150].

**Motion Features:** Gao et al. [41] propose a Spatial-Kinetic Mixture of Gaussians model (SKMGM). A four
dimensional feature vector is taken to describe the state of each pixel: \((x_i, y_i)\) which denotes the pixel’s coordinate and \((x_i, y_i)\) which denotes the corresponding motion speed values. Results [41] show robustness to dynamic backgrounds (MB), quick illumination changes (LS) and noise (NI).

**Video Features:** Chen et al. [61] use video features to model the background directly from compressed video. The proposed approach utilizes the information exploited from DCT coefficients at block level to construct accurate background models at pixel level. An advantage of this block based approach in the DCT domain is a representation of a pixel block in a very compact form. This method is more robust to noise (NI) and local sudden illumination changes (LS) because the block level introduces spatial information. Furthermore, the use of the YCbCr gives more robustness to shadows (S).

The intrinsic model improvements attempt to be more rigorous in the statistical sense like in the foreground detection or to introduce spatial and temporal constraints. In the following Section, we survey the other improvements which introduce these constraints too by adding one or more external process and so we have called them extrinsic model improvements. These kinds of improvements have been used in [151].

### 4. EXTRINSIC MODEL IMPROVEMENTS

The efficacy and robustness can be improved by using the knowledge of temporal and spatial information in the external strategies. In the literature, different approaches can be found by using:

**Markov Random Fields:** Kumar and Sengupta [69] propose to enforce the temporal contiguity in pixel classification by the use of simple Markov sequence models and to capture the spatial contiguity by MRF models. Results [69] show that the proposed algorithm is able to handle with local illumination changes corresponding to shadows and highlights due to Automatic Exposure Correction (CA). False positive and false negative detections are less than in the original MOG. Although the foreground detection is improved by the MRF, some objects may still be connected incorrectly and the boundaries of object are not accurate. To solve this problem, Zhou and Zhang [152] propose to use a level set method to handle topological merging and splitting. This method gives precise boundaries and good results on real images but is computationally expensive. Schindler et al. [153] present a MRF formulation of the labeling task too but only takes into account the spatial contiguity. The advantage of the proposed algorithm is that it is simple enough to be optimized in real-time.

**Hierarchical approaches:** As sharp changes (LS) cannot be detected on the pixel level alone, Sun and Yuan [76] employ state models of different scales. A MOG of the highest scale is used to detect sharp changes in the overall scene, which is built by extracting the global mean vector of each frame. The MOGs of intermediate scales are constructed likewise on a part of each frame to detect sharp changes in the partial scene, while pixel-wise MOGs of the lowest scale are implemented as usual. If a MOG of the upper scale has K states, each state holds a version of MOGs of the lower scale in the same location. Hence, all state models are organized in a hierarchical mode. In the same idea, Zhou et al. [154] use image pyramid. The motivation is that the peak noises and the small movement of the scene are eliminated in the low resolution image. By inferring the corresponding pixel states at different resolution level, a better background label image is computed. Park et al. [124] use a hierarchical data structure based on a regularly decomposed region quadtree for decrease the processing time. The hierarchical MOG is 5 to 10 faster than the Stauffer’s implementation. Hung et al. [102] propose a method that combines pixel-based and block-based approaches into a single framework. An efficient hierarchical background is built by considering that these two approaches are complementary to each other. Indeed, the hierarchical MOG proposed is more robust in the case of movement in the background (MB).

**Multi-level approaches:** Javed et al. [80] propose an algorithm which process at these three distinct levels. At the pixel level, two background models using the MOG are used: one for the color feature, the other one for the gradient feature. The edge and color information obtained from pixel level is used at the region level. The basic idea is that any foreground region that corresponds to an object will have high values of gradient at its boundaries. On the opposite, the boundary pixels did not contain significant edges in the cases of local sudden illumination changes (LS) and the ghosts leaved in (MBO) and (WFO). So, these edges are eliminated in the region level where foreground pixels obtained from the color feature are grouped into regions and gradient feature is then used to make inferences about the validity of these regions. The maintenance of the pixels is made following the results of the region level. Finally frame level analysis is performed to detect global sudden illumination changes (LS). Indeed, if more than 50 percent of the pixels are detected as foreground ones, the results obtained with the color feature are ignored at the frame level and then only gradient information is used. When the frame level model is active, the region level is not applied because the color feature is supposed to be completely unreliable at this point. Furthermore, the initialization of background model is possible with moving objects present in the scene but not more than 50%. So, the method may be not very robust to the bootstrapping (B). Another similar approach is presented by Zang and Klette [155] and consists in three levels also. In the pixel level, the background is model by the MOG only for the color feature in the RGB space. Then, the frame level is used in second and it is defined by two frame differences computed between the frame \(I_{t-1}\) and \(I_{t+1}\) corresponding to \(D_{t-1}\) and the frames \(I_{t-1}\) and \(I_{t+1}\) corresponding to \(D_{t}\). Pixels being identical in all three frames are in the foreground region of frame. So, Zang and Klette [155] ensure that these pixels are also in the foreground mask obtained from the pixel process by the following formula: \(Mask = Mask + (D_{t-1} \land D_{t})\). The region level comes in last position and its objective is to eliminate noise or small holes. For this, a test on the percentage of identical pixels in a 5*5 windows is made: 1)
If there are less than half the number of foreground points surrounding the central pixel, and these points are not connected, then this central foreground pixel is removed from foreground mask. 2) If there are more than half the number of foreground points surrounding it, and these points are connected, then this central point is confirmed to be a foreground pixel. Results show that the foreground detection gives more homogenous region than the MOG only. In [77], the pixel level consists in an improved MOG to get the better adaptability to the slow change of background. On the region level, the moving pixels are grouped to get moving region. Region level processing provides the solution for the slow and fast changes of the dynamic environments. Finally, frame level analysis is performed to detect the global illumination changes and re-initialize the background model. Cristani et al. [78] propose to use only the pixel level and the region level. At the pixel level, the TAPPMOG model proposed by Harville et al. [42] is used. Then, the pixel level information is modulated with the region level information resulting in a variation of the adaptiveness speed of the background modeling system driven by region-based reasoning. This approach is robust to sudden illumination changes (LS). Cristani et al [43] proposed a S-TAPPMOG model which perform the previous scheme using a spatial sampling mechanism. The maintenance is made at the region level. Cristani et al. [92] used a joint pixel-region analysis technique called ASTNA and which is able to automatically select sampling rate with which pixels in different areas are checked out, while adapting the size of the neighborhood region considered.

Multiple backgrounds: Su and Hu [74] use two background models to handle gradual and sudden changes (TD, LS). The first one is the short-term model which represents the change in background over a short time. It updates by the result of background MOG. Pixels belong to background will be used to update the short-term model. It is not desired when the foreground object become a part of background. The second one is the long-term model which represents the change in background over a long time. Each pixel will be used to update the long-term model. It is not desired when foreground objects move slowly or when some stationary variations happened in background, but it overcomes the drawback of the short-term model. Then, the two models are combined to get a better background model. When the mean of short-term model exceeds the mean of long-term model 2.5 standard deviation, the present background model will be replaced by long-term model. Otherwise, the short-term model is used. The threshold, 2.5 standard deviation, can be adjusted by the case of application. The sensitivity of the background model becomes higher by decreasing the threshold value. This scheme is improved by the same authors in [75]. Another approach developed by Porikli [156] use two backgrounds too, i.e a short and a long term ones, to detect temporarily static regions in video surveillance scene.

Graph cuts: Sun et al. [100] use a dynamic graph cut algorithm. First, an energy function is defined integrating a better likelihood term, a shadow elimination term and a contrast term. Exploiting the fact that there is minimal change between consecutive frames and corresponding MRF, Sun et al. [100] alleviate the constraints to construct and cut a new graph for each frame which is a conventionally slow step and uses the dynamic graph cut algorithm proposed by Kohli and Torr [157] to speed up foreground detection significantly. Results [100] show robustness to (MB) and (S).

Multi-layer approach: Yang et al. [39] use a two-layer Gaussian mixture model (TLGMM) of dynamic scenes for moving object detection. The first layer permits to deals with gradually changing pixels specially. The second layer focuses on those pixels changing significantly and irregularly. TLGMM can represent dynamic scenes more accurately and effectively. Additionally, a long term and short term variance are taken into account to alleviate the transparent problems faced by pixel-based methods. Results [39] show better performance in case of waving trees and water rippling (MB). Another approach developed by Porikli and Tuzel [158] models each pixel as layers of 3D multivariate Gaussians. Each layer corresponds to a different appearance of the pixel in the RGB space. The number of layers required to represent a pixel is not known beforehand so background is initialized with more layers than needed (3 to five layers). Using confidence scores, significant layers are retained for each pixel. At each update, at most one layer is updated with the current observation. This assures the minimum overlap over layers.

A different Features-Cameras strategies: Stauffer and Grimson [1] applied the MOG with one camera and color features. Different other strategies can be found in term of number of cameras and features used:

Multiple features - One Camera: The idea is to compensate the failure due to one color features by another color features (i.e intensity or color space) or another type of features (i.e edge features, stereo features and spatial features). For example, Ellis et al. [121] propose an original approach based on the intensity component and the color component. The interest is that the intensity feature is sensitive to both foreground and illumination changes and that the color features respond only to foreground. For the use of other features than the intensity features in addition to the color features, the advantage is related to the characteristic of the features used as seen in the Section 3.7.

One feature – Multiple Cameras: Nadimi and Bhanu [122] use different sensors but only the color features are used. Indeed, the sensors are visible spectrum cameras and the RGB components are used as features but the cameras have different spectral sensitivities. Each camera is looking at the same phenomenon in the scene and is providing independent measurements. Hence, camera configuration can be thought of as either cooperative or competitive in a fusion strategy scheme. Results [122] indicate that a AND strategy is a viable fusion strategy for relatively small moving objects at intermediate distances.

Multiple features – Multiple cameras: Nadimi and Bhanu [72] use the IR feature in addition of the RGB ones and so use two different cameras (IR, visible) The interest is handle adverse illumination conditions such as sudden illumination changes (LS) or lack of illumination in a scene.
Indeed, the IR feature is more reliable in this challenge than the RGB features. Results [72] show that this method maintains high detection rates under a variety of critical illumination conditions (TD, LS) and visible camera failure. In the same idea, Conaire et al. [73] use thermal infrared video and visible spectrum video in the context of video surveillance and models the background for each features (IR and LUV) with the MOG model proposed by [87]. Results show that this method is more robust to changing lighting conditions (TD, LS) and camouflage (C).

By using tracking feedback: Harville et al. [42] propose to use two types of feedback based on the classification obtained from the tracking process. A positive feedback enhances correct foreground segmentations and a negative feedback adjusts the pixel-level background model in order to prevent the re-occurrence of detected foreground mistakes. The feedback interface between the MOG and the higher levels i.e the tracker consists of two bitmaps. One represents pixels where positive feedback should be applied and the other represents pixels where negative feedback should be applied. The precise goal of positive feedback is to prevent incorporation of correctly foreground pixels into the background in the case of sleeping foreground objects (SFO). So, for pixels that are classified by the tracker as foreground pixels too, the learning rate is set to zero. The negative feedback can be viewed as a system that correct pixels classified as foreground by the MOG but as background by the tracker. Another approach developed by Taycher et al. [113] consists in a statistically consistent method for incorporating feedback from high-level motion model to modify adaptation behavior. This approach is based on formulating the background maintenance problem as inference in a continuous state Hidden Markov Model, and combining it with a similarly formulated object tracker in a multichain graphical model framework. The idea is that the background model should not be adapted to pixels that the tracking system predicts to be generated by the foreground objects, and visa versa, pixels that are predicted to belong to background should not be considered by the tracker. Results are better in the case of foreground objects that remained stationary for extended periods of time (SFO).

Post-processing: Standard post-processing is generally applied by using morphological operators, connected component analysis and minin area filtering. Turdu and Erdogan [159] reduce the fragmentation of foreground objects using a hysteresis thresholding method. An evaluation of the MOG with post-processing is presented in [160].

We can remark that the original MOG used in certain extrinsic model improvements can be replaced by one of the improved intrinsic MOG seen in the Section 2. All the intrinsic and extrinsic model improvements concern the quality of the foreground detection but there is another manner to improve the original MOG which consist in reducing the computation time.

5. REDUCING THE COMPUTATION TIME

The algorithm presented by Stauffer and Grimson [1] can be performed in term of time consuming. Different strategies can be found in the literature:

**Region of interest:** To reduce computation time, Atrey et al. [125] propose to use an Experiential Sampling technique. The goal is to apply only the MOG on the region of interest and thus restrict the computational efforts in these regions. Indeed, this region of interest in general is present only in a small part of the image, therefore, the attention should only be in these regions. Results presented by Atrey et al. [125] show a significant gain in processing speed with a minor loss in accuracy.

**A variable adaptation rate:** Another way to reduce the computation time is the use a variable adaptation rate in the maintenance. The goal is to update only when it needed. Indeed, Stauffer and Grimson [1] make the maintenance every frame but in the case of no significant changes pixels doesn’t need to be updated at every frame. In [24], Porikli et al. propose to adapt the time period of the maintenance mechanism following the illumination change score. The idea is that no maintenance is needed if no illumination change is detected and a quick maintenance is necessary otherwise. In the same idea, Magee [126] use a variable adaptation frame rate following the pixel’s activity, which improves temporal history storage for slow changing pixels while running at high adaptation rates for less stable pixels.

**Hardware implementations:** The large amount of calculations due to pixel-wise aspect of the MOG could often only achieve a low frame rate far from real-time requirements on computers. To solve this problem, Jiang et al. [127] propose a hardware accelerator with a dedicated architecture which can address the computation and the memory bandwidth demand. This architecture provides a calculation capacity allowing real-time processing of relatively large images 1024×1024 in RGB at a frame rate of 38 fps. To reduce the large memory required for storing Gaussian parameters, a specific memory scheme is proposed. Furthermore, a test benchmark based on an FPGA platform has been developed, capable of real time evaluation of the system performance as well as parameter properties. Another architecture proposed by Appiah et al. [128] use a camcorder, a FPGA single-chip, four blocks of RAM and a display unit. The system performs 209fps for 640×480 frame size and 145fps for 768×576 frame size in RGB.

**Switching the background model:** Zuo et al. [129] propose a switching based background modeling approach called MSBM. Regions are classified as regions with high or low complexity using an entropy measure. For background regions with high complexity of pixels value distribution, the MOG model is used to guarantee the accuracy of moving object detection. Otherwise a running average is applied to reduce the computational load. Results [129] show that this approach possesses almost the same detection accuracy and much higher image processing frame rate than MOG model only. Another approach proposed by Liang et al. [161] use a mean shift algorithm which classifies the background pixels as single mode or multiple mode pixels so that the single-mode pixel values are updated with a HIR filter, while the multi-mode pixel values are modeled by the MOG. This approach has a faster speed than the original MOG.

**Space sampling:** The meaning of the space sampling is to lower the resolution of the image and was proposed in [131]. For example, take a 320*240 image, its resolution can
be reduced by 1/4. This image is used to execute the MOG method and so the computational load is further lowered. When the binary mask is obtained, a zoom is made to come back to 320*240 using interpolation method. The speed using the space sampling method is 1.65 times, to 2.51 times faster than the original MOG.

All the precedent 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.

6. ENHANCING THE DETECTION WITH ANOTHER SEGMENTATION METHOD

Some authors propose to enhance the performance of the MOG using another method of segmentation:

- **Statistical Background Disturbance Technique**: Gunn et al. [123] combine the original MOG with the statistical background disturbance technique developed by Horprasert et al. [162]. This second model uses the chrominance distortion and the brightness distortion and is more robust in the context of presence of shadows (S) and highlights particularly in indoor scenes. The idea proposed by Gunn et al. [123] consists in the following: The MOG forms the basis of the model and the statistical background disturbance technique is used to force the MOG to expand the background distributions only when a pixel is labeled as a foreground pixel and the background disturbance technique disagrees. Results [123] show that the combination is more robust in the case of shadows and highlights in indoor and outdoor scenes.

- **Color Segmentation**: Fang et al. [145] combine the foreground detection obtained by a block-wise MOG with a color image segmentation [163] using a projecting method [164]. The projection operation consists in estimating every homogeneous region after color image segmentation. The region is validated as a moving region if most of the regions belong to moving region by background model algorithm. Otherwise the region is background. Results [145] show that the combination is more robust in the case of camouflage (CS6) than the original MOG.

- **Optical Flow and Temporal Subtraction**: Zhou et al. [12] associate the original MOG with optical flow and temporal differencing. The idea is to obtain the foreground object using the MOG and the moving objects using the optical flow and the temporal differencing. Temporal differencing is considered as a support technique of the optical flow which presents the inaccuracy to estimate the motion boundaries due to the use of image gradients and fixed support regions. In the data fusion step, foreground objects are validated as moving objects if the amplitude and direction of the optical flow are within the ranges of consideration. Results [103] show that the method locates precisely boundaries. Furthermore, this approach has satisfactory performance on real image sequence but is computational expensive because of the use of optical flow.

- **Region-based motion detection**: Zhang and Chen [64] apply a SVM classifier for block-based motion detection. Each video frame is divided into 8x8 blocks, and then classified as background or foreground. Features are extracted from current video frame, previous video frame, and the video background. The statistical attributes of an image block are computed as features, such as mean, standard deviation, and correlation. Eight features from each block are used for training and testing. By applying the decision function, a block motion detection mask is obtained. Then, the integration of SVM and MOG is given simply by the intersection. These approach remove false classifications due to irregular background motion (MB).

We can remark that to enhance the results obtained with the original MOG used in [64,103,145] can be replaced by one of the intrinsic models improvements see in the Section 2.

7. PERFORMANCE EVALUATION

For the performance evaluation, we have chosen some typical algorithms i.e specifically ones which the authors used the Wallflower dataset to evaluate them. This dataset is the most used and consists in a set of image sequences where each sequence presents a different type of difficulty that a practical task may meet. The performance is evaluated against hand-segmented ground truth. Three terms are used in 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. 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.
- **Time of Day (TOD)** - The light in a room gradually changes from dark to bright. Then, a person enters the room and sits down.
- **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.
- **Waving Trees (WT)** - A tree is swaying and a person walks in front of the tree.
- **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.
- **Boostrapping (B)** - The image sequence shows a busy cafeteria and each frame contains people.
- **Foreground Aperture (FA)** - A person with uniformly colored shirt wakes up and begins to move slowly.

Table 5 and Fig. (3) group the experimental results found in the literature for the algorithms chosen which are: 1) The original algorithm: Stauffer and Grimson [1]. 2) Three intrinsic improvements: White et al. [138] which used a better setting for the learning rates using Particle Swarm Optimisation (See section 3.4), Wang et al. [70] 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 (See section 3.6) and Setiawan et al. [118] which used the IHLS space (See section 3.7). 3) Three extrinsic improvements: Schindler et al. [153] which used the MRFS to smooth the results spatially as seen in Section 4, Cristani et al. [43] which proposed the Spatial-Time Adaptive Per Pixel Mixture Of Gaussian called S-TAPPMOG and Cristani et al. [92] 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. In Fig. (1), we have represented the overall performance for the five first algorithms and in Fig. (2) for the seven algorithms but without the image sequences Moved Object, Time of Day and Light Switch. Fig. (1) and Fig. (2) are not intended to be a definitive ranking of these algorithms. Such a ranking is necessarily task-, sequence-, and application dependent.

From Table 5, we can see that the original MOG gives the bigger total of error. A better setting of the learning rate $\alpha$ and the threshold T using the PSO [138] divides approximately by 2 the number of total errors. The use of the IHLS color space instead of the RGB [118] decreases a lot the number TE which becomes just under 10 0000. The improvement proposed by Wang et al. [70] gives the better results for the intrinsic improvements. For the extrinsic improvements, the best results are obtained by the MOG using the MRF proposed by Schindler et al. [153] followed by the S-TAPPMOG [43] and ASTNA [92]. 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. [153]. The use of IHLS [118] gives it best improvement for the image sequences Camouflage (C) and for the method proposed by Wang et al. [70], it is the image sequences Waving Trees (WT). In resume, this performance evaluation shows that taking into account spatial and temporal consistency improves the results in a significant way.

**CURRENT & FUTURE DEVELOPMENTS**

This paper attempts to provide a comprehensive survey of research on background modeling using Mixture of Gaussians for foreground detection and to provide some structural categories for the strategies described in over 150 papers. Thus, we have proposed a classification in two classes respectively called intrinsic and extrinsic model improvements. Strategies adding spatial and temporal information in the different steps or in added process have shown their abilities to improve the robustness of the original

---

**Fig. (1).** Overall performance.

**Fig. (2).** Overall performance without MO, TD and LS.
MOG model to critical situations. Cooperation with another segmentation 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 since the paper of Stauffer and Grimson [1], there is still work to be done and we believe that a systematic comparative evaluation of improvement proposed at each step must be made and thus determine the best combination of strategies. In this context, we encourage the evaluation using all the Wallflower dataset like in [70, 118, 138, 153]. For a better improvement and understanding of the MOG, we believe that a time series analysis like in [165] must be done and a reusable and pluggable software component must be developed like the one developed by Haque et al. [166]. Furthermore, two main investigations seem to be very promising:

### Table 5. Experimental Results Obtained by Different Improvements

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Problem Type</th>
<th>Error</th>
<th>Moved</th>
<th>Time of Light</th>
<th>Waving</th>
<th>Camouflag</th>
<th>Bootstrap</th>
<th>Foreground</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stauffer et al. [1]</td>
<td>False neg.</td>
<td>0</td>
<td>1008</td>
<td>1633</td>
<td>1323</td>
<td>398</td>
<td>1874</td>
<td>2442</td>
<td></td>
</tr>
<tr>
<td>MOG</td>
<td>False pos.</td>
<td>0</td>
<td>20</td>
<td>14169</td>
<td>341</td>
<td>3098</td>
<td>217</td>
<td>530</td>
<td>27053</td>
</tr>
<tr>
<td>White et al. [138]</td>
<td>False neg.</td>
<td>0</td>
<td>807</td>
<td>1716</td>
<td>43</td>
<td>2386</td>
<td>1551</td>
<td>2392</td>
<td></td>
</tr>
<tr>
<td>MOG with PSO</td>
<td>False pos.</td>
<td>0</td>
<td>6</td>
<td>772</td>
<td>1689</td>
<td>1463</td>
<td>519</td>
<td>572</td>
<td>13916</td>
</tr>
<tr>
<td>Setiawan et al. [118]</td>
<td>False neg.</td>
<td>0</td>
<td>379</td>
<td>1146</td>
<td>31</td>
<td>188</td>
<td>1647</td>
<td>2327</td>
<td></td>
</tr>
<tr>
<td>MOG IILS</td>
<td>False pos.</td>
<td>0</td>
<td>99</td>
<td>2298</td>
<td>270</td>
<td>467</td>
<td>333</td>
<td>554</td>
<td>9739</td>
</tr>
<tr>
<td>Wang et al. [70]</td>
<td>False neg.</td>
<td>0</td>
<td>597</td>
<td>1481</td>
<td>44</td>
<td>106</td>
<td>1176</td>
<td>1274</td>
<td></td>
</tr>
<tr>
<td>Improved MOG - FD</td>
<td>False pos.</td>
<td>0</td>
<td>358</td>
<td>669</td>
<td>288</td>
<td>413</td>
<td>134</td>
<td>541</td>
<td>7081</td>
</tr>
<tr>
<td>Schindler et al. [153]</td>
<td>False neg.</td>
<td>0</td>
<td>47</td>
<td>204</td>
<td>15</td>
<td>16</td>
<td>1060</td>
<td>34</td>
<td>3808</td>
</tr>
<tr>
<td>MOG with MRF</td>
<td>False pos.</td>
<td>0</td>
<td>402</td>
<td>546</td>
<td>311</td>
<td>467</td>
<td>102</td>
<td>604</td>
<td></td>
</tr>
<tr>
<td>Cristani et al. [43]</td>
<td>False neg.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>153</td>
<td>643</td>
<td>1414</td>
<td>1912</td>
<td></td>
</tr>
<tr>
<td>S-TAPM0G</td>
<td>False pos.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1152</td>
<td>1382</td>
<td>811</td>
<td>377</td>
<td>7844</td>
</tr>
<tr>
<td>Cristani et al. [92]</td>
<td>False neg.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>253</td>
<td>823</td>
<td>2349</td>
<td>1900</td>
<td></td>
</tr>
<tr>
<td>ASTNA</td>
<td>False pos.</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>100</td>
<td>1173</td>
<td>73</td>
<td>360</td>
<td>7031</td>
</tr>
</tbody>
</table>

![Fig. (3). Wallflower datasets results.](image-url)
REFERENCES


[28] ATON Dataset: http://icpr.ucsd.edu/atont/sond/

[29] OTCVS Dataset: http://www.waseda-it.edu/otcvs-bench/

[30] VSSN Dataset: http://mimc36.informatik.uni-augsburg.de/VSSN06_06AC/

[31] PETS Dataset: http://pets2006.net/


subpixel edges. 
 Modeling based on tree dynamic programming. 

Rao N, Di H, Xu G. Joint correspondence and background Proc Electronic Imaging, Conf on Visual Communications and Image, models for real-time background maintenance. 

Achkar F, Amer A. Hysteresis -based selective Gaussian mixture dynamic video backgrounds. 


Xu L. Robust detection and tracking of multiple objects in cluttered 168. 


Lee D. Improved adaptive mixture learning for robust video background modeling. IAPR Workshop on Machine Vision for Application (MVA 2002), Nara, Japan, December 2002; 443-446.


Ren Y. Chua C. Ho Y. Motion detection with non-stationary background. ICIAP 2002, 2002; 78-83.


Bhaskar H. Mihaylova L. Maskell S. Automatic target detection based on background modeling using adaptive cluster density estimation. LNCS from the 3rd German Workshop on Sensor Data Fusion: Trends, Solutions, Applications, Universität Bremen, Germany, September 2007.


Ramadan S. Using time series analysis to visualize and evaluate background subtraction results in computer vision applications, Master thesis, University of Maryland, Spring 2006.

http://www.mahfuzulhaque.com/research_cur.html


