Machine Vision and Applications (2014) 25:1145–1162
DOI 10.1007/s00138-013-0568-z

SPECIAL ISSUE PAPER

Background subtraction using finite mixtures of asymmetric Gaussian distributions and shadow detection

Tarek Elguebaly · Nizar Bouguila

Received: 29 December 2012 / Revised: 14 July 2013 / Accepted: 7 October 2013 / Published online: 24 October 2013

Abstract Foreground segmentation of moving regions in image sequences is a fundamental step in many vision systems including automated video surveillance, human-machine interface, and optical motion capture. Many models have been introduced to deal with the problems of modeling the background and detecting the moving objects in the scene. One of the successful solutions to these problems is the use of the well-known adaptive Gaussian mixture model. However, this method suffers from some drawbacks. Modeling the background using the Gaussian mixture implies the assumption that the background and foreground distributions are Gaussians which is not always the case for most environments. In addition, it is unable to distinguish between moving shadows and moving objects. In this paper, we try to overcome these problem using a mixture of asymmetric Gaussians to enhance the robustness and flexibility of mixture modeling, and a shadow detection scheme to remove unwanted shadows from the scene. Furthermore, we apply this method to real image sequences of both indoor and outdoor scenes. The results of comparing our method to different state of the art background subtraction methods show the efficiency of our model for real-time segmentation.

Keywords Foreground segmentation · Shadow detection · Mixture models · Asymmetric Gaussian · Maximum likelihood estimate · Expectation–maximization · Minimum message length

T. Elguebaly
Electrical and Computer Engineering (ECE),
Concordia University, Montreal, QC H3G 2W1, Canada
e-mail: t_elgue@encs.concordia.ca

N. Bouguila (✉)
Concordia Institute for Information Systems Engineering (CIISE),
Concordia University, Montreal, QC H3G 2W1, Canada
e-mail: nizar.bouguila@concordia.ca

1 Introduction

Over the last decade, automatic segmentation of foreground from background in video sequences has attracted lots of attention in computer vision [1–3]. Foreground segmentation is often used as the primary step in video surveillance [4–6], optical motion capture [7,8], and multimedia processing [9,10] to model the background and to detect the moving objects in the scene. Background subtraction involves the extraction of a background image which does not include any moving object, reference image, then subtracting each new frame from this image and thresholding the result to highlight regions of non-stationary objects. Normally, video surveillance systems can be employed in two kinds of environments: controlled and uncontrolled. Monitoring systems in controlled or indoor environments (i.e., airports, warehouses, and production plants) are easier to implement as they do not depend on weather changes. Uncontrolled environment is used to refer to outdoor scenes where illumination and temperature changes occur frequently, and where various atmospheric conditions can be observed. Normally, when developing background subtraction algorithms, there are two major problems that must be taken into consideration, namely, robustness and adaptation. These methods should be robust to illumination and weather changes, as well as able to detect addition, occlusion, and removal of objects in the scene. To take into account these problems of robustness and adaptation, many background modeling methods have been developed (a complete detailed survey can be found in [11]).

In the past, computational barriers have limited the complexity of real-time video processing applications. As a consequence, most systems were either too slow to be practical, or succeeded by restricting themselves to very controlled situations. Recently, faster computers have enabled researchers to consider more complex, robust models for real-time analysis of streaming data. These new methods allow researchers to begin modeling real-world processes under varying conditions. Most recent methods assume that the images of the scene without the intruding objects exhibit some regular behavior that can be well described by a statistical model. If we have a statistical model of the scene, an intruding object can be detected by spotting the parts of the image that do not fit the model. In the majority of these methods, a common bottom-up approach has been applied to construct a probability density function for each pixel separately. Its idea is to segment the foreground moving objects by constructing over time a mixture model for each pixel and deciding, in a new input frame, whether the pixel belongs to the foreground or the background [12,13]. Among the vast amount of approaches that have been proposed to accomplish this task, adaptive Gaussian mixture models (GMMs) [12,14] have proven their outstanding suitability in several computer vision application [15] and particularly in the surveillance domain because of their ability to achieve many of the requirements of a surveillance system, e.g., adaptability and multimodality, in real time with low memory requirements. GMMs model the history of each pixel by a mixture of K Gaussian distributions. In [16], the authors implemented a pixel-wise EM framework for detection of vehicles by attempting to explicitly classify the pixel values into three separate predetermined distributions corresponding to the road color, the shadow color, and colors corresponding to vehicles. Stauffer et al. [12] generalized this idea by implementing online K-means approximation algorithm for modeling each pixel using a mixture of K Gaussians, where K was chosen in the range (3–5) depending on the computational power of the machine. In [14] the use of a negative prior evidence was introduced to discard the components that are not supported by the data, therefore being able to automatically select the number of components of the mixture used for each pixel. In [17] the use of an adaptive learning rate calculated for each Gaussian at every frame was proposed which led to an improved segmentation performance compared to the standard method. However, these methods have some drawbacks. Modeling the background using the GMM implies the assumption that the background and foreground distributions are Gaussians which is not always the case as argued by [18]. Figure 1 shows the probability density function of a pixel throughout a video. From this figure we can notice that the distribution is not symmetrical. Applying the GMM, we can observe its inefficiency in modeling the data. To overcome these problems, some researchers have shown that the generalized Gaussian mixture (GGM) can be a good choice to model non-Gaussian data [19–21]. Compared to the Gaussian distribution (GD), the generalized Gaussian distribution (GGD) has one more parameter λ that controls the tail of the distribution: the larger the value of λ is, the flatter is the distribution; the smaller λ is, the more peaked is the distribution. Despite the higher flexibility that GGD offers, it is still a symmetric distribution inappropriate to model non-symmetrical data. From Fig. 1, we can recognize that the GGM is not suitable in modeling our data. In this article, we suggest the use of the asymmetric Gaussian distribution (AGD) capable of modeling asymmetrical data. The AGD uses two variance parameters for left and right parts of the distribution, which allow it to change its shape to conclude the asymmetry. As shown in Fig. 1, we can notice that the asymmetric Gaussian mixture (AGM) was able to accurately model the data.

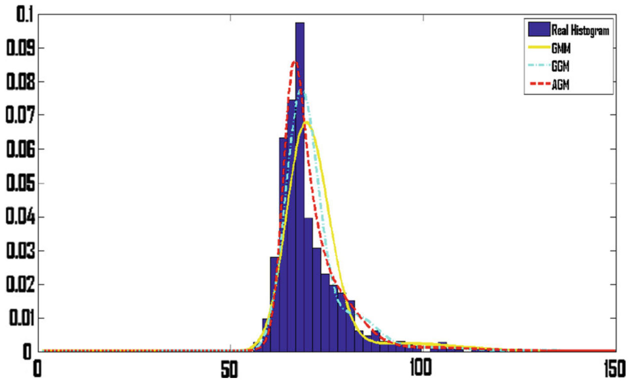


Fig. 1 Probability density function of a pixel throughout a video sequence

An important part of the mixture modeling problem concerns learning the model parameters and determining the number of consistent components (M) which best describes the data. For this purpose, many approaches have been suggested. The vast majority of these approaches can be classified, from a computational point of view, into two classes: deterministic and stochastic methods. Deterministic methods estimate the model parameters for different range of M then choose the best value that maximize a model selection criteria such as the Akaike's information criterion (AIC) [22], the minimum description length (MDL) [23] and the Laplace empirical criterion (LEC) [24]. Stochastic methods such as Markov chain Monte Carlo (MCMC) can be used to sample from the full a posteriori distribution with M considered unknown [25]. Despite their formal appeal, MCMC methods are too computationally demanding, therefore cannot be applied for online applications such as foreground segmentation. For this reason, we are interested in deterministic approaches. In our proposed method, we use K-means algorithm to initialize the AGM parameters and successfully solve the initialization problem. The number of mixture components is automatically determined by implementing the minimum message length (MML) criterion [26] into the expectation–maximization (EM) algorithm. Therefore, the method can integrate simultaneously parameter estimation and model selection in a single algorithm, thus it is totally unsupervised.

Shadows, areas where direct light from a light source cannot reach due to obstruction by different objects, are an ever-present aspect of color images. As a result of the difference between the light intensity reaching a shaded region and a directly lit region, shadows are often characterized by conspicuous strong brightness gradients. In outdoor scenes, the change between shadow and non-shadow regions is not entirely a brightness difference, but a color one as well. This property makes shadow detection task a highly problematic one in a number of different fields. Recently, there have been few studies concerning shadow removal; however, the best performing methods still require user interaction with image sequences to perform optimally. In our moving shadow detection algorithm, we implement a method compatible with the RGB color model and able to use our mixture model. It is noteworthy that the proposed learning approach is completely different from recent efforts published for instance in [21,27]. In fact, [27] proposed the use of the asymmetric Gaussian with different parametrization for pattern recognition applications. Furthermore, this method used the two-dimensional AGM for data classification when the number of components is known in advance. In [21] a Bayesian non-parametric approach based on infinite GGM was developed for pedestrian detection and foreground segmentation.

The rest of this paper is organized as follows. Section 2 describes the AGM model and its learning algorithm. In Sect. 3, we assess the performance of the new model with shadow removal scheme for foreground segmentation; while comparing it to other models. Our last section is devoted to the conclusion and some perspectives.

2 Finite AGM model

Formally we say that a d-dimensional random variable $\mathbf{X} = [X_1, \ldots, X_d]^T$ follows a M components mixture distribution if its probability function can be written in the following form:

$$p(\mathbf{X}|\Theta) = \sum_{j=1}^{M} p_j\, p(\mathbf{X}|\xi_j) \tag{1}$$

where

- ξ_j is the set of parameters of the component j,
- p_j are the mixing proportions which must be positive and sum to one,
- $\Theta = \{p_1, \ldots, p_M, \xi_1, \ldots, \xi_M\}$ is the complete set of parameters fully characterizing the mixture,
- $M \geq 1$ is number of components in the mixture.

For the AGM, each component density $p(\mathbf{X}|\xi_j)$ is an AGD given by:

$$p(\mathbf{X}|\xi_j) = \prod_{k=1}^{d} \sqrt{\frac{2}{\pi}} \frac{1}{(\sigma_{l_j} + \sigma_{r_j})} \times \begin{cases} \exp\left[-\frac{(X_k-\mu_{jk})^2}{2\sigma_{l_{jk}}^2}\right] & \text{if } X_k < \mu_{jk} \\ \exp\left[-\frac{(X_k-\mu_{jk})^2}{2\sigma_{r_{jk}}^2}\right] & \text{if } X_k \geq \mu_{jk} \end{cases} \tag{2}$$

where $\xi_j = (\boldsymbol{\mu}_j, \boldsymbol{\sigma}_{l_j}, \boldsymbol{\sigma}_{r_j})$ is the set of parameters of component j where $\boldsymbol{\mu}_j = (\mu_{j1} \ldots, \mu_{jd})$, $\boldsymbol{\sigma}_{\mathbf{l}_j} = (\sigma_{l_{j1}}, \ldots, \sigma_{l_{jd}})$, and $\boldsymbol{\sigma}_{\mathbf{r}_j} = (\sigma_{r_{j1}}, \ldots, \sigma_{r_{jd}})$ are the mean, the left standard deviation, and the right standard deviation of the d-dimensional AGD, respectively. The AGD is chosen to be able to fit, in analytically simple and realistic way, symmetric or non-symmetric data by the combination of the left and right variances.

Let $\mathcal{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ be a set of N independent and identically distributed vectors, assumed to arise from a finite AGM model with M components. Thus, it can be expressed as follows:

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{M} p(\mathbf{X}_i|\xi_j) p_j \tag{3}$$

where the set of parameters of the mixture with M classes is defined by $\Theta = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M, \boldsymbol{\sigma}_{l_1}, \ldots, \boldsymbol{\sigma}_{l_M}, \boldsymbol{\sigma}_{r_1}, \ldots, \boldsymbol{\sigma}_{r_M}, p_1, \ldots, p_M)$.

We introduce membership vectors, $Z_i = (Z_{i1}, \ldots, Z_{iM})$, one for each observation, whose role is to encode to which component the observation belongs. In other words, Z_{ij}, the unobserved or missing variable in each membership vector, equals 1 if $\mathbf{X}_i$ belongs to class j and 0, otherwise. The complete-data likelihood for this case is then:

$$p(\mathcal{X}, Z|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left(p(\mathbf{X}_i|\xi_j) p_j\right)^{Z_{ij}} \tag{4}$$

2.1 Maximum likelihood estimation of the mixture parameters

For the moment, we suppose that the number of mixture M is known. The maximum likelihood method consists of getting the mixture parameters that maximize the log-likelihood function given by:

$$L(\Theta, Z, \mathcal{X}) = \sum_{i=1}^{N} \sum_{j=1}^{M} Z_{ij} \log\left(p(\mathbf{X}_i|\xi_j) p_j\right) \tag{5}$$

by replacing each Z_{ij} by its expectation, defined as the posterior probability that the ith observation arises from the jth component of the mixture as follows:

$$\hat{Z}_{ij} = p(j|\mathbf{X}_i) = \frac{p(\mathbf{X}_i|\xi_j) p_j}{\sum_{j=1}^{M} p(\mathbf{X}_i|\xi_j) p_j} \tag{6}$$

Using Eq. 6, we can affect each observation to one of the M clusters. Now, using these expectations, we want to maximize the complete-data log-likelihood with respect to our model parameters. This can be done by taking the gradient of the log-likelihood with respect to p_j, $\boldsymbol{\mu}_j$, $\boldsymbol{\sigma}_{l_j}$, and $\boldsymbol{\sigma}_{r_j}$. When estimating p_j, we actually need to introduce Lagrange multiplier to ensure that the constraints $p_j > 0$ and $\sum_{j=1}^{M} p_j = 1$ are satisfied. Thus, the augmented log-likelihood function can be expressed by:

$$\Phi(\Theta, Z, \mathcal{X}, \Lambda) = \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{ij} \log\left(p(\mathbf{X}_i|\xi_j)p_j\right) + \Lambda\left(1 - \sum_{j=1}^{M} p_j\right) \tag{7}$$

where Λ is the Lagrange multiplier. Differentiating the augmented function with respect to p_j we get:

$$\hat{p}_j = \frac{1}{N}\sum_{i=1}^{N} p(j|\mathbf{X}_i) \tag{8}$$

Taking the gradient of the complete log-likelihood with respect to $\boldsymbol{\mu}_j$, $\boldsymbol{\sigma}_{l_j}$, and $\boldsymbol{\sigma}_{r_j}$, we obtain the following for $k = 1, \ldots, d$:

$$\hat{\mu}_{jk} = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} X_{ik}}{\sum_{i=1}^{N} \hat{Z}_{ij}} \tag{9}$$

$$\sum_{i=1, X_{ik} < \mu_{jk}}^{N} \frac{\hat{Z}_{ij}(X_{ik} - \mu_{jk})^2}{\sigma_{l_{jk}}^3} - \sum_{i=1}^{N} \frac{\hat{Z}_{ij}}{\sigma_{l_{jk}} + \sigma_{r_{jk}}} = 0 \tag{10}$$

$$\sum_{i=1, X_{ik} \geq \mu_{jk}}^{N} \frac{\hat{Z}_{ij}(X_{ik} - \mu_{jk})^2}{\sigma_{r_{jk}}^3} - \sum_{i=1}^{N} \frac{\hat{Z}_{ij}}{\sigma_{l_{jk}} + \sigma_{r_{jk}}} = 0 \tag{11}$$

We can notice that Eqs. 10 and 11 are non-linear, therefore we have decided to use the Newton–Raphson method for estimation:

$$\hat{\sigma}_{l_{jk}} \simeq \sigma_{l_{jk}} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk}}^2}\right)^{-1}\left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk}}}\right)\right] \tag{12}$$

$$\hat{\sigma}_{r_{jk}} \simeq \sigma_{r_{jk}} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk}}^2}\right)^{-1}\left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk}}}\right)\right] \tag{13}$$

where $\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma_{l_{jk}}^2}$, $\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \sigma_{l_{jk}}}$, $\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma_{r_{jk}}^2}$, and $\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \sigma_{r_{jk}}}$ are given in Appendix A.

2.2 Model selection using the minimum message length criterion

Different model selection methods have been introduced to estimate the number of components of a mixture. In this article, we are interested with deterministic approaches, especially MML. The MML approach is based on evaluating statistical models according to their ability to compress a message containing the data (minimum coding length criteria). High compression is obtained by forming good models of the data to be coded. For each model in the model space, the message includes two parts. The first part encodes the model, using only prior information about the model and no information about the data. The second part encodes only the data in a way that makes use of the model encoded in the first part. When applying the MML, the optimal number of classes of the mixture is obtained by minimizing the following function [26,28]:

$$MessLen \approx -\log(p(\Theta)) - L(\Theta, Z, \mathcal{X}) + \frac{1}{2}\log|F(\Theta)| + \frac{N_p}{2} - \frac{1}{2}\log(12) \tag{14}$$

where $p(\Theta)$ is the prior probability, $|F(\Theta)|$ is the determinant of the Fisher information matrix minus the log-likelihood of the mixture, and N_p is the number of parameters to be estimated and is equal to $M(3d + 1)$ in our case. In the following subsections, we give the derivation of both the prior probability $p(\Theta)$ and the determinant of the Fisher information matrix of minus the log-likelihood of the mixture $|F(\Theta)|$.

2.2.1 Derivation of the prior $p(\Theta)$

We specify a prior $p(\Theta)$ that expresses the lack of knowledge about the mixture parameters. It is reasonable to assume that the parameters of different components in the mixture are independent, since having knowledge about a parameter in one class does not provide any knowledge about the parameters of another class. Thus, we can assume that the mixture parameters are mutually independent, then:

$$p(\Theta) = p(P)\prod_{j=1}^{M} p(\boldsymbol{\mu}_j)p(\boldsymbol{\sigma}_{l_j})p(\boldsymbol{\sigma}_{r_j}) \tag{15}$$

where $P = (p_1, \ldots, p_M)$. In what follows, we will specify each of these priors separately. Starting with $p(P)$, we know that $P = (p_1, \ldots, p_M)$ is defined on the simplex $\{(p_1, \ldots, p_M) : \sum_{j=1}^{M} p_j = 1\}$. Then, a natural choice as a prior for this vector is the Dirichlet distribution

$$p(P) = \frac{\Gamma\left(\sum_{j=1}^{M} \eta_j\right)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \tag{16}$$

where $(\eta_1, \ldots, \eta_M)$ is the parameter vector of the Dirichlet distribution. When $\eta_1, \ldots, \eta_M = \eta = 1$ we get a uniform prior over the space $p_1 + \cdots + p_M = 1$. This prior is represented by

$$p(P) = (M-1)! \tag{17}$$

For $\boldsymbol{\mu}_j$, we take a uniform prior for each μ_{jk}. Each μ_{jk} is chosen to be uniform in the region $(\mu_{jk} - \sigma_{l_k} \leq \mu_{jk} \leq \mu_{jk} + \sigma_{r_k})$, then the prior for $\boldsymbol{\mu}_j$ is given by

$$p(\boldsymbol{\mu}_j) = \prod_{k=1}^{d} p(\mu_{jk}) = \prod_{k=1}^{d} \frac{1}{(\sigma_{l_k} + \sigma_{r_k})} \tag{18}$$

For both $\boldsymbol{\sigma}_{l_j}$ and $\boldsymbol{\sigma}_{r_j}$, knowing that $(0 \leq \sigma_{l_{jk}} \leq \sigma_{l_k})$ and $(0 \leq \sigma_{r_{jk}} \leq \sigma_{r_k})$, then a good choice of prior for $\sigma_{l_{jk}}$ and $\sigma_{r_{jk}}$ is the uniform distribution:

$$p(\boldsymbol{\sigma}_{l_j}) = \prod_{k=1}^{d} p(\sigma_{l_{jk}}) = \prod_{k=1}^{d} \frac{1}{\sigma_{l_k}} \tag{19}$$

$$p(\boldsymbol{\sigma}_{r_j}) = \prod_{k=1}^{d} p(\sigma_{r_{jk}}) = \prod_{k=1}^{d} \frac{1}{\sigma_{r_k}} \tag{20}$$

Finally, by replacing the priors of the parameter in Eq. 15 by each prior value in Eqs. 17, 18, 19, and 20 we get

$$p(\Theta) = (M-1)! \prod_{k=1}^{d} \frac{1}{\sigma_{l_k}^{M} \sigma_{r_k}^{M} (\sigma_{l_k} + \sigma_{r_k})^{M}} \tag{21}$$

2.2.2 Derivation of the determinant of the Fisher information matrix $|F(\Theta)|$

The Fisher information matrix is the expected value of the Hessian of minus the logarithm of the likelihood. It is difficult, in general, to obtain the expected Fisher information matrix of a mixture analytically. Therefore, we use the complete Fisher information matrix where its determinant is equal to the product of the determinant of the information matrix for each component times the determinant of the information matrix of P

$$|F(\Theta)| = |F(P)| \prod_{j=1}^{M} |F(\boldsymbol{\mu}_j)| |F(\boldsymbol{\sigma}_{l_j})| |F(\boldsymbol{\sigma}_{r_j})| \tag{22}$$

in which $|F(\boldsymbol{\mu}_j)|$, $|F(\boldsymbol{\sigma}_{l_j})|$, and $|F(\boldsymbol{\sigma}_{r_j})|$ are the Fisher information with regards to $\boldsymbol{\mu}_j, \boldsymbol{\sigma}_{l_j}$, and $\boldsymbol{\sigma}_{r_j}$, respectively, for the AGD that corresponds to component j in the mixture model. $|F(P)|$ is the Fisher information with regards to the mixing parameters vector that satisfy the requirement $\{\sum_{j=1}^{M} p_j = 1\}$. Consequently, it is possible to consider the generalized Bernoulli process with a series of trials, each of which has M possible outcomes labeled first cluster, second cluster,..., Mth cluster. Therefore, the number of trials of the jth cluster is a multinomial distribution of parameters $p_1, p_2, \ldots, p_M$. Then, the determinant of the Fisher information matrix is

$$|F(P)| = \frac{N^{M-1}}{\prod_{j=1}^{M} p_j} \tag{23}$$

For $|F(\boldsymbol{\mu}_j)|$, $|F(\boldsymbol{\sigma}_{l_j})|$, and $|F(\boldsymbol{\sigma}_{r_j})|$ let us consider the jth class $\mathcal{X}_j = (\mathbf{X}_l, \ldots, \mathbf{X}_{l+n_j-1})$ of the mixture as the data in class j after classifying all the data $\mathcal{X}$ using the maximum a posteriori probability defined by Eq. 6. Note that n_j is the number of data vectors belonging to the jth distribution. This given choice of the jth class allows us to simplify the notation without loss of generality. Then, the Hessian matrices when we consider the vectors $\boldsymbol{\mu}_j, \boldsymbol{\sigma}_{l_j}$, and $\boldsymbol{\sigma}_{r_j}$ are given by:

$$F(\boldsymbol{\mu}_j)_{k_1,k_2} = \frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \mu_{jk_1} \partial \mu_{jk_2}} \tag{24}$$

$$F(\boldsymbol{\sigma}_{l_j})_{k_1,k_2} = \frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk_1}} \partial \sigma_{l_{jk_2}}} \tag{25}$$

$$F(\boldsymbol{\sigma}_{r_j})_{k_1,k_2} = \frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk_1}} \partial \sigma_{r_{jk_2}}} \tag{26}$$

where $(k_1, k_2) \in (1, \ldots, d)$. Using Appendix B to compute the derivatives in Eqs. 24, 25, 26, we obtain

$$|F(\boldsymbol{\mu}_j)| = \prod_{k=1}^{d} \left[\sum_{i=l, X_{ik} < \mu_{jk}}^{l+n_j-1} \frac{1}{\sigma_{l_{jk}}^2} + \sum_{i=l, X_{ik} \geq \mu_{jk}}^{l+n_j-1} \frac{1}{\sigma_{r_{jk}}^2} \right] \quad \text{for} \quad j = 1, \ldots, M \tag{27}$$

$$|F(\boldsymbol{\sigma}_{l_j})| = \prod_{k=1}^{d} \left[\sum_{i=l}^{l+n_j-1} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})^2} - 3 \times \sum_{i=l, X_{ik} < \mu_{jk}}^{l+n_j-1} \frac{(X_{ik} - \mu_{jk})^2}{\sigma_{l_{jk}}^4} \right] \quad \text{for} \quad j = 1, \ldots, M \tag{28}$$

$$|F(\boldsymbol{\sigma}_{r_j})| = \prod_{k=1}^{d} \left[\sum_{i=l}^{l+n_j-1} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})^2} - 3 \times \sum_{i=l, X_{ik} \geq \mu_{jk}}^{l+n_j-1} \frac{(X_{ik} - \mu_{jk})^2}{\sigma_{r_{jk}}^4} \right] \quad \text{for} \quad j = 1, \ldots, M \tag{29}$$

2.3 The AGM model learning algorithm

In the following steps, we summarize the algorithm used for the AGM parameters estimation and model selection. Given a number of components, the mixture parameters are estimated iteratively using the EM algorithm:

```
Input: Data set $\mathcal{X}$ and $M_{\max}$
Output: $\Theta_{M^*}$ (the values of $\Theta$ when $M^*$ components are chosen) and $M^*$

Step 1: For $M = 1 : M_{max}$ do{
  1. Initialize the parameters.
  2. Repeat until convergence.
     (a) The E-step given by Eq. 6.
     (b) The M-step given by Eqs. 8, 9, 12, 13.
  3. Calculate the associated message length using Eq. (14).
}END FOR
Step 2: Select the model $M^*$ with the smallest message length.
```

To initialize the parameters, we used the K-means algorithm. Note that, we initialized both the left and right standard deviations with the standard deviation values obtained from the K-means. To detect the convergence of the EM, we stop the iterations when the difference of the log-likelihood from two successive iterations, ℓ and $\ell+1$, is smaller than a predefined threshold ϵ.

3 Background subtraction

In this section, we investigate the efficiency of the AGM algorithm for background subtraction. Our method can be divided into two main components: background modeling and shadows detection.

3.1 Adaptive AGM algorithm

Adaptive Gaussian mixture algorithms are widely applied for background subtraction. In [12], the authors presented an online learning of a GMM for each pixel in the video frames. Their idea was to model each pixel in the scene by a mixture of K Gaussian distributions, where K is taken in the range (3–5). Then, they ordered the K distributions based on the fitness value p_j/σ_j and used the first B distributions to model the background of the scene, where B is estimated as

$$B = \arg\min_b \sum_{j=1}^{b} p_j > T, \tag{30}$$

T is a measure of the minimum portion of the data that represents the background in the scene. Then, the foreground pixels were detected as any pixel that is more than 2.5 standard deviations away from any of the B distributions. For the $(\ell+1)$ frame, the first Gaussian component that matches the test value will be updated by the following equations:

$$\hat{p}_j^{\ell+1} = (1-\alpha)\hat{p}_j^{\ell} + \alpha\hat{p}(j|X^{\ell+1}) \tag{31}$$

$$\hat{\mu}_j^{\ell+1} = (1-\alpha)\hat{\mu}_j^{\ell} + \rho X^{\ell+1} \tag{32}$$

$$\hat{\Sigma}_j^{\ell+1} = (1-\alpha)\hat{\Sigma}_j^{\ell} + \rho\left(X^{\ell+1} - \hat{\mu}_j^{\ell+1}\right)\left(X^{\ell+1} - \hat{\mu}_j^{\ell+1}\right)^T \tag{33}$$

where $1/\alpha$ defines the time constant which determines change. $\hat{p}_j^{\ell+1}$, $\hat{\mu}_j^{\ell+1}$, and $\hat{\Sigma}_j^{\ell+1}$ are the estimated value of the weight, mean, and covariance of the component j of the mixture at the $(\ell+1)$ frame, respectively. Note that $\hat{p}(j|X^{\ell+1})$ is formulated as:

$$\hat{p}(j|X^{\ell+1}) = \begin{cases} 1 & \text{if } X^{\ell} \text{ match the class } j \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

Finally, ρ is defined as:

$$\rho = \alpha\mathcal{N}\left(X^{\ell+1}; \hat{\mu}_j^{\ell}, \hat{\Sigma}_j^{\ell}\right) \tag{35}$$

where $\mathcal{N}(X^{\ell+1}; \hat{\mu}_j^{\ell}, \hat{\Sigma}_j^{\ell})$ represents the Gaussian probability density function with mean $\hat{\mu}_j^{\ell}$ and covariance $\hat{\Sigma}_j^{\ell}$) at $X^{\ell+1}$.

In the case when none of the K distributions matches that pixel value, the least probable component is replaced by a distribution with the current value as its mean, an initially high variance, and a low weight parameter. According to their papers [12,13,29], only two parameters, α and T, have to be set in the method. However, there are four major problems when using this method. First, modeling both foreground and background pixels by a mixture of Gaussians implies the assumption that they are symmetrical which is not always the case. Second, using a prefixed number of components to represent all pixel mixtures is not practical in real life. The method is not robust when dealing with busy environments because a clean background is rare. Last but not least, slow adaptations in the means and the covariance matrices, therefore the tracker can fail within a few seconds after initialization. In this paper, we are trying to overcome these drawbacks by:

1. Using the AGM to model the non-symmetricity of the pixels distributions.
2. Using the MML to choose the right number of components for each pixel distribution.
3. Using another method to update the mixture parameters.
4. Removing the connection between the likelihood term and ρ.

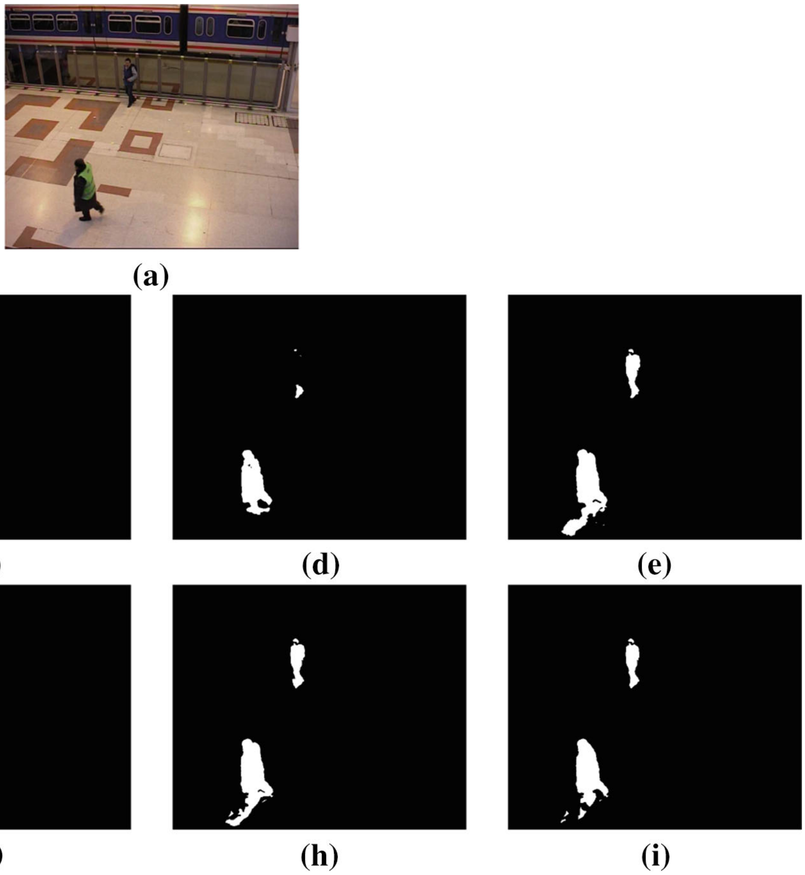


Fig. 2 **a** Sample frame from Pets2006 video sequence in the baseline category, **b** Stauffer et al. [12], **c** Zivkovic [14], **d** KaewTraKulPong et al. [37], **e** Evangelio et al. [41], **f** ELgammal et al. [42], **g** Nonaka et al. [43], **h** AGM, **i** AGM+SD

In our method, we start by representing every pixel at a given time frame ℓ by a vector of three values: $\mathbf{X}^{(\ell)} = [R, G, B]$, where R, G, B are the red, green, and blue values taken from the color camera. Then, as argued above, we model each pixel by an AGM to enhance the robustness of our algorithm in modeling the non-symmetricity of pixels distributions. To update the parameters of the AGM mixture at an input frame $(\ell + 1)$, we check whether its new value matches one of the M components of its AGM mixture. A match to a component occurs when the value of the pixel $\mathbf{X}^{(\ell+1)}$ falls within K standard deviations of the mean of the component (depending on the position of the pixel value from the mean, we use the left or right standard deviation). If a match occurs then we update the component parameters by:

$$p_j^{\ell+1} = p_j^{\ell} + B_\ell \left[p(j|\mathbf{X}^{\ell+1}) - p_j^{\ell} \right] \tag{36}$$

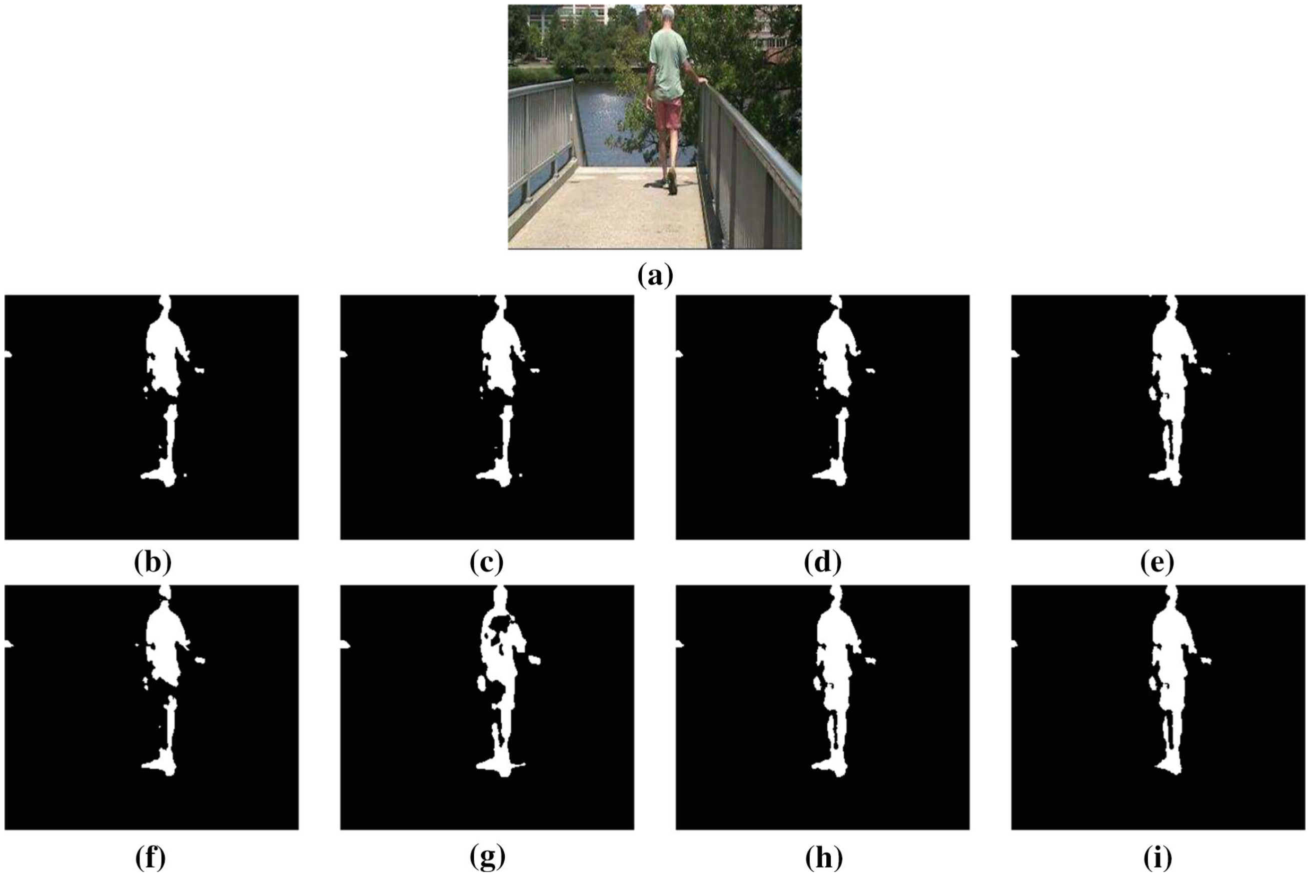
$$\xi_j^{\ell+1} = \xi_j^{\ell} + B_\ell \left[p(j|\mathbf{X}^{\ell+1}) \frac{\partial L(\Theta, Z, \mathbf{X}^{\ell+1})}{\partial \xi_j} \right] \tag{37}$$

where B_ℓ represents any sequence of positive numbers that decreases to zero. The derivatives in Eq. 37 are given in Appendix A. If no match occurs, we create a new component for the mixture with the mean equal to the new value of the pixel. Then, we evaluate the new model $\mathcal{M}^{\ell+1}$ with MML (note that $\mathcal{M}^{\ell+1}$ denotes the mixture model associated with the pixel $\mathbf{X}$ at time $\ell + 1$). In other words, we calculate the message length for the new mixture model with $M + 1$ components: if $MessLen(\ell) > MessLen(\ell + 1)$, then we use $\mathcal{M}^{\ell+1}$ else we use $\mathcal{M}^{\ell}$ and update the parameters using Eqs. (36) and (37). In the case when there is an empty component j, $p_j = 0$, we discard the component j of the mixture and set $M \leftarrow M - 1$. Finally, using the same idea of [12], we order the mixture components by the value of $\frac{p_j}{||\boldsymbol{\sigma}_{l_j}||+||\boldsymbol{\sigma}_{r_j}||}$, where $||\boldsymbol{\sigma}_{l_j}||$ and $||\boldsymbol{\sigma}_{r_j}||$ are the norm of the left and right standard deviations of the component j, respectively. At that point, we use Eq. 30 to model the background by the first B components.

The complete algorithm for adaptive background subtraction can be summarized in the following steps:

Step 1- Initialization for each pixel X:

1. Set M= 1, p_1= 1
2. $\forall k = 1, \ldots, d$: set $\sigma_{l_{1k}} = \sigma_{r_{1k}} = 0.2$, and $\mu_{1k} = X_k^{(0)}$.


(a)

(b) (c) (d) (e)

(f) (g) (h) (i)

Fig. 3 **a** Sample frame from Overpass video sequence in the dynamic background category, **b** Stauffer et al. [12], **c** Zivkovic [14], **d** KaewTraKulPong et al. [37], **e** Evangelio et al. [41], **f** ELgammal et al. [42], **g** Nonaka et al. [43], **h** AGM, **i** AGM+SD

```
Step 2- For each pixel $\mathbf{X}^{(\ell+1)}$ in a new frame do {
  1. Verify if there is a match that exists for the new pixel value.
     (a) If True
           i Update the new pixel model parameters using Eqs. 36 and 37.
     (b) If False
           i Add a new component to the mixture with mean equal the new pixel value.
          ii Evaluate the new model $\mathcal{M}^{\ell+1}$ with MML.
             A. If $MessLen(\ell) > MessLen(\ell+1)$ {$M \leftarrow M + 1$}
             B. If $MessLen(\ell) < MessLen(\ell+1)$ { use $\mathcal{M}^{\ell}$ and update it using Eqs. 36 and 37}.
  2. Check if there is any $p_j = 0$ then {
     $M \leftarrow M - 1$ }.
  3. Order the pixel mixture components by the values of $\frac{p_j}{||\sigma_{l_j}||+||\sigma_{r_j}||}$.
  4. Use Eq. 30 to extract foreground objects.
```

3.2 Shadow detection algorithm

3.2.1 Related works

Moving shadows are a major problem for foreground detection algorithms. Shadows pixels in any image differentiate themselves from the background and generally fall within the group of pixels associated with foreground objects. Labeling cast shadows as foreground objects lead to silhouette distortions and object fusions, thus reducing vision algorithm efficiency for scene monitoring and target recognition and tracking. Therefore, an effective shadow detection method is indispensable for accurate foreground segmentation.

Moving shadows in the scene are caused by the occlusion of light sources due to the moving objects. Therefore, shadow points have lower luminance values but similar chromaticity values. However, the texture characteristic around the shadow points remains unchanged since shadows do not alter the background surfaces. Shadow detection is known to be a challenging task because: (1) shadow points are mostly

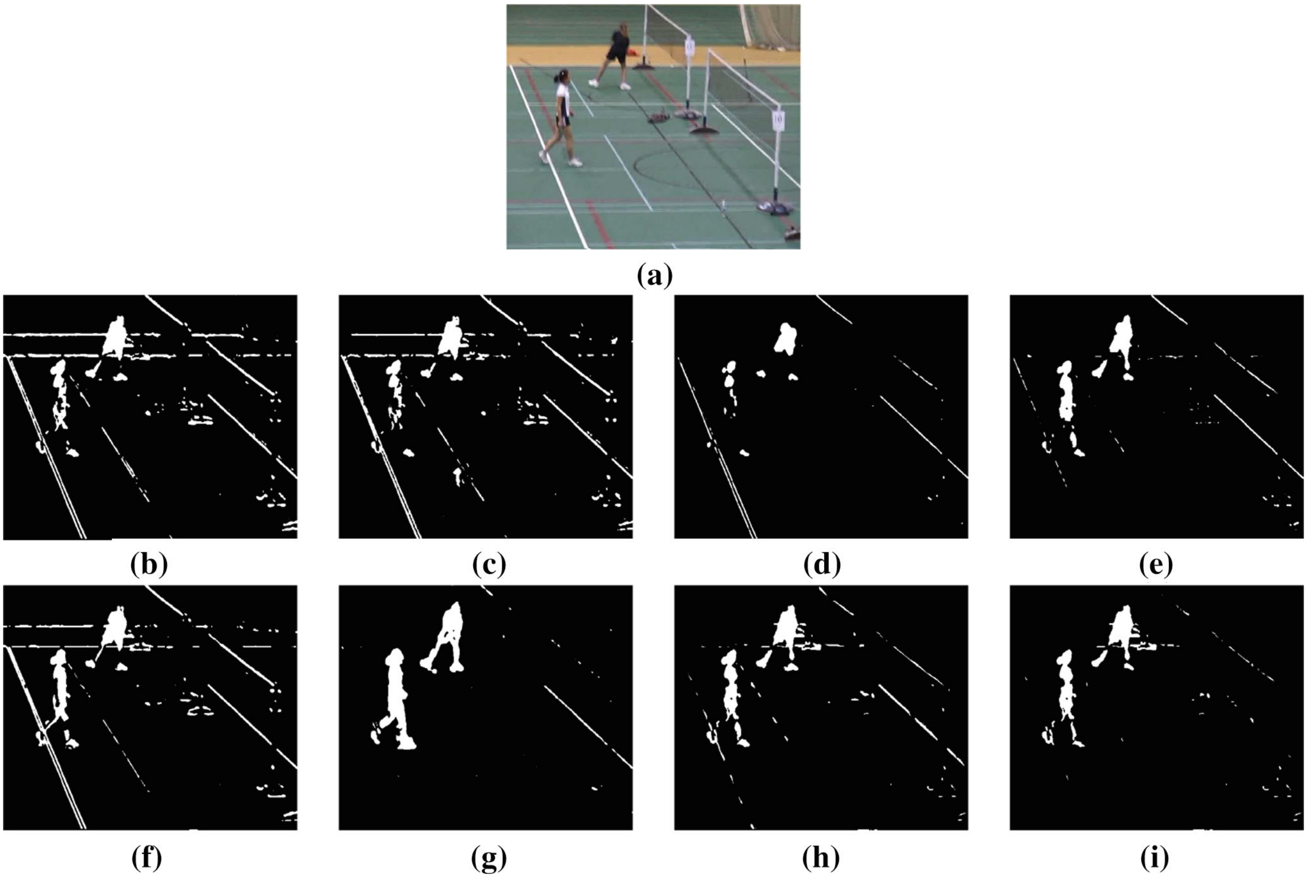


Fig. 4 **a** Sample frame from Badminton video sequence in the Camera Jitter category, **b** Stauffer et al. [12], **c** Zivkovic [14], **d** KaewTraKulPong et al. [37], **e** Evangelio et al. [41], **f** ELgammal et al. [42], **g** Nonaka et al. [43], **h** AGM, **i** AGM+SD

classified as foreground, since they differ significantly from the background; (2) shadow has the same motion as the moving object causing it, which make the task of differentiating between them very difficult; (3) shadow is always adjacent to moving object, which make it hard to remove using common segmentation techniques.

Generally, shadow detection algorithms can be classified into three categories: color-based, texture-based, and statistic-based. The color-based approaches attempt to describe the change in the color features of shadow pixels. In [30], the authors presented a robust shadow detection approach based on brightness, saturation, and hue properties in the HSV color space. Their idea is built on the hypothesis that shadows reduce surface brightness and saturation while maintaining hue properties in the HSV color space. The work in [31] addressed the shadow detection problem in YUV color space to avoid the time consuming HSV color space transformation. They distinguished the shadow regions from the foreground regions according to the observation that the YUV pixel value of shadows is lower than the linear pixels. According to the shadow model, Salvador et al. [32] identified an initial set of shadow pixels in RGB color space basing on the fact that shadow region darkens the surface. Then, combining color invariance with geometric properties of shadow they were able to detect the shadows in different scenes. In Horprasert et al. [33], built a model in the RGB color space to express normalized luminance variation and chromaticity distortions. Though color-based methods have shown their efficiency in shadow detection, they may not be reliable in the case when moving objects have similar color as moving shadows.

On the other hand, texture-based approaches are based on the fact that texture of shadow regions and the background are similar, while the texture of moving objects are different from the background. In [34], the authors explored ratio edges for shadow detection. They proved that ratio edge is illumination invariant and that the distribution of normalized ratio edge difference is a chi-square distribution. Then, a significance test was used to detect shadows. In addition to using scene brightness distortion and chromaticity distortion, Choi et al. [35] proposed three estimators which use the properties of chromaticity, brightness, and local intensity ratio. Hence, creating a chromaticity difference that obey a standard normalize distribution between the shadow region and

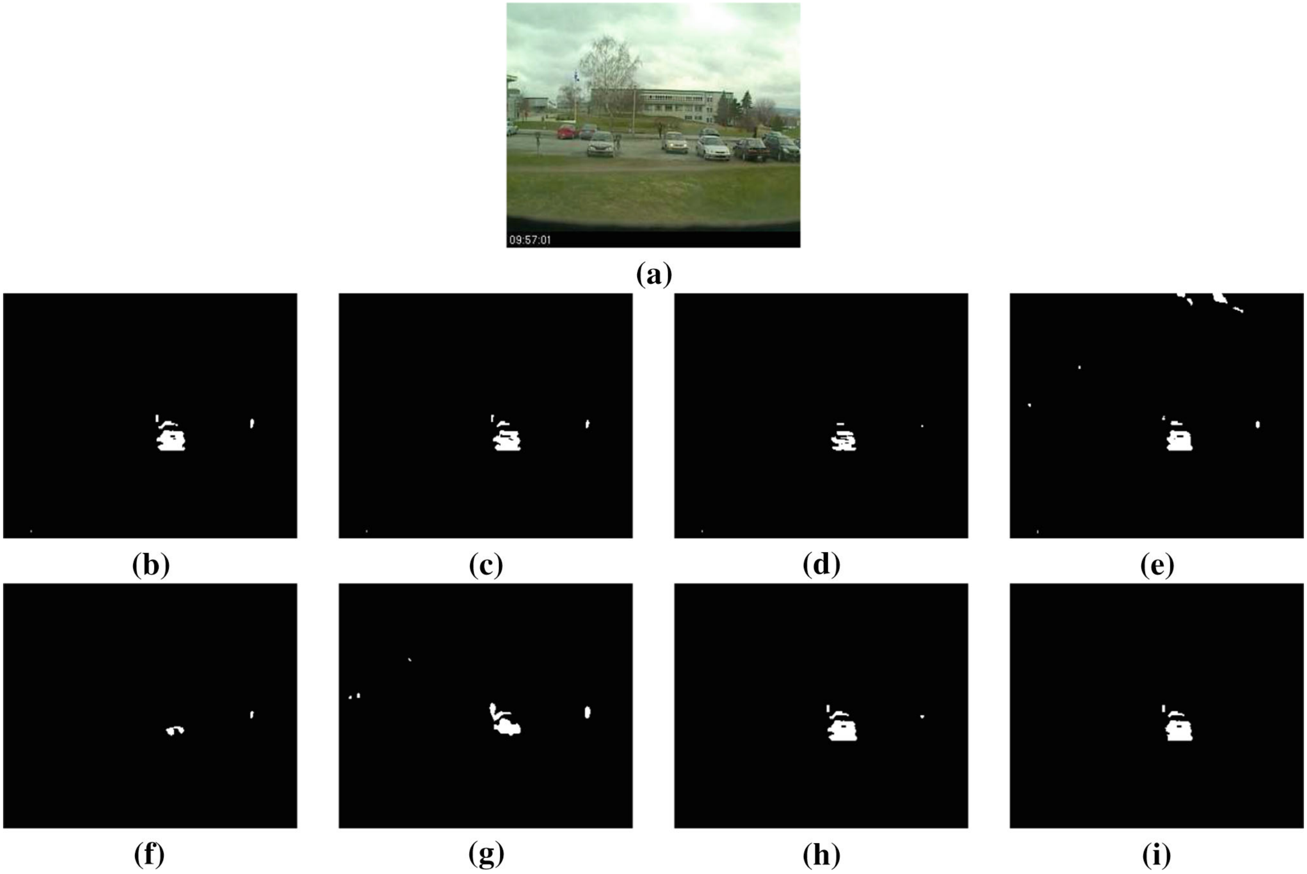


Fig. 5 **a** Sample frame from Parking video sequence in the intermittent object motion category, **b** Stauffer et al. [12], **c** Zivkovic [14], **d** KaewTraKulPong et al. [37], **e** Evangelio et al. [41], **f** ELgammal et al. [42], **g** Nonaka et al. [43], **h** AGM, **i** AGM+SD

the background. Finally, Finlayson et al. [36] used shadow edges along with illuminant invariant images to recover full color shadow-free images. Hence, texture-based methods may be the most promising technique for shadow detection because they are capable of capturing textual information of different scenes. However, they suffer from three major problems: (1) they require to set parameters for different scenes, (2) they can not handle complex and time-varying lighting conditions, and (3) they are too computationally demanding which limit their applications.

Recently, the statistical prevalence of cast shadows had been employed to learn shadows in the scenes. The principle of statistic-based methods is to build pixel-based statistical models to detect cast shadows. In [37], the authors used an adaptive Gaussian mixture model to detect moving cast shadows. The method consists of building a GMM to segment moving objects. Then, they identified the distribution of moving objects from shadows using an effective computational color model similar to the one proposed in [33]. The work in [38] proposed the use of a Gaussian mixture shadow model (GMSM). The algorithm models moving cast shadows of non-uniform and varying intensity, and builds statistical models to segment moving cast shadows using the GMM learning ability. However, the shadow models need a long time to converge while the lighting conditions should remain stable which is a major drawback. Liu et al. [39] were able to remove shadow using multilevel information in HSV color space. They attempted to improve the convergence speed of pixel-based shadow model using multilevel information. They used region-level information to increase the number of samples and global level information to update a pre-classifier. However, the method still suffers from the slow learning of the conventional GMM [12], and the low discriminativity of the pre-classifier in scenes having different types of shadows. Statistical-based methods are widely applied due to their robustness in different scenes; however, they are less effective in a real-world environment.

3.2.2 Proposed approach

In this section, we present a novel pixel-based statistical approach to model moving cast shadows. Normally, when

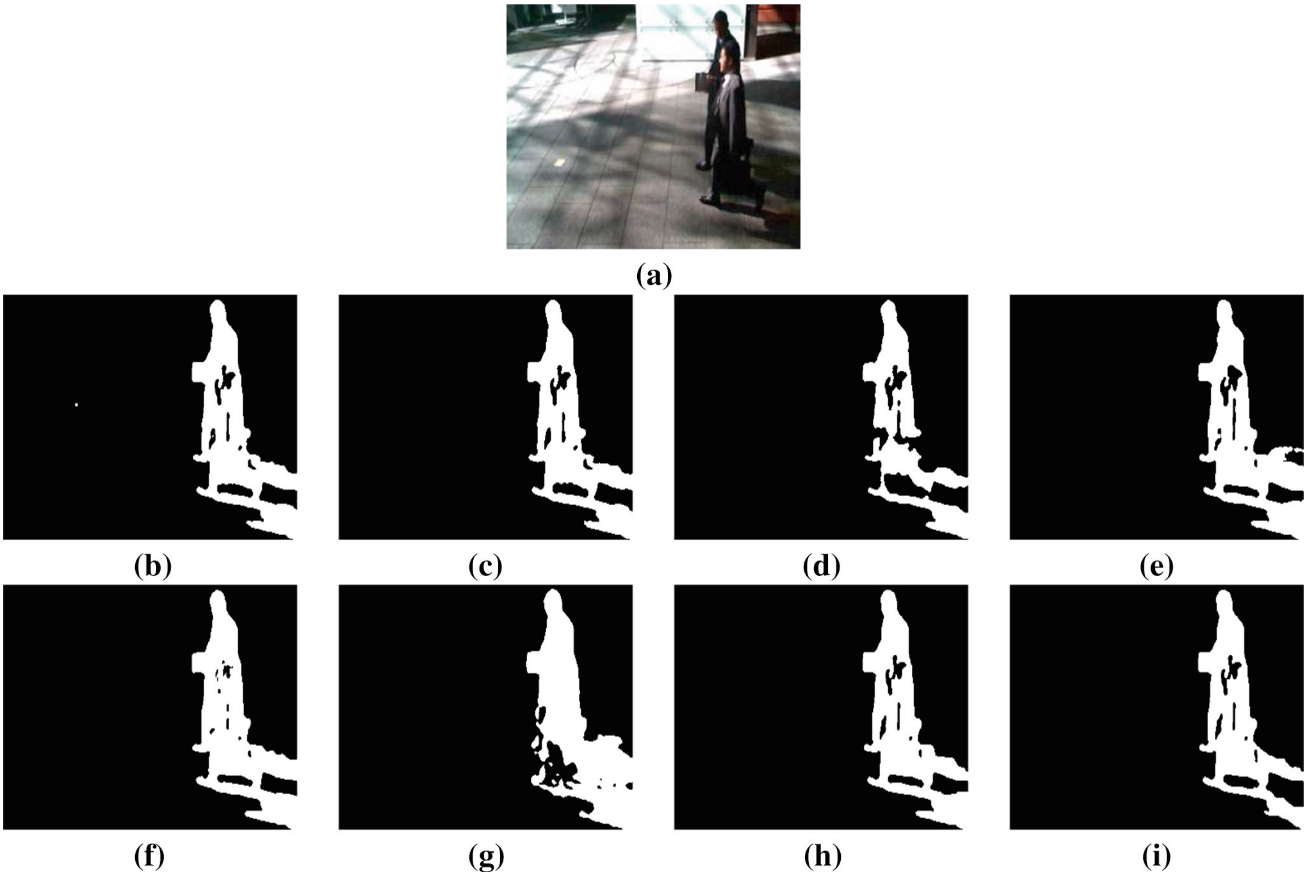


Fig. 6 **a** Sample frame from PeopleInShade video sequence in the shadows category, **b** Stauffer et al. [12], **c** Zivkovic [14], **d** KaewTraKulPong et al. [37], **e** Evangelio et al. [41], **f** ELgammal et al. [42], **g** Nonaka et al. [43], **h** AGM, **i** AGM+SD

using adaptive mixture algorithms, values that are frequently seen by a pixel are captured into stable background distributions while values that are infrequently seen are classified into foreground objects. Shadow values lie between both situations: they are not as frequent as background values, but their rate of appearance is higher than random foreground values. Therefore, in most cases, they are classified as foreground objects. Hence, the purpose of our shadow detection algorithm is to remove cast shadows classified by the adaptive AGM as foreground objects. Our idea is very simple and makes use of the property that shadows darken the surface upon which they are cast [32]. Let us consider $\mathbf{X}^{\ell+1} = (X_R^{\ell+1}, X_G^{\ell+1}, X_B^{\ell+1})$ belonging to the foreground distribution. We classify the pixel $\mathbf{X}^{\ell+1}$ as shadow if its distribution mean is smaller than that of the pixel background models for all three channels. The steps of this approach can be summarized as:

```
Input: The output of the asymmetric Gaussian mixture.
Output: Shadow candidates
For for each foreground distribution $F$ do {
    1. Compare its mean $\mu_R^F$ to the mean $\mu_R^B$ values of the
       $B$ background distributions.
    2. Compare its mean $\mu_G^F$ to the mean $\mu_G^B$ values of the
       $B$ background distributions.
    3. Compare its mean $\mu_B^F$ to the mean $\mu_B^B$ values of the
       $B$ background distributions.
    4. If $(\mu_R^F - \mu_R^B < 0 \,\&\, \mu_G^F - \mu_G^B < 0 \,\&\, \mu_B^F - \mu_B^B < 0)$
       then Consider this distribution to be a candidate for
       the shadow model.
} END
```

where $\mu_R^B, \mu_G^B, \mu_B^B$ are the B background distributions red, green, and blue means, respectively. Hence, after applying this algorithm we will end up with three different models for the background, foreground, and shadow.

3.3 Results

Our approach performance has been evaluated using the change detection dataset described in [40]. This dataset

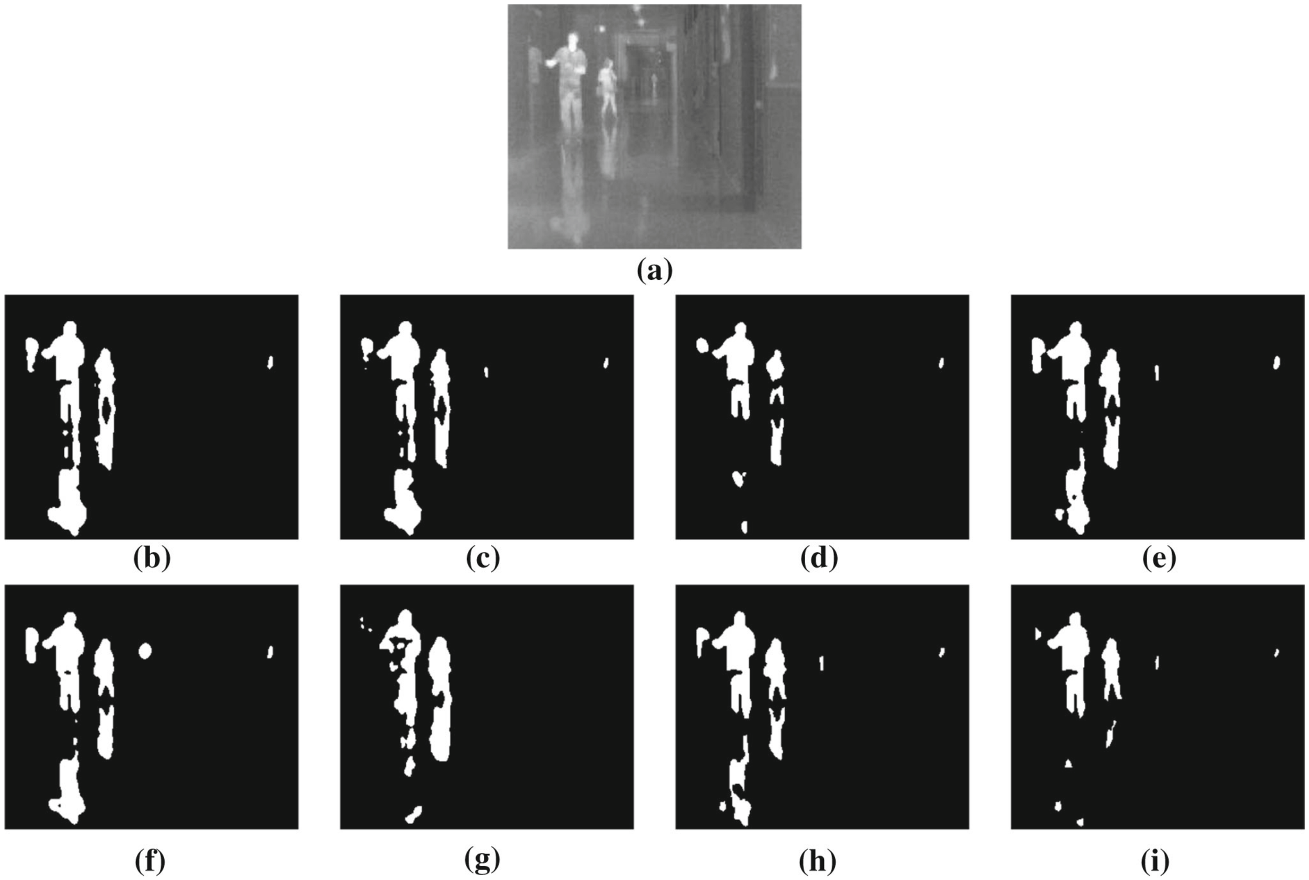


Fig. 7 **a** Sample frame from Corridor video sequence in the thermal category, **b** Stauffer et al. [12], **c** Zivkovic [14], **d** KaewTraKulPong et al. [37], **e** Evangelio et al. [41], **f** ELgammal et al. [42], **g** Nonaka et al. [43], **h** AGM, **i** AGM+SD

consists of 31 videos depicting indoor and outdoor scenes with boats, cars, trucks, and pedestrians that have been captured in different scenarios. The videos were taken with different cameras ranging from low-resolution IP cameras to thermal cameras. Therefore, spatial resolutions of the videos vary from 320 × 240 to 720 × 576 and the level of noise and compression artifacts varies from one video to another due to diverse lighting conditions present.

The videos are grouped into six categories according to the type of challenge each represents. The baseline category contains four videos, two indoor and two outdoor. There are six videos in the dynamic background category depicting outdoor scenes with strong background motion. The third category, Camera Jitter, contains one indoor and three outdoor videos captured by unstable cameras. Shadows: this category consists of two indoor and four outdoor videos exhibiting strong as well as faint shadows. Intermittent object motion is the fifth category which includes six videos with scenarios known for causing "ghosting" artifacts in the detected motion. The last category is composed of five (three outdoor and two indoor) sequences taken by far-infrared cameras. Figures (2,3,4,5,6,7) show some sample frames taken from this dataset.

In order to validate our method, we have compared it with six state-of-the-art methods. These methods can be divided into two main groups: pixel based and non-parametric Kernel Density Estimation (KDE) methods. For pixel based methods, we have used Stauffer et al. [12], Zivkovic [14], KaewTraKulPong et al. [37], and Evangelio et al. [41]; as for KDE methods, we have chosen the methods introduced by Elgammal et al. [42] and Nonaka et al. [43]. Figures (2–7) show the segmentations of our method with and without shadow detection as well as the six other methods. Note that in this application, we set the maximum number of components for the AGM to 9, the standard deviation factor $K = 2$, and the threshold $T = 0.6$. From Fig. 6, we can distinguish that the AGM+SD was not able to remove the shadow completely from the image because the difference in value between the shadow and foreground was not large enough to construct a model that represents this shadow distribution. However, from qualitative evaluation, we can notice the higher efficiency of our method.

In order to have a quantitative evaluation of the performance, we have used two well-known metrics, precision and recall, to quantify how well each algorithm works

Table 1 Baseline precision and recall

	Stauffer et al. [12] (%)	Zivkovic [14] (%)	KaewTraKulPong et al. [37] (%)	Evangelio et al. [41] (%)	ELgammal et al. [42] (%)	Nonaka et al. [43] (%)	AGM (%)	AGM+SD (%)
Highway								
Prec.	92.98	91.63	90.83	91.29	93.28	90.33	93.01	93.89
Rec.	91.82	89.16	64.96	87.81	93.79	94.59	92.27	89.07
Office								
Prec.	74.63	92.90	99.08	81.58	96.76	81.54	93.87	96.69
Rec.	49.04	50.75	36.36	67.48	90.54	30.36	71.31	65.10
Pedestrians								
Prec.	92.25	93.86	97.78	79.64	96.05	73.72	94.55	97.91
Rec.	98.68	98.20	80.15	95.93	95.40	99.72	98.33	97.18
PETS2006								
Prec.	78.56	81.35	93.60	90.86	82.84	74.31	84.51	94.82
Rec.	87.65	85.28	53.04	95.97	79.04	74.22	89.23	89.20

Table 2 Dynamic background precision and recall

	Stauffer et al. [12] (%)	Zivkovic [14] (%)	KaewTraKulPong et al. [37] (%)	Evangelio et al. [41] (%)	ELgammal et al. [42] (%)	Nonaka et al. [43] (%)	AGM (%)	AGM+SD (%)
Boats								
Prec.	70.14	80.12	89.72	96.76	60.89	44.97	92.61	93.31
Rec.	23.87	70.04	68.34	48.63	65.75	61.15	71.40	70.56
Canoe								
Prec.	89.82	91.94	99.51	98.81	93.96	87.09	94.18	96.42
Rec.	86.59	85.33	67.76	80.12	83.15	79.11	86.15	86.06
Fountain01								
Prec.	4.01	4.31	47.77	3.99	5.65	6.51	50.91	51.07
Rec.	79.73	75.06	82.91	73.76	79.30	96.35	83.76	82.13
Fountain02								
Prec.	74.51	74.59	97.99	74.03	79.55	71.80	95.30	96.14
Rec.	87.17	84.37	59.06	85.33	85.28	97.04	86.10	86.12
Overpass								
Prec.	91.91	93.66	85.97	85.56	85.12	82.29	91.90	95.30
Rec.	82.94	80.76	73.74	90.25	80.03	88.68	84.22	82.16
Fall								
Prec.	3.91	28.17	69.20	40.33	18.75	32.12	66.12	71.27
Rec.	88.38	85.60	76.64	84.79	87.21	81.75	89.14	84.66

in classifying the data [44]. Precision (Eq. 38) represents the percentage of detected true positives to the total number of items detected by the algorithm. Recall (Eq. 39) is the percentage of number of detected true positives by the algorithm to the total number of true positives in the dataset:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{38}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{39}$$

where TP is the total number of true positives correctly classified by the algorithm, FP is the total number of false positives, and FN is the number of true positives that were wrongly classified as background (false negatives). Tables (1–7) show the average recall and precision for all methods. From Table 7, we can deduce that our model is capable of detecting changes under different scenarios efficiently.

According to quantitative and analytical analysis, we can conclude that the use of AGM in background detection with shadow detection increased the performance greatly.

Table 3 Camera Jitter precision and recall

	Stauffer et al. [12] (%)	Zivkovic [14] (%)	KaewTraKulPong et al. [37] (%)	Evangelio et al. [41] (%)	ELgammal et al. [42] (%)	Nonaka et al. [43] (%)	AGM (%)	AGM+SD (%)
Badminton								
Prec.	63.70	62.51	92.11	90.43	66.68	76.00	89.41	90.77
Rec.	75.53	71.47	54.80	80.44	79.04	81.61	78.07	77.95
Boulevard								
Prec.	40.02	43.79	62.25	65.21	33.59	70.57	61.13	61.90
Rec.	83.21	79.77	62.96	75.82	77.64	58.73	79.54	77.89
Sidewalk								
Prec.	42.71	35.99	53.20	89.86	49.89	64.25	80.82	88.16
Rec.	58.12	51.06	28.57	50.49	52.49	64.65	61.25	57.92
Traffic								
Prec.	58.61	52.58	68.33	64.57	44.31	68.88	66.10	70.44
Rec.	76.47	73.68	56.64	76.76	85.89	87.63	78.46	72.72

Table 4 Intermittent object motion precision and recall

	Stauffer et al. [12] (%)	Zivkovic [14] (%)	KaewTraKulPong et al. [37] (%)	Evangelio et al. [41] (%)	ELgammal et al. [42] (%)	Nonaka et al. [43] (%)	AGM (%)	AGM+SD (%)
AbondonedBox								
Prec.	65.52	62.14	67.75	66.53	53.73	79.67	67.41	72.15
Rec.	45.74	45.64	39.51	42.23	87.45	40.54	45.18	42.60
Parking								
Prec.	75.82	73.82	65.57	78.93	61.53	92.49	77.92	81.15
Rec.	74.09	69.87	36.60	65.80	26.77	46.45	73.08	70.22
StreetLight								
Prec.	89.16	92.47	99.69	92.40	48.01	78.52	97.56	98.79
Rec.	32.25	33.94	23.57	33.19	31.46	20.76	30.33	28.61
Sofa								
Prec.	85.92	89.25	96.93	94.02	85.72	87.34	92.52	95.70
Rec.	51.62	51.41	32.60	57.75	51.91	43.58	59.90	51.63
Tramstop								
Prec.	68.54	56.36	71.89	68.20	18.91	97.01	68.23	75.44
Rec.	33.74	59.34	15.79	39.88	30.09	38.64	42.41	32.62
WinterDriveway								
Prec.	16.32	13.41	15.37	19.49	8.64	54.94	25.68	25.79
Rec.	71.10	67.84	60.50	61.93	74.40	80.75	60.15	60.08

In order to evaluate the effect of changing the parameters on the performance of our models, we have used precision-recall curves. For simplicity, we have generated precision-recall curves by systematically changing the threshold parameter T and the standard deviation factor K. Figure 8 shows the effect of changing T and K on our method with and without Shadow detection.

Based on the measurements shown in Fig. 8, we can notice that both methods perform consistently very well. In addition, we can remark that varying T and K have little effect on the AGM+SD performance, as for the AGM it is affected by T alteration. Furthermore, the high overall precision of both algorithms allows our methods to operate with a low false positive rate at their sensitive operating point.

Table 5 Shadows precision and recall

	Stauffer et al. [12] (%)	Zivkovic [14] (%)	KaewTraKulPong et al. [37] (%)	Evangelio et al. [41] (%)	ELgammal et al. [42] (%)	Nonaka et al. [43] (%)	AGM (%)	AGM+SD (%)
Backdoor								
Prec.	50.81	50.94	94.33	60.84	82.26	89.94	66.43	94.89
Rec.	85.32	82.34	75.39	88.56	86.74	99.07	89.73	76.82
Bungalows								
Prec.	71.97	71.58	63.36	72.95	70.73	81.27	72.62	74.59
Rec.	89.41	87.40	58.29	94.88	83.31	89.36	94.89	93.13
BusStation								
Prec.	88.28	88.32	93.74	88.2	84.02	82.93	88.50	92.88
Rec.	73.40	71.04	45.19	86.05	74.37	63.07	90.11	89.25
Cubicle								
Prec.	10.82	55.11	88.26	67.22	55.31	85.86	66.16	87.19
Rec.	30.12	78.61	70.88	92.64	83.48	77.28	92.48	90.24
PeopleInShade								
Prec.	84.05	84.11	84.66	83.67	84.21	78.66	84.61	87.15
Rec.	94.39	92.69	79.47	96.49	96.01	69.62	96.55	95.21
CopyMachine								
Prec.	79.42	83.84	90.29	84.07	83.06	75.99	88.54	92.19
Rec.	53.92	54.14	50.15	56.19	88.29	33.41	57.43	55.70

Table 6 Thermal precision and recall

	Stauffer et al. [12] (%)	Zivkovic [14] (%)	KaewTraKulPong et al. [37] (%)	Evangelio et al. [41] (%)	ELgammal et al. [42] (%)	Nonaka et al. [43] (%)	AGM (%)	AGM+SD (%)
Corridor								
Prec.	80.75	83.93	96.20	85.84	88.06	89.55	90.72	93.90
Rec.	82.52	83.26	65.71	84.68	83.20	56.00	89.24	87.17
Library								
Prec.	84.76	81.76	96.68	93.86	97.14	96.35	94.66	94.81
Rec.	28.00	28.68	24.03	30.23	92.20	8.07	31.33	31.05
Park								
Prec.	80.66	85.07	99.95	92.57	85.85	80.42	88.14	89.47
Rec.	63.96	59.30	16.24	39.98	60.81	89.03	64.00	62.28
DiningRoom								
Prec.	93.37	92.31	98.54	94.03	88.42	95.55	93.74	94.44
Rec.	70.21	69.43	43.16	77.45	75.74	40.11	79.57	79.07
Lakeside								
Prec.	93.04	92.23	94.10	96.86	89.21	96.36	97.48	98.53
Rec.	39.88	36.41	20.59	35.80	24.29	14.12	40.84	39.78

4 Conclusion

Adaptive mixture models are popular methods for background modeling. The proposed method has provided three main improvements to the well-known adaptive Gaussian mixture model [12]. First, we have adopted the asymmetric Gaussian distribution capable of modeling non-symmetrical data. Second, we have eliminated the problem of determining the number of clusters of the AGM by the consideration of the MML criterion. Finally, we have presented a novel pixel-based statistical approach for shadow detection and removal. Our shadow scheme identifies distributions of pixel values

Table 7 Overall precision and recall

	Stauffer et al. [12] (%)	Zivkovic [14] (%)	KaewTraKulPong et al. [37] (%)	Evangelio et al. [41] (%)	ELgammal et al. [42] (%)	Nonaka et al. [43] (%)	AGM (%)	AGM+SD (%)
Prec.	70.12	70.79	82.28	78.12	68.43	76.63	81.06	83.11
Rec.	71.08	69.64	50.72	70.73	74.42	65.07	79.18	73.07

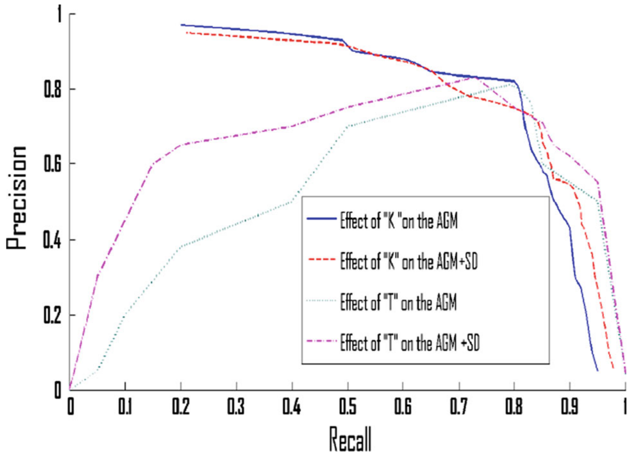


Fig. 8 Precision and recall curves for: the AGM and the AGM+SD when varying T and K

that could represent shadowed surfaces then uses them to build a second asymmetric Gaussian mixture model for shadows; hence, building shadow models capable of evolving over time. Our background subtraction approach shows good performance in terms of adaptability, accuracy and robustness, in different indoor and outdoor scenes with complex illumination variations, background movements, shadows, and ghosting artifacts.

Acknowledgments The completion of this research was made possible, thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC). The author would like to thank the anonymous referees and the associate editor for their helpful comments.

Appendix A

In this Appendix, we calculate the derivative of $\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \mu_{jk}}$, $\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \sigma_{l_{jk}}}$, $\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \sigma_{r_{jk}}}$, $\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma^2_{l_{jk}}}$, and $\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma^2_{r_{jk}}}$ used in the EM algorithm and background subtraction.

$$\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \mu_{jk}} = \sum_{i=1, X_{ik}<\mu_{jk}}^{N} \frac{\hat{Z_{ij}}(X_{ik}-\mu_{jk})}{\sigma^2_{l_{jk}}} + \sum_{i=1, X_{ik}\geq\mu_{jk}}^{N} \frac{\hat{Z_{ij}}(X_{ik}-\mu_{jk})}{\sigma^2_{r_{jk}}} \tag{40}$$

$$\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk}}} = -\sum_{i=1}^{N} \frac{\hat{Z_{ij}}}{\sigma_{l_{jk}}+\sigma_{r_{jk}}} + \sum_{i=1, X_{ik}<\mu_{jk}}^{N} \frac{\hat{Z_{ij}}(X_{ik}-\mu_{jk})^2}{\sigma^3_{l_{jk}}} \tag{41}$$

$$\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma^2_{l_{jk}}} = \sum_{i=1}^{N} \frac{\hat{Z_{ij}}}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} - 3\sum_{i=1, X_{ik}<\mu_{jk}}^{N} \frac{\hat{Z_{ij}}(X_{ik}-\mu_{jk})^2}{\sigma^4_{l_{jk}}} \tag{42}$$

$$\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk}}} = -\sum_{i=1}^{N} \frac{\hat{Z_{ij}}}{\sigma_{l_{jk}}+\sigma_{r_{jk}}} + \sum_{i=1, X_{ik}\geq\mu_{jk}}^{N} \frac{\hat{Z_{ij}}(X_{ik}-\mu_{jk})^2}{\sigma^3_{r_{jk}}} \tag{43}$$

$$\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma^2_{r_{jk}}} = \sum_{i=1}^{N} \frac{\hat{Z_{ij}}}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} - 3\sum_{i=1, X_{ik}\geq\mu_{jk}}^{N} \frac{\hat{Z_{ij}}(X_{ik}-\mu_{jk})^2}{\sigma^4_{r_{jk}}} \tag{44}$$

Appendix B

In this Appendix, we develop the solutions for Eqs. (24, 25, 26) used in the MML algorithm

$$-\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \mu^2_{jk}} = \sum_{i=l, X_{ik}<\mu_{jk}}^{l+n_j-1} \frac{1}{\sigma^2_{l_{jk}}} + \sum_{i=l, X_{ik}\geq\mu_{jk}}^{l+n_j-1} \frac{1}{\sigma^2_{r_{jk}}} \tag{45}$$

$$\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \mu_{jk_1}\mu_{jk_2}} = 0 \tag{46}$$

$$\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma^2_{l_{jk}}} = \sum_{i=l}^{l+n_j-1} \frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} - 3\sum_{i=l, X_{ik}<\mu_{jk}}^{l+n_j-1} \frac{(X_{ik}-\mu_{jk})^2}{\sigma^4_{l_{jk}}} \tag{47}$$

$$\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk_1}} \sigma_{l_{jk_2}}} = 0 \tag{48}$$

$$\begin{aligned}\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk}}^2} &= \sum_{i=l}^{l+n_j-1} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})^2} \\ &\quad - 3 \sum_{i=l, X_{ik} \geq \mu_{jk}}^{l+n_j-1} \frac{(X_{ik} - \mu_{jk})^2}{\sigma_{r_{jk}}^4} \end{aligned} \tag{49}$$

$$\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk_1}} \sigma_{r_{jk_2}}} = 0 \tag{50}$$

References

1. Tavakkoli, A., Nicolescu, M., Bebis, G., Nicolescu, M.: Non-parametric statistical background modeling for efficient foreground region detection. Mach. Vis. Appl. **7**(2), 1–15 (2009)
2. Farcas, D., Marghes, C., Bouwmans, T.: Background subtraction via incremental maximum margin criterion: a discriminative subspace approach. Mach. Vis. Appl. **23**(6), 1083–1101 (2012)
3. Cheng, J., Yang, J., Zhou, Y., Cui, Y.: Flexible background mixture models for foreground segmentation. Image Vis. Comput. **24**(5), 473–482 (2006)
4. Abbott, R.G., Williams, L.R.: Multiple target tracking with lazy background subtraction and connected components analysis. Mach. Vis. Appl. **20**(2), 93–101 (2009)
5. Fuentes, L.M., Velastin, S.A.: People tracking in surveillance applications. Image Vis. Comput. **24**(11), 1165–1171 (2006)
6. Tian, Y.L., Senior, A., Lu, M.: Robust and efficient foreground analysis in complex surveillance videos. Mach. Vis. Appl. **23**(5), 967–983 (2012)
7. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 302–309 (2004)
8. Ren, Y., Chua, C.-S., Ho, Y.-K.: Motion detection with nonstationary background. Mach. Vis. Appl. **13**(332–343), 93–101 (2003)
9. Or, S-H., Wong, Kin-H., Lee, K-S., Lao, T-K.: Panoramic video segmentation using color mixture models. In: The15th International Conference on Pattern Recognition (ICPR), Vol. 3, pp. 387–390 (2000)
10. El Baf, F., Bouwmans, T., Vachon, B.: Comparison of background subtraction methods for a multimedia learning space. In: The International Conference on Signal Processing and Multimedia Applications (SIGMAP), pp. 153–158 (2007)
11. Bouwmans, T., El Baf, F., Vachon, B.: Background modeling using mixture of gaussians for foreground detection: a survey. Recent Pat. Comput. Sci. **1**(3), 219–237 (2008)
12. Stauffer, C., Grimson, W.E.L.: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 252–258 (1999)
13. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 747–757 (2000)
14. Zivkovic, Z.: Improved adaptive Gaussian mixture model for background subtraction. In: The 17th International Conference on Pattern Recognition (ICPR), pp. 28–31 (2004)
15. Peng, A., Pieczynski, W.: Adaptive mixture estimation and unsupervised local Bayesian image segmentation. Graph. Models Image Process. **57**(5), 389–399 (1995)
16. Friedman, N., Russell, S.: Image segmentation in video sequences: a probabilistic approach. In: The Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI), pp. 1–3 (1997)
17. Lee, D.-S.: Effective Gaussian mixture learning for video background subtraction. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 827–832 (2005)
18. Wang, D., Xie, W., Pei, J., Lu, Z.: Moving area detection based on estimation of static background. J. Inf. Comput. Sci. **2**(1), 129–134 (2005)
19. Allili, M.S., Bouguila, N., Ziou, D.: Finite general gaussian mixture modeling and application to image and video foreground segmentation. J. Electron. Imaging **17**(1), 1–13 (2008)
20. Elguebaly, T., Bouguila, N.: Bayesian learning of finite generalized gaussian mixture models on images. Signal Process. **91**(4), 801–820 (2011)
21. Elguebaly, T., Bouguila, N.: A nonparametric Bayesian approach for enhanced pedestrian detection and foreground segmentation. In: IEEE computer society conference on computer vision and pattern recognition workshops (CVPRW), pp. 21–26 (2011)
22. Akaike, H.: A new look at the statistical model identification. IEEE Trans. Autom. Control **19**(6), 716–723 (1974)
23. Rissanen, J.: Modeling by shortest data description. Automatica **14**, 465–471 (1987)
24. McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
25. Bouguila, N., Ziou, D.: A Dirichlet process mixture of generalized Dirichlet distributions for proportional data modeling. IEEE Trans. Neural Netw. **21**(1), 107–122 (2010)
26. Wallace, C.S., Boulton, D.M.: An information measure for classification. Comput. J. **11**(2), 195–209 (1968)
27. Kato, T., Omachi, S., Aso, H.: Asymmetric gaussian and its application to pattern recognition. In: The Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition (SSPR), pp. 405–413 (2002)
28. Baxter, R.A., Olivier, J.J.: Finding overlapping components with MML. Stat. Comput. **10**(1), 5–16 (2000)
29. Grimson, W.E.L., Stauffer, C., Romano, R., Lee, L.: Using adaptive tracking to classify and monitor activities in a site. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 22–29 (1998)
30. Cucchiara, R., Grana, C., Piccardi, M., Prati, A., Sirotti, S.: Improving shadow suppression in moving object detection with HSV color information. In: IEEE Intelligent Transportation Systems, pp. 334–339 (2001)
31. Schreer, O., Feldmann, I., Goelz, U., Kauff, P.: Fast and robust shadow detection in videoconference applications. In: Fourth IEEE International Symposium on Video/Image Processing and Multimedia Communications, pp. 371–375 (2002)
32. Salvador, E., Cavallaro, A., Ebrahimi, T.: Cast shadow segmentation using invariant color features. Comput. Vis. Image Underst. **95**(2), 238–259 (2004)
33. Horprasert, T., Hardwood, D., Davis, L.S.: A statistical approach for real-time robust background subtraction and shadow detection. In: IEEE International Conference on Computer Vision, pp. 1–19 (1999)
34. Zhang, SW., Fang, X. Z., Yang, X.K.: Moving cast shadows detection using ratio edge. IEEE Trans. Multimed. **9**(6):1202–1214 (2007)
35. Choi, J.M., Yoo, Y.J., Choi, J.Y.: Adaptive shadow estimator for removing shadow of moving object. Comput. Vis. Image Underst. **114**(9), 1017–1029 (2010)
36. Finlayson, G.D., Hordley, S.D., Lu, C., Drew, M.S.: On the removal of shadows from images. IEEE Trans. Pattern Anal. Mach. Intell. **28**(1), 5968 (2006)
37. Kaewtrakulpong, P., Bowden, R.: An improved adaptive background mixture model for realtime tracking with shadow detection.

In: Workshop on Advanced Video Based Surveillance Systems, pp. 1–5 (2001)

38. Martel-Brisson, N., Zaccarin, A.: Learning and removing cast shadows through a multidistribution approach. IEEE Trans. Pattern Anal. Mach. Intell. **29**(7), 1133–1146 (2007)
39. Liu, Z., Huang, K., Tan, T., Wang, L.: Cast shadow removal combining local and global features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), p. 18 (2007)
40. Goyette, N., Jodoin, P.-M., Porikli, F., Konrad, J., Ishwar, P.: changedetection.net: a new change detection benchmark dataset. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1–8 (2012)
41. Evangelio, R.H., Patzold, M., Sikora, T.: Splitting gaussians in mixture models. In: IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance, pp. 300–305 (2012)
42. Elgammal, A., Harwood, D., Davis, L.: Non-parametric model for background subtraction. In: The 6th European Conference on Computer Vision (ECCV), pp. 751–767 (2000)
43. Nonaka, Y., Shimada, A., Nagahara, H., Taniguchi, R.: Evaluation report of integrated background modeling based on spatio-temporal features. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 9–14 (2012)
44. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance application. IEEE Trans. Image Process. **17**(7), 1168–1177 (2008)

Author Biographies

Tarek Elguebaly received a Bachelor's degree in Electronics and Communication Engineering from Cairo University in 2006 and a Master's degree in Information Systems from Concordia University in 2009. He is currently pursuing Ph.D. in Computer Engineering from Concordia University. He is the recipient of the NSERC–CGS (Natural Sciences and Engineering Research Council of Canada) scholarship and the FQRNT (Fonds Québecois de la Recherche sur la Nature et les Technologies) Scholarship at the doctorate level. His research interests include machine learning, computer vision, and image processing.

Nizar Bouguila received the Engineer degree from the University of Tunis, Tunis, Tunisia, in 2000, and M.Sc. and Ph.D. degrees in Computer Science from Sherbrooke University, Sherbrooke, QC, Canada, in 2002 and 2006, respectively. He is currently an Associate Professor with the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Montreal, Qc, Canada. His research interests include image processing, machine learning, data mining, 3D graphics, computer vision, and pattern recognition. Prof. Bouguila received the best Ph.D Thesis Award in Engineering and Natural Sciences from Sherbrooke University in 2007. He was awarded the prestigious Prix d'excellence de l'association des doyens des études supérieures au Québec (best Ph.D Thesis Award in Engineering and Natural Sciences in Québec), and was a runner-up for the prestigious NSERC doctoral prize. He is the author or coauthor of more than 100 publications. He is an IEEE senior member.