Finite general Gaussian mixture modeling and application to image and video foreground segmentation

Mohand Saïd Allili
University of Sherbrooke
Department of Computer Science
Faculty of Science
Sherbrooke, J1K 2R1
Quebec, Canada
E-mail: ms.allili@usherbrooke.ca

Nizar Bouguila
Concordia University
Concordia Institute for Information Systems Engineering
1515 Saint Catherine Street West
EV007.632
Montreal H3G 2W1
Quebec, Canada

Djemel Ziou
University of Sherbrooke
Department of Computer Science
Faculty of Science
Sherbrooke, J1K 2R1
Quebec, Canada

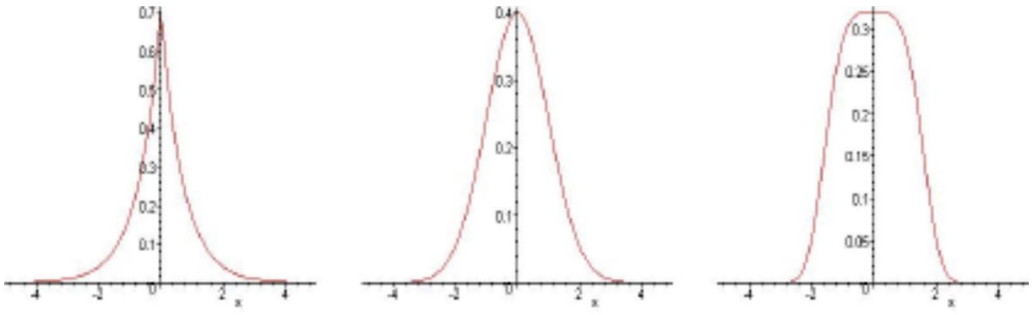
Abstract. *We propose a new finite mixture model based on the formalism of general Gaussian distribution (GGD). Because it has the flexibility to adapt to the shape of the data better than the Gaussian, the GGD is less prone to overfitting the number of mixture classes when dealing with noisy data. In the first part of this work, we propose a derivation of the maximum likelihood estimation for the parameters of the new mixture model, and elaborate an information-theoretic approach for the selection of the number of classes. In the second part, we validate the proposed model by comparing it to the Gaussian mixture in applications related to image and video foreground segmentation. © 2008 SPIE and IS&T.* [DOI: 10.1117/1.2898125]

1 Introduction

Finite Gaussian mixture models are widely used in various fields of computer vision and image processing.[1–5] This model-based approach to clustering makes it possible to validate a given model order in a formal way.[6] However, it is well known that Gaussian density has some drawbacks, such as the rigidity of its shape, which prevents it from yielding a good approximation to data with outliers.[7] For this reason, many researchers, especially in the signal processing community, have started to use general Gaussian density (GGD) for its flexibility to model data with different shapes. The GGD has been used recently in speech modeling,[8] wavelet-based texture retrieval,[9] and video coding.[10] Other applications based on GGD have also been proposed for signal deconvolution[11] and discrete cosine transform (DCT) image coding.[12] Recently, the authors in Ref. 13 used GGD to estimate reliable location parameters and regression coefficients in data containing noise or outliers. However, most of these research efforts have focused on using GGD to model unimodal data, i.e., the data are modeled using a single GGD. In applications involving data with many clusters, such as segmentation, a multicomponent probabilistic model using mixture modeling is required.

The focus of the present work is the utilization of GGD for robust mixture modeling in the context of noisy image and video foreground segmentation. A mixture model using the formalism of general Gaussian distribution is proposed, and we denote it by MoGG (by analogy to the notation commonly used in the literature for a mixture of Gaussian distributions: MoG). Because it has the flexibility to fit the shape of data better than a MoG, the new model is capable of yielding a robust mixture representation for noisy data. By robustness, we mean the ability of the model to repre-

Paper 06159R received Sep. 4, 2006; revised manuscript received Apr. 26, 2007; accepted for publication Jul. 16, 2007; published online Mar. 28, 2008.
1017-9909/2008/17(1)/013005/13/$25.00 © 2008 SPIE and IS&T.



(a) (b) (c)

Fig. 1 Different representations of a 1-D general Gaussian distribution according to the parameter λ. μ and σ are set to 0 and 1, respectively. (a) λ=1, (b) λ=2, and (c) λ=4.

sent the shape of data accurately, with less sensitivity to overfitting the number of classes in the presence of noise or outliers. In this regard, we show that the MoGG offers a better performance than the MoG, a finding we have validated on applications involving image and video foreground segmentation.

This work is organized as follows. In Sec. 2, we present the maximum likelihood estimation of the parameters of the MoGG. Section 3 describes the method for model selection using the minimum message length (MML) criterion. In Sec. 4, we test the performance of the new model on examples of segmentation. We end with conclusions and some perspectives.

2 Mixture of General Gaussian Distributions

General Gaussian distribution for a variable $X \in \mathbb{R}$ is defined as follows:

$$p(X|\mu,\sigma,\lambda) = \frac{\lambda\left[\frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)}\right]^{1/2}}{2\sigma\Gamma(1/\lambda)} \exp\left[-A(\lambda)\left|\frac{X-\mu}{\sigma}\right|^{\lambda}\right], \quad (1)$$

where

$$A(\lambda) = \left[\frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)}\right]^{\lambda/2},$$

$\Gamma(\cdot)$ denotes the gamma function, and μ and σ are the distribution mean and standard deviation parameters. The parameter $\lambda \geq 1$ controls the tails of the pdf and determines whether it is peaked or flat: the larger the value of λ, the flatter the pdf, and the smaller λ is, the more peaked the pdf (see Fig. 1). This gives the pdf a flexibility to fit the shape of heavy-tailed data.[13] Note that the Laplacian and Gaussian distributions are particular cases of the GGD where λ =1 and 2, respectively. With a mixture of M GGDs, the probability of random variable X is given by:

$$p(X|\Theta) = \sum_{j=1}^{M} p(X|\mu_j,\sigma_j,\lambda_j)p_j, \quad (2)$$

where $0 < p_j \leq 1$ and $\Sigma_{j=1}^{M} p_j = 1$. The parameters of the mixture with M classes are $\Theta=(\xi_1,\xi_2,\xi_3,\xi_4)$, where $\xi_1 =(\mu_1,\ldots,\mu_M)$, $\xi_2=(\sigma_1,\ldots,\sigma_M)$, $\xi_3=(\lambda_1,\ldots,\lambda_M)$, and $\xi_4 =(p_1,\ldots,p_M)$ is the vector constituted by the mixing parameters. Two important problems commonly arise in finite mixture modeling: the estimation of the set of parameters Θ and the determination of the number of classes M. When the number of classes is known, statistical inferential methods about the parameters can be used, primarily via maximum likelihood estimation. For the selection of the number of classes, many approaches have been suggested, such as the minimum message length (MML),[14] Akaike information criterion (AIC),[15] the minimum description length (MDL),[16] and the Laplace empirical criterion (LEC).[6] Several papers have also demonstrated the performance of Bayesian methods in the selection of the Gaussian mixture model, for example, Refs. 6 and 17.

2.1 *Maximum Likelihood Estimation of the Mixture of General Gaussian Parameters*

Let us consider a set of data $\mathcal{X}=\{X_1,X_2,\ldots,X_N\}$. For the moment, we suppose the number of mixture components M is known. The maximum likelihood method consists of getting the mixture parameters that maximize the log-likelihood function given by:

$$\max_{\Theta}\{\log[p(\mathcal{X}|\Theta)]\} = \max_{\Theta}\left\{\log\left[\prod_{X_i\in\mathcal{X}} p(X_i|\Theta)\right]\right\}, \quad (3)$$

with the constraint $\Sigma_{j=1}^{M} p_j = 1$. To take into account this constraint, we use a Lagrange multiplier and maximize the following function:

$$\Phi(\mathcal{X},\Theta,\Lambda) = \log[p(\mathcal{X}|\Theta)] + \Lambda\left(1 - \sum_{j=1}^{M} p_j\right), \quad (4)$$

where Λ is the Lagrange multiplier. The estimation of the parameters Θ is then reduced to solving the following two equations:

$$\frac{\partial\Phi(\mathcal{X},\Theta,\Lambda)}{\partial\Theta} = 0, \quad (5)$$

$$\frac{\partial\Phi(\mathcal{X},\Theta,\Lambda)}{\partial\Lambda} = 0. \quad (6)$$

Straightforward manipulations yield the iterative equations:

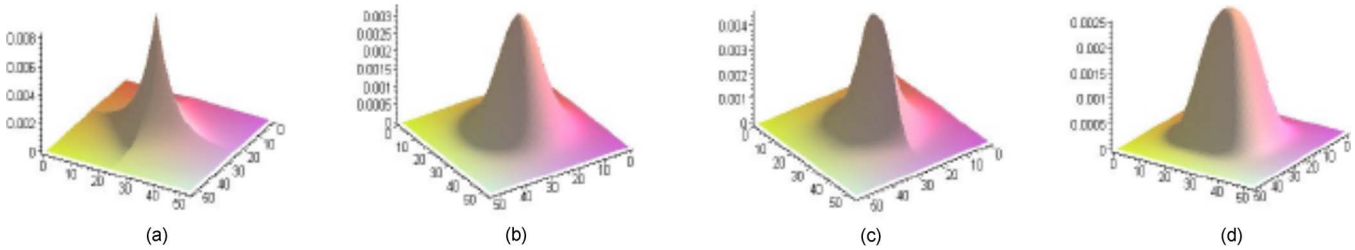


Fig. 2 Different representations of a 2-D general Gaussian distribution with respect to the parameter $\vec{\lambda}$. $\vec{\mu}$, and $\vec{\sigma}$ are set to (23,24) and (7,7), respectively. (a) $\vec{\lambda}=(1.1,1.1)$, (b) $\vec{\lambda}=(2,2)$, (c) $\vec{\lambda}=(1.1,2.8)$, and (d) $\vec{\lambda}=(2.8,2.8)$.

$$\hat{p}_j = \frac{1}{N}\sum_{i=1}^{N} p(j|X_i), \tag{7}$$

$$\hat{\mu}_j = \frac{\sum_{i=1}^{N} p(j|X_i)|X_i-\mu_j|^{\lambda_j-2}X_i}{\sum_{i=1}^{N} p(j|X_i)|X_i-\mu_j|^{\lambda_j-2}}, \tag{8}$$

$$\hat{\sigma}_j = \left[\frac{\lambda_j A(\lambda_j)\sum_{i=1}^{N} p(j|X_i)|X_i-\mu_j|^{\lambda_j}}{\sum_{i=1}^{N} p(j|X_i)}\right]^{1/\lambda_j}, \tag{9}$$

where $p(j|X_i)$ is the posterior probability of the class j, given the observation X_i. It follows that X_i is taken from the class l if $p(l|X_i)=\arg\max_j[p(j|X_i)]$, which is the Bayes rule, and we have:

$$p(j|X_i) = \frac{p(X_i|j)p_j}{\sum_{l=1}^{M} p(X_i|l)p_l}. \tag{10}$$

For the parameter λ_j, we use the Newton-Raphson method, which is based on developing the function $\partial \log[p(\mathcal{X}|\Theta)]/\partial\lambda_j$ in a power series with respect to the parameter λ_j. We obtain the following updating equation:

$$\hat{\lambda}_j \simeq \lambda_j - \left\{\frac{\partial^2 \log[p(\mathcal{X}|\Theta)]}{\partial\lambda_j^2}\right\}^{-1}\frac{\partial \log[p(\mathcal{X}|\Theta)]}{\partial\lambda_j}. \tag{11}$$

The calculation of the terms $\partial \log[p(\mathcal{X}|\Theta)]/\partial\lambda_j$ and $\partial^2 \log[p(\mathcal{X}|\Theta)]/\partial\lambda_j^2$ is given in Appendix B.

2.2 *Mixture of General Gaussian for Multidimensional Data*

A multidimensional generalization of the function in Eq. (1) is not trivial. In the past, multivariate elliptically symmetric distributions, such as Kotz-type[18] or multivariate power exponential distributions,[19] have been proposed, where the Mahalanobis distance in the exponent is raised to a power of a real number. However, such a generalization assumes the same shape parameter λ for all the dimensions of the data, which is very restrictive if different dimensions of the data have different shapes (for illustration, see Fig. 2). Research in the past has shown the performance of nonlinear regression models, for example, where the input variables have different powers, in achieving good approximation to data.[20] In our case, we seek a multidimensional GGD that contains a different shape parameter for each dimension. In practice, however, this objective is intractable if the data are correlated. To preserve the shape property, we suppose the dimensions are independent, which is a common and reasonable choice for high-dimensional data.[17] Given a d-dimensional data vector $\vec{X}=(X_1,\ldots,X_d)$, the probability of the vector $\vec{X}$ with a GGD is given by:

$$p(\vec{X}|\vec{\mu},\vec{\sigma},\vec{\lambda}) = \prod_{k=1}^{d}\frac{\lambda_k\left[\frac{\Gamma(3/\lambda_k)}{\Gamma(1/\lambda_k)}\right]^{1/2}}{2\sigma_k\Gamma(1/\lambda_k)} \times\exp\left[-A(\lambda_k)\left|\frac{X_k-\mu_k}{\sigma_k}\right|^{\lambda_k}\right], \tag{12}$$

where $\vec{\mu}=(\mu_1,\ldots,\mu_d)$ and $\vec{\sigma}=(\sigma_1,\ldots,\sigma_d)$. The parameter $\lambda_k\geq 1$ controls the tails of the pdf and determines whether it is peaked or flat in the k'th dimension. Let us consider a set of data $\mathcal{X}=\{\vec{X}_1,\vec{X}_2,\ldots,\vec{X}_N\}$. The parameters to estimate are now $\Theta=(\xi_1,\xi_2,\xi_3,\xi_4)$, where $\xi_1=(\vec{\mu}_1,\ldots,\vec{\mu}_M)$, $\xi_2=(\vec{\sigma}_1,\ldots,\vec{\sigma}_M)$, $\xi_3=(\vec{\lambda}_1,\ldots,\vec{\lambda}_M)$, and $\xi_4=(p_1,\ldots,p_M)$. Using maximum likelihood estimation, we obtain the following parameters for the mixture:

$$\hat{p}_j = \frac{1}{N}\sum_{i=1}^{N} p(j|\vec{X}_i), \tag{13}$$

$$\hat{\mu}_{jk} = \frac{\sum_{i=1}^{N} p(j|\vec{X}_i)|X_{ik}-\mu_{jk}|^{\lambda_{jk}-2}X_{ik}}{\sum_{i=1}^{N} p(j|\vec{X}_i)|X_{ik}-\mu_{jk}|^{\lambda_{jk}-2}}, \tag{14}$$

$$\hat{\sigma}_{jk} = \left[\frac{\lambda_{jk}A(\lambda_{jk})\sum_{i=1}^{N} p(j|\vec{X}_i)|X_{ik}-\mu_{jk}|^{\lambda_{jk}}}{\sum_{i=1}^{N} p(j|\vec{X}_i)}\right]^{1/\lambda_{jk}}, \tag{15}$$

with $i=1,\ldots,N$, $j=1,\ldots,M$, and $k=1,\ldots,d$. The parameters λ_{jk} are estimated in the same fashion as in the 1-D case, using the Newton-Raphson method.

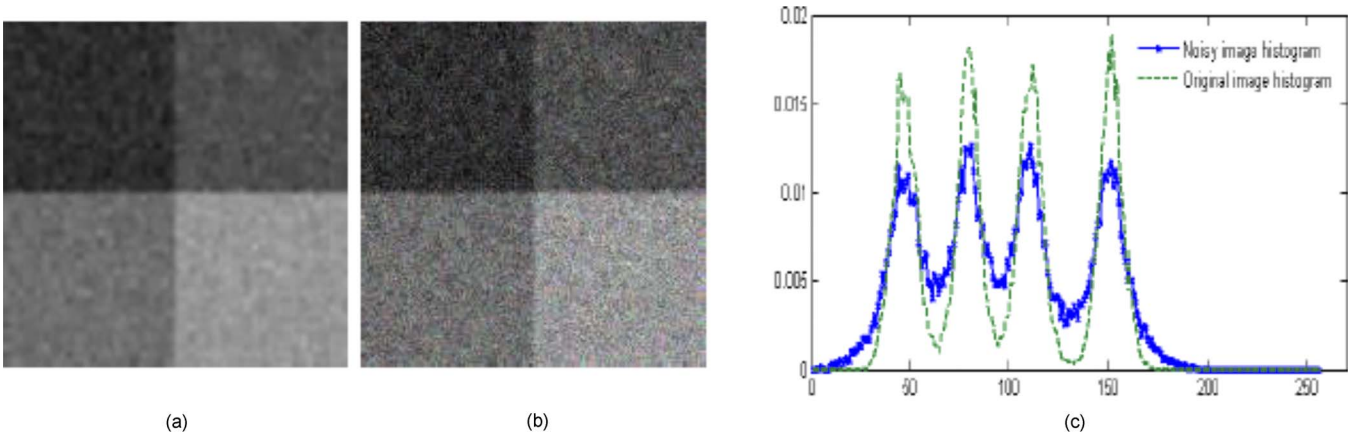


Fig. 3 Example with a synthetic image showing the problem of mixture overfitting: (a) and (b) show the original image and the noisy version of it, respectively; and (c) shows the histograms of the original and noisy images.

3 Mixture of General Gaussian Model Selection Using the Minimum Message Length Criterion

For the mixture model selection, we use the MML criterion, which is a Bayesian criterion that has shown good performance for the Gaussian mixture model.[14,17] Using the MML, the optimal number of classes of the mixture is obtained by minimizing the following function:[14,21]

$$mess\ length(M) \simeq -\log[p(\Theta_M)] - \log[p(\mathcal{X}|\Theta_M)] + \frac{1}{2}\log|\mathbf{F}(\Theta_M)| - \frac{1}{2}\log(12) + \frac{N_p}{2}, \quad (16)$$

where N_p is the number of parameters in the mixture model, and Θ_M is the set of parameters when the mixture contains M components. In what follows, we drop this notation by assuming the calculations are performed for a given M. In Eq. (16), $p(\Theta)$ is the prior probability, $p(\mathcal{X}|\Theta)$ is the likelihood given in Eq. (3), and $|\mathbf{F}(\Theta)|$ is the determinant of the Fisher information matrix minus the log-likelihood of the mixture. The estimation of the number of classes is carried out by finding the minimum with regard to Θ of the message length (*mess length*). In the following, we give the derivation of the terms $p(\Theta)$ and $|\mathbf{F}(\Theta)|$.

3.1 *Derivation of the Prior Distribution* $p(\Theta)$

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 the class i does not provide any knowledge about the parameters of a class j. Furthermore, we can assume the parameters ξ_1, ξ_2, ξ_3, and ξ_4 are mutually independent, which yields the following prior distribution over the parameters:

$$p(\Theta) = p(\xi_1)p(\xi_2)p(\xi_3)p(\xi_4). \quad (17)$$

We now define the four densities $p(\xi_1)$, $p(\xi_2)$, $p(\xi_3)$, and $p(\xi_4)$. We know that the vector ξ_4 is defined on the simplex defined by $\{(p_1,\ldots,p_M):\Sigma_{j=1}^{M-1}p_j<1\}$. Thus, a natural prior for ξ_4 is the Dirichlet distribution:

$$p(\xi_4) = \frac{\Gamma(\sum_{j=1}^{M}\eta_j)}{\prod_{j=1}^{M}\Gamma(\eta_j)}\prod_{j=1}^{M}p_j^{\eta_j-1}, \quad (18)$$

where $\vec{\eta}=(\eta_1,\ldots,\eta_M)$ is the parameter vector of the Dirichlet distribution. The choice of $\eta_1=1,\ldots,\eta_M=1$ gives a uniform prior over the space where $p_1+\ldots+p_M=1$. This prior is formulated by:

$$p(\xi_4) = (M-1)\,!\,. \quad (19)$$

Note that this uniform prior is defined over the $(M-1)$-dimensional region of hypervolume $1/(M-1)!$. It represents the inverse of the hypervolume such that the prior integrates to 1. For the parameter ξ_2, we have:

$$p(\xi_2) = \prod_{j=1}^{M}p(\vec{\sigma}_j), \quad (20)$$

where we have assumed that the different components of the vector $\vec{\sigma}_j$ are independent. Further, in the absence of other knowledge about σ_{ik}, $k=1,\ldots,d$, we use the principle of ignorance by taking a uniform prior. Suppose that $\vec{\sigma}=(\sigma_1,\ldots,\sigma_d)$ and $\vec{\mu}=(\mu_1,\ldots,\mu_d)$ are the standard deviation and mean vectors of the entire population (the whole of dataset $\mathcal{X}$). Then for each σ_{jk}, we choose the following uniform prior:

$$p(\sigma_{jk}) = \frac{1}{\sigma_k}, \quad (21)$$

where $0\leq\sigma_{jk}\leq\sigma_k$, $k=1,\ldots,d$. It follows that

$$p(\vec{\sigma}_j) = \prod_{k=1}^{d}\frac{1}{\sigma_k}. \quad (22)$$

By substituting Eq. (22) into Eq. (20), we obtain:

$$p(\xi_2) = \prod_{j=1}^{M}\prod_{k=1}^{d}\frac{1}{\sigma_k} = \prod_{k=1}^{d}\frac{1}{\sigma_k^M}. \quad (23)$$

Likewise, we take a uniform prior for each μ_{jk}. Each μ_{jk} is chosen to be uniform in the region within one standard deviation of the population mean; i.e., $\mu_k - \sigma_k \leq \mu_{jk} \leq \mu_k + \sigma_k$. Thus, the prior is given by the following equation:

$$p(\xi_1) = \prod_{j=1}^{M}\prod_{k=1}^{d} p(\mu_{jk}) = \prod_{k=1}^{d}\frac{1}{(2\sigma_k)^M}. \quad (24)$$

For the last parameter ξ_3, we adopt for each λ_{jk} a uniform distribution $\mathcal{U}[0,h]$, where the value of h is chosen sufficiently large. We obtain the following prior:

$$p(\xi_3) = \prod_{j=1}^{M}\prod_{k=1}^{d} p(\lambda_{jk}) = \frac{1}{h^{M\cdot d}}. \quad (25)$$

Finally, by substituting Eqs. (19) and (23)–(25) into Eq. (17), we obtain:

$$p(\Theta) = \frac{(M-1)!}{(2h)^{M\cdot d}}\prod_{k=1}^{d}\frac{1}{\sigma_k^{2M}}. \quad (26)$$

3.2 *Derivation of the Determinant of the Fisher Information Matrix* $|\mathbf{F}(\Theta)|$

The Hessian matrix of a mixture leads to a complicated analytical form of MML, which cannot be easily reproduced. Therefore, we approximate it by the complete Fisher information matrix,[17] which yields the following expression for the determinant $|\mathbf{F}(\Theta)|$:

$$|\mathbf{F}(\Theta)| \simeq |\mathbf{F}(\xi_4)|\prod_{j=1}^{M}|\mathbf{F}(\vec{\mu}_j)||\mathbf{F}(\vec{\sigma}_j)||\mathbf{F}(\vec{\lambda}_j)|, \quad (27)$$

where $|\mathbf{F}(\xi_4)|$ is the determinant of the Fisher information matrix with regard to the mixing parameters, and $|\mathbf{F}(\vec{\mu}_j)|$, $|\mathbf{F}(\vec{\sigma}_j)|$, and $|\mathbf{F}(\vec{\lambda}_j)|$ are the determinants of the Fisher information matrices with respect to the vectors $\vec{\mu}_j$, $\vec{\sigma}_j$, and $\vec{\lambda}_j$ of component j of the mixture. In what follows, we compute each of these terms separately.

For $|\mathbf{F}(\xi_4)|$, it should be noted that the mixing parameters satisfy the constraint $\Sigma_{j=1}^{M} p_j = 1$. 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, …, M'th cluster. The number of trials of the j'th cluster is, therefore, a multinomial distribution of parameters $p_1, p_2, \ldots, p_M$, which gives the following determinant of the Fisher information matrix:

$$|\mathbf{F}(\xi_4)| = \frac{N^{M-1}}{\prod_{j=1}^{M} p_j}, \quad (28)$$

where N is the number of data vectors. For $|\mathbf{F}(\xi_1)|$, $|\mathbf{F}(\xi_2)|$, and $|\mathbf{F}(\xi_3)|$, let us consider the j'th class $\mathcal{X}_j = (\vec{X}_l, \ldots, \vec{X}_{l+n_j-1})$ of the mixture; $\mathcal{X}_j$ here denotes the data in class j after classifying all the data $\mathcal{X}$ using the maximum *a posteriori* probability defined by Eq. (10). The choice of the j'th class allows us to simplify the notation without loss of generality. The Hessian matrices when we consider the vectors $\vec{\mu}_j$, $\vec{\sigma}_j$, and $\vec{\lambda}_j$ are given by

$$\mathbf{F}(\vec{\mu}_j)_{k_1k_2} = \frac{\partial^2}{\partial\mu_{jk_1}\partial\mu_{jk_2}}\{\log[p(\mathcal{X}_j|\Theta)]\}, \quad (29)$$

$$\mathbf{F}(\vec{\sigma}_j)_{k_1k_2} = \frac{\partial^2}{\partial\sigma_{jk_1}\partial\sigma_{jk_2}}\{\log[p(\mathcal{X}_j|\Theta)]\}, \quad (30)$$

$$\mathbf{F}(\vec{\lambda}_j)_{k_1k_2} = \frac{\partial^2}{\partial\lambda_{jk_1}\partial\lambda_{jk_2}}\{\log[p(\mathcal{X}_j|\Theta)]\}, \quad (31)$$

where $k_1, k_2 \in \{1, \ldots, d\}$. After computing the derivatives in Eqs. (29)–(31) (see Appendix A), we obtain:

$$|\mathbf{F}(\vec{\mu}_j)| = \prod_{j=1}^{M}\prod_{k=1}^{d}\left|\frac{\lambda_{jk}(1-\lambda_{jk})}{\sigma_{jk}^{\lambda_{jk}}}A(\lambda_{jk})\right|\sum_{i=1}^{l+n_j-1}|X_{ik}-\mu_{jk}|^{\lambda_{jk}-2}, \quad (32)$$

$$|\mathbf{F}(\vec{\sigma}_j)| = \prod_{j=1}^{M}\prod_{k=1}^{d}\left|\frac{n_j}{\sigma_{jk}^2} + A(\lambda_{jk})\frac{\lambda_{jk}(\lambda_{jk}+1)}{\sigma_{jk}^{\lambda_{jk}+2}}\right| \times \left|\sum_{i=1}^{l+n_j-1}|X_{ik}-\mu_{jk}|^{\lambda_{jk}}\right|, \quad (33)$$

$$|\mathbf{F}(\vec{\lambda}_j)| = \prod_{j=1}^{M}\prod_{k=1}^{d} -\frac{\partial^2\log[p(\mathcal{X}_j|\Theta)]}{\partial^2\lambda_{jk}}. \quad (34)$$

The model selection and parameter estimation of the MoGG are summarized in the algorithm given next. Given a number of components, the mixture parameters are estimated iteratively using the expectation-maximization (EM) algorithm.[6] Note that convergence of the EM is detected when the distance between the parameters resulting from two successive iterations ℓ and $\ell+1$ is smaller than a predefined threshold ϵ; i.e., $\|\Theta^{(\ell+1)} - \Theta^{(\ell)}\| < \epsilon$. Note also that the initialization of the mixing parameters and the mean and standard deviation vectors is performed using the K-means algorithm.[22]

Algorithm 1. For each candidate value of M do:

- initialization
- repeat until convergence:

E-step: compute the posterior probabilities $p(j|\vec{X}_i)$
M-step: $\forall j = 1, \ldots, M$, $\forall k = 1, \ldots, d$

update $(\lambda_{jk}, p_j, \mu_{jk}, \sigma_{jk})$ using Eqs. (11) and (13)–(15)

- calculate the associated MML criterion using Eq. (16).

Select the optimal model M^* such that $M^* = \arg\min_M$ *mess length* (M).

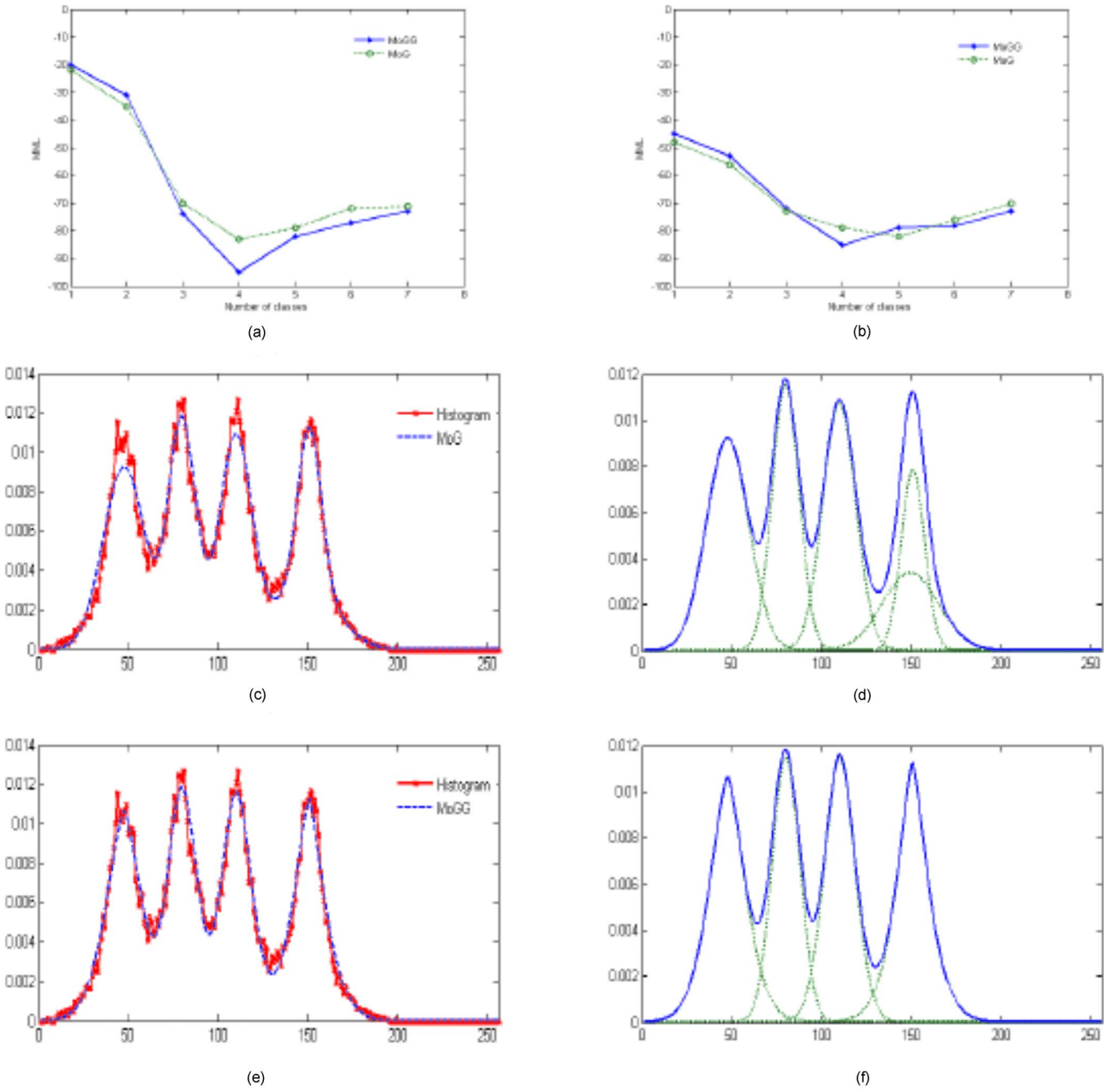


Fig. 4 Example showing the sensitivity of the MML to noise in the case of the MoG and MoGG models: (a) and (b) show the values of the MML for the two models using the original and noisy image data, respectively; (c) and (e) show the approximation of the noisy image histogram using the MoG and MoGG models, respectively; (d) and (f) show the pdfs (dashed lines) composing the MoG and MoGG models (solid lines), respectively. The estimated MoGG parameters are $\xi_1=(46.1,79.1,108.9,149.6)$, $\xi_2=(11.7,7.64,8.9,11.03)$, $\xi_3=(1.96,1.98,1.97,1.92)$, and $\xi_4=(0.27,0.22,0.25,0.26)$, while those of the MoG are $\xi_1=(46.5,79.12,108.8,145.31,149.8)$, $\xi_2=(11.7,7.68,8.9,20.1,8.4)$, and $\xi_4=(0.27,0.22,0.24,0.08,0.19)$.

4 Experiments and Discussion

4.1 *Application to Image Segmentation*

Image segmentation is one of the most important problems in computer vision and image processing. In the past, mixture models have been used for segmentation, where the aim is to build a partition of the image in which each region's data distribution is modeled using a component of the mixture.[1,4,23] Gaussian distribution has been used in these methods because of its simplicity and practicality. The fact remains, however, that with noisy data, the image histogram may be heavy-tailed and the mixture model lose its accuracy by overfitting the actual number of regions in

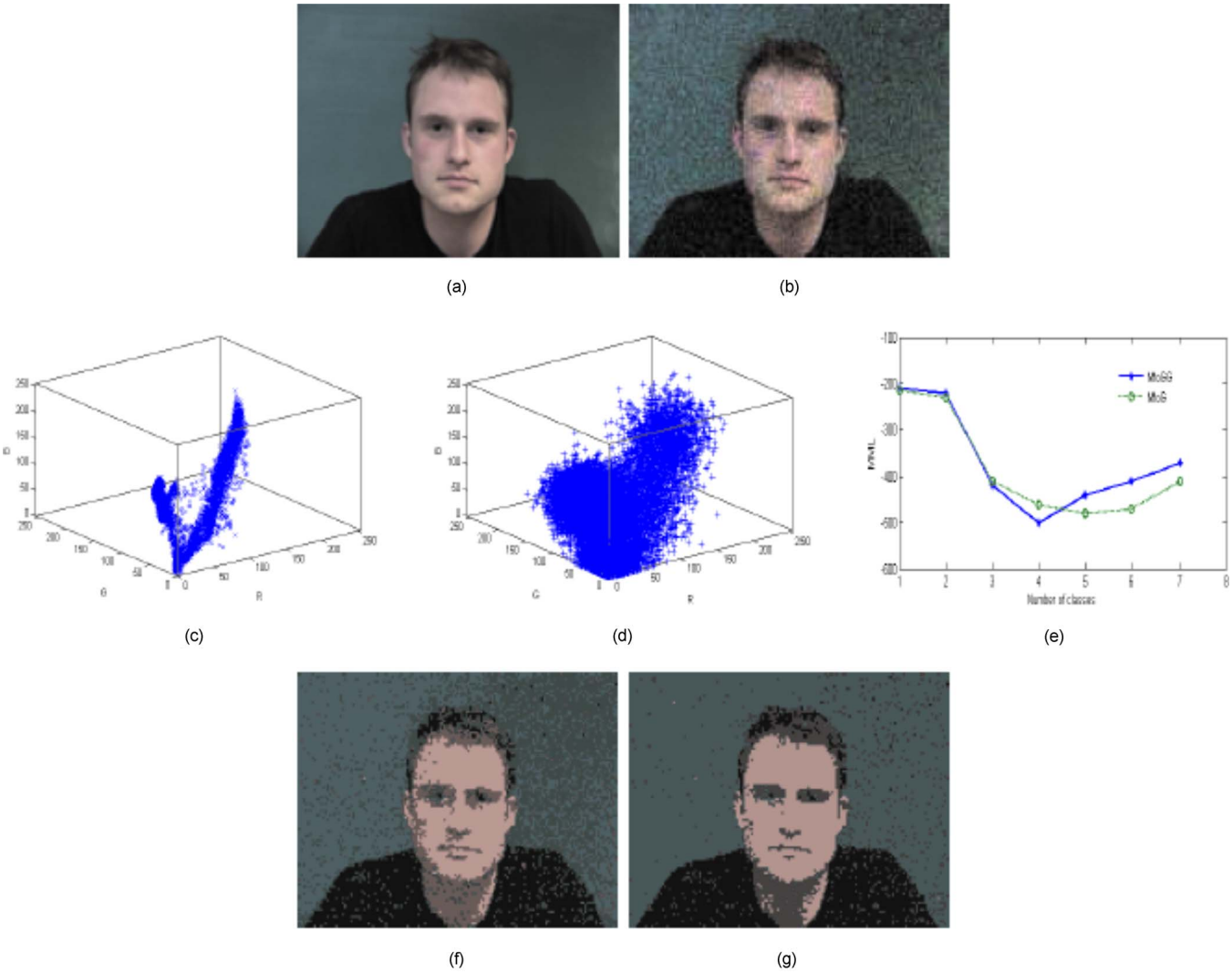


Fig. 5 Comparison of the performance of the MoG and MoGG models using an example of real-world image segmentation: (a) and (b) show the original and noisy images; (c), (d), and (e) show respectively the RGB color values contained in the original and noisy images, and the MML value obtained for the MoG and MoGG models using the noisy image data; and (f) and (g) show the segmentation of the noisy image using the MoG and MoGG models, respectively. (Color online only).

the image. To reduce the problem of overfitting, we propose applying the new mixture model for segmentation.

To illustrate the problem of mixture overfitting caused by noise, Figs. 3 and 4 show an example of mixture modeling for the intensity distribution of a synthetic image. Figures 3(a) and 3(b) show the original and noisy images, respectively. Here, the noisy image is obtained by adding to the original image a Gaussian noise with distribution $\mathcal{N}(0,\sigma_n)$; we set $\sigma_n=3$. Figure 3(c) shows the histograms of these images, where the original image histogram contains four different modes that correspond to the four regions of the image. Notice the effect of noise on the histogram, adding tails to the histogram modes and increasing their overlap. Figure 4 shows the values of the MML obtained using a MoG and MoGG to model the intensity distribution of the noisy image. Adding noise resulted in the MoG creating a fifth class to better fit the histogram. The MoG model and the pdfs associated with its components are shown in Figs. 4(c) and 4(d). In contrast, the MoGG preserved the four classes. Figures 4(e) and 4(f) show the MoGG model and the pdfs associated with its different components. Notice, for example, the right side of the histograms, where an overfitting occurred for the MoG [see Figs. 4(c) and 4(d)]. The fourth GGD of the MoGG is heavy-tailed, allowing it to fit the data in this part of the histogram without adding a new mixture component [see Figs. 4(e) and 4(f)].

Finally, Fig. 5 shows a real-world image segmentation example (the test image, of size 180×240, was downloaded from a public face database on the internet). From the color image in Fig. 5(a), we created the noisy image in Fig. 5(b) by adding Gaussian noise with distribution $\mathcal{N}(0,\sigma_n)$ to each color band. Similar to the previous experiment, we set $\sigma_n=3$. Figures 5(c) and 5(d) present 3-D graphs of the RGB color values contained in the original and noisy images, respectively. Notice in Fig. 5(d) how adding noise creates overlap between the different classes of color contained in the original image. The MML of the

MoG gave five components, while it gave only four components for the MoGG [see Fig. 5(e)]. After segmenting the noisy image using the maximum *a posteriori* (MAP) criterion {i.e., assign a pixel to the class k such that $k = \arg\max_j[p(j|\vec{X})]$}, we obtain the results shown in Figs. 5(f) and 5(g) for the MoG and MoGG models, respectively. Note that for the segmentation, we used the optimal number of regions, calculated using the MML, for both the MoG and MoGG models. To illustrate the performance of the proposed model, notice the oversegmentation of the skin region in the face of the person, obtained using the MoG. The same region has clearly been less oversegmented using the MoGG model [see Fig. 5(g)].

4.2 *Application to Video Foreground Segmentation*

Recently, adaptive Gaussian mixture models have been used for segmenting video foregrounds in sequences acquired using a static camera. The problem consists of segmenting 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. The approach showed promising results in Refs. 2 and 24 However, major challenges remain, such as handling sudden illumination changes, slow moving objects, shadows, and other phenomena that produce nonstationary backgrounds (outliers), which may cause erroneous classification of pixels to the foreground. One part of the problem comes from the sensitivity of the Gaussian mixture model in handling false foreground pixels caused by the previously mentioned phenomena. In this section, we exploit the properties of the MoGG to enhance the robustness of mixture modeling against such phenomena.

For segmenting the foreground in a video sequence, the authors in Ref. 2 propose online learning of a Gaussian mixture model for each pixel in the video frames. The components that occur frequently in the mixture, i.e., with high prior probability and small variance, are used to model the background. To segment the video foreground, the mixture components are first ordered by the value of this term: $p_j/\|\vec{\sigma}_j\|$, $j=1,\ldots,M$. Then, the first $\mathbf{B}$ components are chosen to model the background, such that

$$\mathbf{B} = \arg\min_b\left(\sum_{j=1}^{b} p_j > \mathbf{T}\right), \tag{35}$$

where $\mathbf{T}$ is a threshold and $\|\cdot\|$ designates the norm of a vector.

In what follows, we propose to build an online estimation of the parameters of the MoGG model. We suppose the frames of the video are acquired online and numbered $I^{(\ell)}$, $0 \le \ell < \infty$, according to the order of their arrival. Following the approach presented in Refs. 25–27, an iterative scheme is built for the online estimation of the MoGG parameters for each pixel (x,y). We denote by $\mathcal{M}^{(\ell)}(x,y)$ the mixture model associated with the pixel (x,y) at time ℓ. Given the parameters of the mixture model $\mathcal{M}^{(\ell)}(x,y)$, and a new value for the pixel $\vec{X}^{(\ell+1)}$ [from a new input frame $I^{(\ell+1)}$], the parameters of the mixture are updated as follows:

$$p_j^{(\ell+1)} = p_j^{(\ell)} + \beta_\ell \cdot p[j|\vec{X}^{(\ell+1)}] - p_j^{(\ell)}, \tag{36}$$

$$\theta_j^{(\ell+1)} = \theta_j^{(\ell)} + \beta_\ell \cdot p[j|\vec{X}^{(\ell+1)}]\frac{\partial \log\{p[\vec{X}^{(\ell+1)}|\theta_j]\}}{\partial \theta_j}, \tag{37}$$

where β_ℓ represents any sequence of positive numbers that decreases to zero. The derivatives in Eq. (37), with respect to the different parameters of the mixture distributions, $\vec{\mu}_j$, $\vec{\sigma}_j$, and $\vec{\lambda}_j$, are given in Appendices A and B. For the selection of the number of classes in the mixture model, we use the MML. The algorithm for foreground segmentation works as follows. Given an input frame of the sequence, for each pixel, check whether its new value matches one of the components of its MoGG mixture. A match to a component occurs when the value of the pixel $\vec{X}$ falls within two standard deviations of the mean of the component. If no match occurs, we create a new component for the mixture with the mean equal to the new value of the pixel. We calculate the MML for the new mixture model: if *mess length* $(M)>$ *mess length* $(M+1)$, then $M \leftarrow M+1$; otherwise, we update the old mixture parameters using Eqs. (36) and (37). If $\exists p_j < 0$, we discard the component j of the mixture and set $M \leftarrow M-1$. The algorithm for foreground segmentation is summed up in the following script:

Algorithm 2. Mixture initialization for each pixel $\mathbf{x} = (x,y)$:

- set $M=1$, $p_1=1$
- $\forall k=1,\ldots,d$: set $\sigma_{1k}=0.2$, $\vec{\mu}_{1k}=X_k^{(0)}$ and $\lambda_{1k}=2$.

For a new frame $I^{(\ell+1)}$, $\ell \ge 0$:

- for each new value $\vec{X}^{(\ell+1)}$ of a pixel (x,y):

  verify whether there is a match for the new pixel value $\vec{X}^{(\ell+1)}$

  update the pixel mixture parameters using Eqs. (36) and (37)

- extract the foreground object according to Eq. (35).

To compare the performance of the MoGG and MoG models, we tested both of them using two examples of foreground segmentation. In the first example, the video sequence contains shadows casted by the moving objects, whereas in the second example it contains a sudden illumination change. The first video is taken in an indoor scene (a subway station) and contains 966 frames. The first row in Fig. 6 shows the first frame of the first video. In the second row, we show the 2-D histogram of the RG colors for three typical pixels, characteristic of the following events: 1. no event (background), 2. light shadow, 3. deep shadow, and 4. moving object (foreground). In the third and fourth rows, we show the result of foreground segmentation for frames 120 and 210 of the sequence, with $\mathbf{T}=0.3$. We can see that the MoGG model shows more resilience than the MoG model against segmenting the shadows of the moving objects to the foreground. The second video is also taken in an indoor scene (an office) and contains 500 frames. It begins

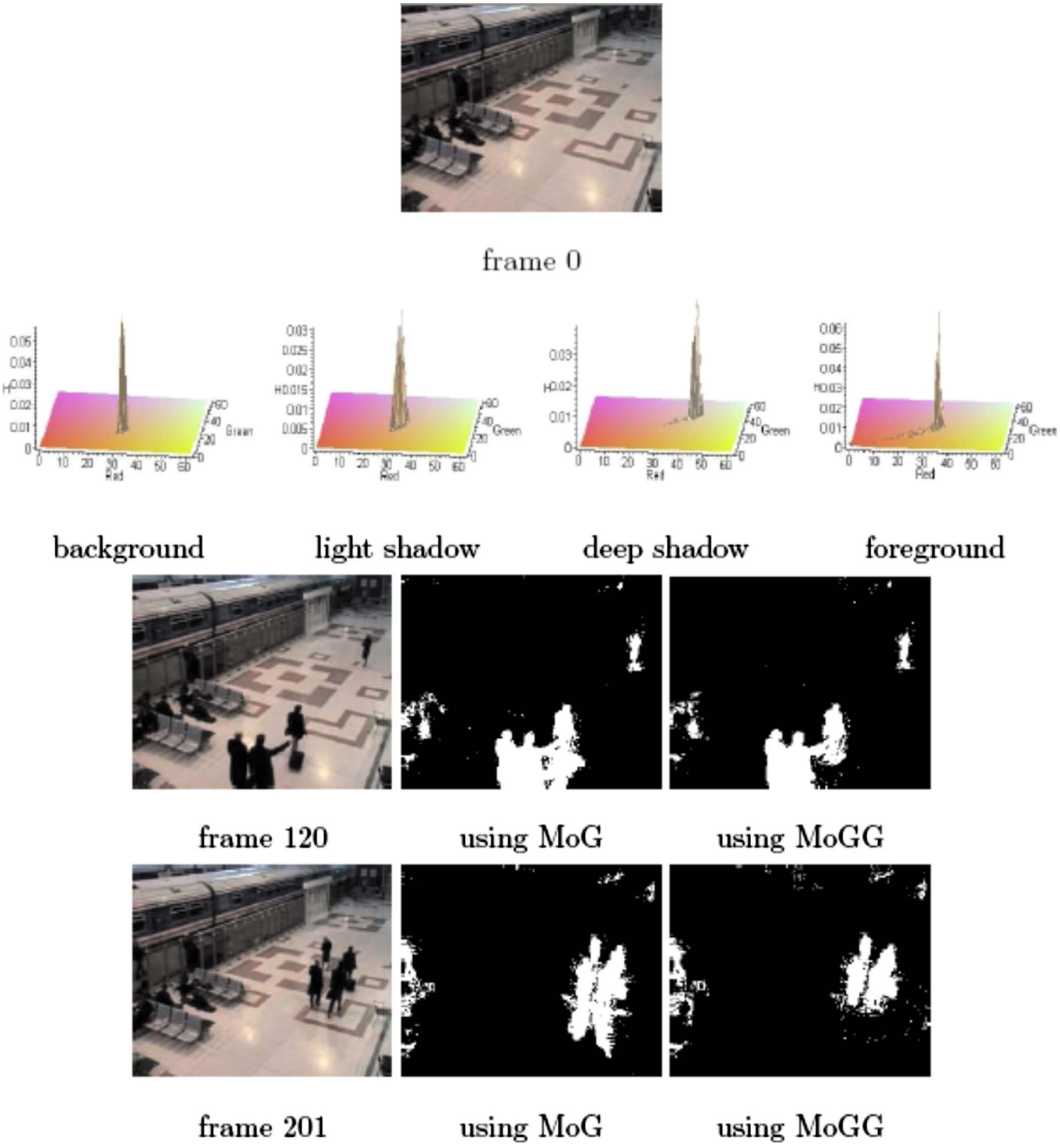


Fig. 6 Example of foreground segmentation in a video sequence containing shadows. The second row shows the different pixel histogram profiles in the sequence. The second and third rows show the result of foreground segmentation for two frames containing shadows using the MoG and MoGG models.

with the person entering the office, sitting down, and switching on the light on the desk for 2 sec (corresponding to five frames of the video sequence). The first row in Fig. 7 shows the first frame of the second video. The second row shows the different profiles for a pixel RG histogram characteristic, respectively, of the following events: 1. no event (background), 2. sudden illumination change, and 3. moving object (foreground). In the last row, we show the result of foreground segmentation on frame 306, which belongs to the set of frames where the sudden illumination change occurred; here also, we set $\mathbf{T}=0.3$. Again, the MoGG model showed less sensitivity to illumination

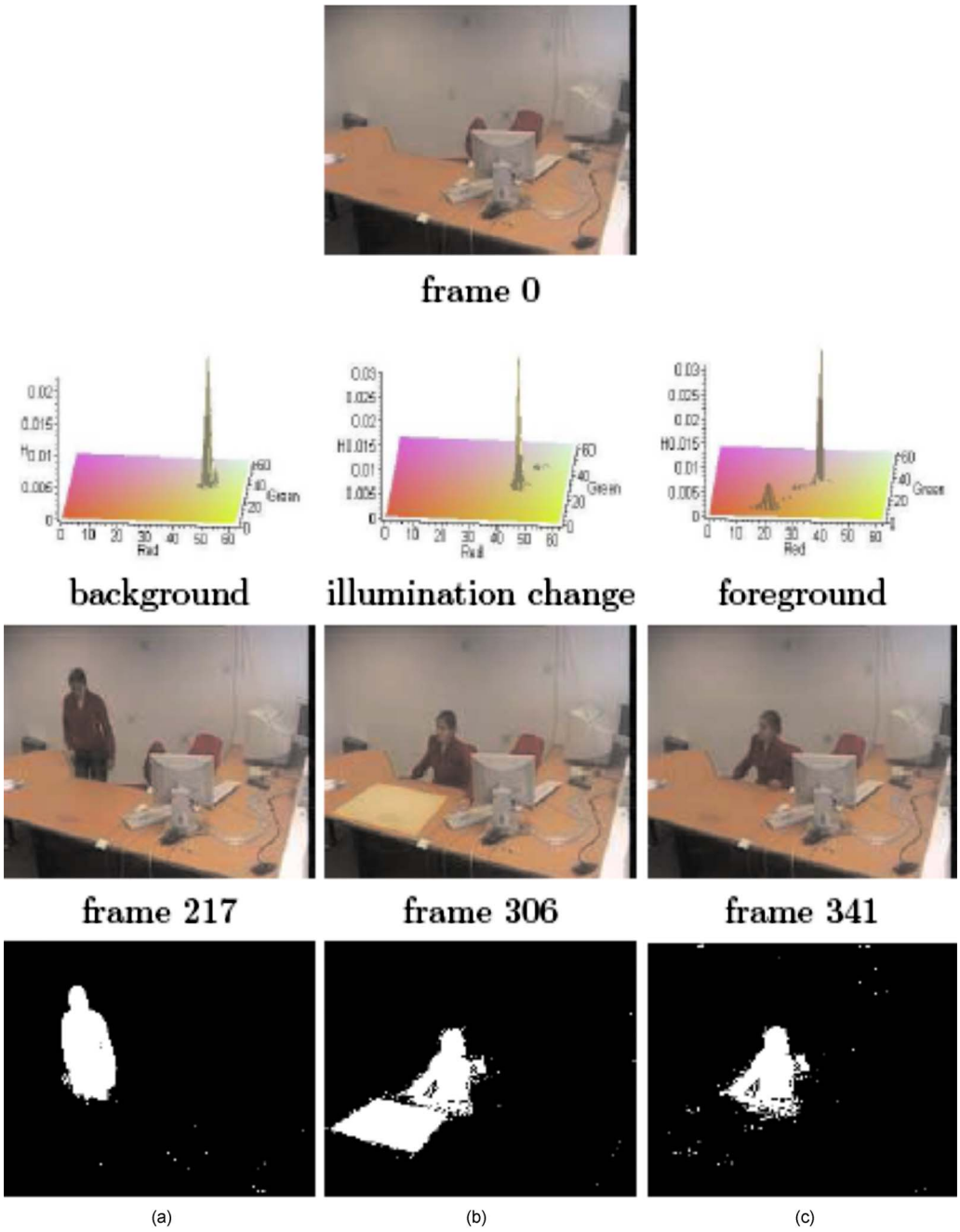


Fig. 7 Example of foreground segmentation in a video sequence containing a sudden illumination change. The first row shows the first frame of the sequence. The second row shows different pixel RG histogram profiles in the sequence. The third row shows frames 217, 306, and 341 of the sequence. The last row shows the result of foreground segmentation in: (a) frame 217 and (b) frame 306 using the MoG model, and (c) frame 306 using the MoGG model.

change than the MoG model. The major part of the lighted area has been segmented correctly to the background for the MoGG model, whereas the MoG assigned it to the foreground.

We should note, finally, that other experiments have also been conducted to evaluate the performance of the proposed model. The comparison to the MoG model showed that, globally, the MoGG model performs better than the MoG for preserving the optimal number of mixture components when the data contains a small proportion of noise or outliers. However, when the proportion of noise becomes sufficiently great to create distinct classes, the MoGG behaves almost the same as the MoG model. Another issue for the MoGG model concerns the computation time. It takes about one to two seconds to estimate the parameters of the MoGG than those of the MoG. Nevertheless, for a small cost in computation time, the MoGG model provides a better precision than the MoG model, which motivates its

use for segmentation and, more generally, for any classification problem involving mixture analysis.

5 Conclusions

We propose a new mixture model based on the formalism of general Gaussian distribution. We derive the maximum likelihood estimation and the MML criterion for the selection of model parameters. Our experiments show that the new model outperforms the MoG model, as it has more tolerance to noise and less sensitivity to mixture overfitting. Tests performed on noisy image and video foreground segmentation demonstrate the performance of the model. In future work, other applications of the proposed model will be investigated for problems involving robust mixture modeling.

Appendix A. Calculation of the Derivatives for F(Θ)

Note first that we can rewrite the likelihood functions in Eqs. (29)–(31) as follows:

$$p(\mathcal{X}_j|\vec{\mu}_j) \propto \prod_{i=1}^{n_j}\prod_{k=1}^{d} \exp\left[A(\lambda_{jk})\left|\frac{X_{ik}-\mu_{jk}}{\sigma_{jk}}\right|^{\lambda_{jk}}\right], \tag{38}$$

$$p(\mathcal{X}_j|\vec{\sigma}_j) \propto \prod_{i=1}^{n_j}\prod_{k=1}^{d} \frac{1}{\sigma_{jk}} \exp\left[A(\lambda_{jk})\left|\frac{X_{ik}-\mu_{jk}}{\sigma_{jk}}\right|^{\lambda_{jk}}\right], \tag{39}$$

$$p(\mathcal{X}_j|\vec{\lambda}_j) \propto \prod_{i=1}^{n_j}\prod_{k=1}^{d} \frac{\lambda_{jk}\sqrt{\frac{\Gamma(3/\lambda_{jk})}{\Gamma(1/\lambda_{jk})}}}{\Gamma(1/\lambda_{jk})} \times \exp\left[A(\lambda_{jk})\left|\frac{X_{ik}-\mu_{jk}}{\sigma_{jk}}\right|^{\lambda_{jk}}\right], \tag{40}$$

where n_j is the number of data vectors in the j'th class of the mixture. For the calculation of $|\mathbf{F}(\vec{\mu}_j)|$, the following derivatives are required:

$$-\frac{\partial \log[p(\mathcal{X}_j|\Theta)]}{\partial \mu_{jk}} = \frac{\lambda_{jk}}{\sigma_{jk}^{\lambda_{jk}}} \sum_{i=l}^{l+n_j-1} \text{sign}(X_{ik}-\mu_{jk})|X_{ik}-\mu_{jk}|^{\lambda_{jk}-1}, \tag{41}$$

where sign(x) is equal to 1, if $x \geq 0$, and −1, otherwise. We have also:

$$-\frac{\partial^2 \log[p(\mathcal{X}_j|\Theta)]}{\partial^2 \mu_{jk}} = \frac{\lambda_{jk}(1-\lambda_{jk})}{\sigma_{jk}^{\lambda_{jk}}} A(\lambda_{jk}) \sum_{i=l}^{l+n_j-1} |X_{ik}-\mu_{jk}|^{\lambda_{jk}-2}, \tag{42}$$

$$-\frac{\partial^2 \log[p(\mathcal{X}_j|\Theta)]}{\partial \mu_{jk_1}\partial \mu_{jk_2}} = 0. \tag{43}$$

Using Eqs. (41)–(43), we get Eq. (32). We perform the same calculations for $|\mathbf{F}(\vec{\sigma}_j)|$ and get the following equations:

$$-\frac{\partial \log[p(\mathcal{X}_j|\Theta)]}{\partial \sigma_{jk}} = \sum_{i=l}^{l+n_j-1}\left[\frac{1}{\sigma_{jk}} + \frac{\lambda_{jk}A(\lambda_{jk})}{\sigma_{jk}^{\lambda_{jk}+1}}|X_{ik}-\mu_{jk}|^{\lambda_{jk}}\right], \tag{44}$$

$$-\frac{\partial^2 \log[p(\mathcal{X}_j|\Theta)]}{\partial^2 \sigma_{jk}} = -\frac{n_j}{\sigma_{jk}^2} - A(\lambda_{jk})\frac{\lambda_{jk}(1+\lambda_{jk})}{\sigma_{jk}^{\lambda_{jk}+2}} \times \sum_{i=l}^{l+n_j-1} |X_{ik}-\mu_{jk}|^{\lambda_{jk}}, \tag{45}$$

$$-\frac{\partial^2 \log[p(\mathcal{X}_j|\Theta)]}{\partial \sigma_{jk_1}\partial \sigma_{jk_2}} = 0. \tag{46}$$

Using Eqs. (44)–(46), we get Eq. (33). Finally, for the calculation of $|\mathbf{F}(\vec{\lambda}_j)|$, we need the following equations:

$$-\frac{\partial \log[p(\mathcal{X}_j|\Theta)]}{\partial \lambda_{jk}} = -\frac{n_j}{\lambda_{jk}} + \frac{3n_j}{2\lambda_{jk}^2}[\Psi(3/\lambda_{jk}) - \Psi(1/\lambda_{jk})] - A(\lambda_{jk})\sum_{i=l}^{l+n_j-1}\left[\left|\frac{X_{ik}-\mu_{jk}}{\sigma_{jk}}\right|^{\lambda_{jk}} \times \log\left|\frac{X_{ik}-\mu_{jk}}{\sigma_{jk}}\right|\right] - B(\lambda_{jk})\sum_{i=l}^{l+n_j-1}\left|\frac{X_{ik}-\mu_{jk}}{\sigma_{jk}}\right|^{\lambda_{jk}}, \tag{47}$$

with

$$B(\lambda_{jk}) = \frac{\partial A(\lambda_{jk})}{\partial \lambda_{jk}} = \frac{1}{2}\left\{\log\left[\frac{\Gamma(3/\lambda_{jk})}{\Gamma(1/\lambda_{jk})}\right]\right\} - \frac{3\Psi(3/\lambda_{jk})}{2\lambda_{jk}} + \frac{3\Psi(1/\lambda_{jk})}{2\lambda_{jk}}. \tag{48}$$

We have also:

$$-\frac{\partial^2 \log[p(\mathcal{X}_j|\Theta)]}{\partial^2 \lambda_{jk}} = \frac{n_j}{\lambda_{jk}^2} + \frac{3n_j}{\lambda_{jk}^3}[\Psi(1/\lambda_{jk}) - \Psi(3/\lambda_{jk})] - \frac{9n_j}{\lambda_{jk}^4}\Psi'(3/\lambda_{jk}) - 2B(\lambda_{jk}) \sum_{i=l}^{l+n_j-1} \left[\left| \frac{X_{ik}-\mu_{jk}}{\sigma_{jk}} \right|^{\lambda_{jk}} \log \left| \frac{X_{ik}-\mu_{jk}}{\sigma_{jk}} \right| \right]$$
$$- A(\lambda_{jk}) \sum_{i=l}^{l+n_j-1} \left[\left| \frac{X_{ik}-\mu_{jk}}{\sigma_{jk}} \right|^{\lambda_{jk}} \left(\log \left| \frac{X_{ik}-\mu_{jk}}{\sigma_{jk}} \right| \right)^2 \right] - C(\lambda_{jk}) \sum_{i=l}^{l+n_j-1} \left| \frac{X_{ik}-\mu_{jk}}{\sigma_{jk}} \right|^{\lambda_{jk}} + \frac{3n_j}{2\lambda_{jk}^4}\Psi'(1/\lambda_{jk}), \quad (49)$$

with

$$C(\lambda_{jk}) = \frac{\partial^2 A(\lambda_{jk})}{\partial^2 \lambda_{jk}}$$
$$= \frac{1}{2}A(\lambda_{jk})\left\{ \frac{1}{2\lambda_{jk}^2}[\Psi(1/\lambda_{jk}) - 3\Psi(3/\lambda_{jk})] + \frac{1}{2\lambda_{jk}^3}[9\Psi'(3/\lambda_{jk}) - \Psi'(1/\lambda_{jk})] \right\}$$
$$+ \frac{1}{2}B(\lambda_{jk})\left\{ \log\left[\frac{\Gamma(3/\lambda_{jk})}{\Gamma(1/\lambda_{jk})} \right] - \frac{3\Psi(3/\lambda_{jk})}{2\lambda_{jk}} + \frac{\Psi(1/\lambda_{jk})}{2\lambda_{jk}} \right\}, \quad (50)$$

where we have

$$\Psi(x) = \frac{\partial \log[\Gamma(x)]}{\partial x} \quad \text{and} \quad \Psi'(x) = \frac{\partial^2 \log[\Gamma(x)]}{\partial^2 x}.$$

In addition, we have:

$$-\frac{\partial^2 \log[p(\mathcal{X}_j|\Theta)]}{\partial \lambda_{jk_1} \partial \lambda_{jk_2}} = 0. \quad (51)$$

This yields Eq. (34).

Appendix B: Calculation of the Derivatives for the Estimation of λ_{jk} Using the Maximum Likelihood Method

The following equations are required in Eq. (11):

$$\frac{\partial \log[p(\mathcal{X}|\Theta)]}{\partial \lambda_{jk}} = \sum_{i=1}^{N} p(j|\vec{X}_i) \frac{\partial \log[p(\vec{X}_i|\Theta)]}{\partial \lambda_{jk}}, \quad (52)$$

$$\frac{\partial^2 \log[p(\mathcal{X}|\Theta)]}{\partial^2 \lambda_{jk}} = \sum_{i=1}^{N} p(j|\vec{X}_i) \frac{\partial^2 \log[p(\vec{X}_i|\Theta)]}{\partial^2 \lambda_{jk}} + \frac{\partial p(j|\vec{X}_i)}{\partial \lambda_{jk}} \frac{\partial \log[p(\vec{X}_i|\Theta)]}{\partial \lambda_{jk}}, \quad (53)$$

where we have:

$$\frac{\partial (j|\vec{X}_i)}{\partial \lambda_{jk}} = p(j|\vec{X}_i)[1 - p(j|\vec{X}_i)] \frac{\partial \log[p(\vec{X}_i|\Theta)]}{\partial \lambda_{jk}}. \quad (54)$$

Note finally that the terms

$$\frac{\partial \log[p(\vec{X}_i|\Theta)]}{\partial \lambda_{jk}} \quad \text{and} \quad \frac{\partial^2 \log[p(\vec{X}_i|\Theta)]}{\partial^2 \lambda_{jk}}$$

in Eqs. (52) and (53) can be directly deduced from Eqs. (47) and (49).

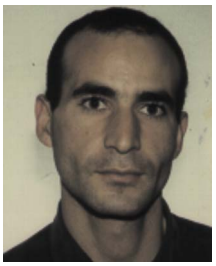
Acknowledgments

The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC), Bell Canada's support through its Bell University Laboratories Research and Development programs and a start-up grant from Concordia University. We thank the reviewers for their helpful comments.

References

1. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: image segmentation using expectation-maximization and its application to image querying," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(8), 1026–1038 (2002).
2. J. Cheng, J. Yang, Y. Zhou, and Y. Cui, "Flexible background mixture models for foreground segmentation," *Image Vis. Comput.* **24**(5), 473–482 (2006).
3. S. J. McKenna, Y. Raja, and S. Gong, "Tracking colour objects using adaptive mixture models," *Image Vis. Comput.* **17**(3,4), 225–231 (1999).
4. J. Puzicha, T. Hofmann, and J. M. Buhmann, "Discrete mixture models for unsupervised image segmentation," *Proc. DAGM-Symp.*, pp. 135–142 (1998).
5. X. Yang and S. M. Krishnan, "Image segmentation using finite mixtures and spatial information," *Image Vis. Comput.* **22**(9), 735–745 (2004).
6. G. McLachlan and D. Peel, *Finite Mixture Models*, Wiley Series in Probability and Statistics, Wiley and Sons, New York (2000).
7. A. E. Raftery and J. D. Banfield, "Model-based Gaussian and non-Gaussian clustering," *Biometrics* **49**, 803–821 (1993).
8. K. Kokkinakis and A. K. Nandi, "Exponent parameter estimation for generalized Gaussian probability density functions with application to speech modelling," *Signal Process.* **85**(9), 1852–1858 (2005).
9. M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance," *IEEE Trans. Image Process.* **11**(2), 146–158 (2002).
10. K. Sharifi and A. Leon-Garcia, "Estimation of shape parameter for generalized Gaussian distibution in subband decomposition of video," *IEEE Trans. Circuits Syst. Video Technol.* **5**(1), 52–56 (1995).
11. T. Pham and R. J. P. deFigueiredo, "Maximum likelihood estimation of a class of non-Gaussian densities with application to l_p deconvolution," *IEEE Trans. Acoust., Speech, Signal Process.* **37**(1), 73–82 (1989).
12. R. L. Joshi and T. R. Fischer, "Comparison of generalized Gaussian and Laplacian modelling in DCT image coding," *IEEE Signal Process. Lett.* **2**(5), 81–82 (1995).
13. M. Baccar, L. A. Gee, and M. A. Abidi, "Reliable location and regression estimates with application to range image segmentation," *J. Math. Imaging Vision* **11**(3), 195–205 (1999).
14. R. A. Baxter and J. J. Olivier, "Finding overlapping components with MML," *Stat. Comput.* **10**(1), 5–16 (2000).
15. H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control* **19**(6), 716–723 (1974).

16. J. Rissanen, "Modeling by shortest data description," *Automatica* **14**(5), 465–471 (1978).
17. M. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(3), 381–396 (2002).
18. K. T. Fang, S. Kotz, and K. W. Ng, *Symmetric Multivariate and Related Distributions*, Chapman and Hall, New York (1990).
19. J. K. Lindsey, "Multivariate elliptically contoured distributions for repeated measurements," *Biometrics* **55**, 1277–1280 (1999).
20. G. E. P. Box and P. W. Tidwell, "Transformation of independent variables," *Technometrics* **4**(4), 531–550 (1962).
21. C. S Wallace and D. M. Boulton, "An information measure for classification," *Comput. J.* **11**(2), 195–209 (1968).
22. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley, New York (2001).
23. Y. Weiss and E. H. Adelson, "A unified mixture framework for motion segmentation: incorporating spatial coherence and estimating the number of models," *Proc. IEEE Conf. Computer Vision Patt. Recog.*, pp. 321–326 (1996).
24. C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 747–757 (2000).
25. N. Bouguila and D. Ziou, "Online clustering via finite mixtures of Dirichlet and minimum message length," *Eng. Applic. Artif. Intell.* **19**(4), 371–379 (2006).
26. D. M. Titterington, "Recursive parameter estimation using incomplete data," *J. R. Stat. Soc.* Ser. B **46**(2), 257–267 (1984).
27. J. F. Yao, "On recursive estimation in incomplete data models," *Statistics* **34**, 27–51 (2000).



Mohand Saïd Allili received the BEng degree in computer science from Mouloud Mammeri de Tizi-Ouzou University (Algeria) in 2001, and MSc degree in computer science from Sherbrooke University in 2004. Since 2004, he has been pursuing PhD studies at University of Sherbrooke under the supervision of Professor Djemel Ziou. His primary research interests include deformable models and PDEs applied to segmentation and tracking, computer vision, pattern recognition, and information retrieval.

Nizar Bouguila received the engineering degree from the University of Tunis in 2000, and the MSc and PhD degrees from the University of Sherbrooke in 2002 and 2006, respectively, all in computer science. He is currently an assistant professor with the Concordia Institute for Information Systems Engineering (CIISE) at Concordia University, Montreal, Quebec, Canada. His research interests include image processing, machine learning, 3-D graphics, computer vision, and pattern recognition.

Djemel Ziou received the BEng degree in computer science from University of Annaba (Algeria) in 1984, and PhD degree in computer science from the Institut National Polytechnique de Lorraine (INPL), France, in 1991. From 1987 to 1993, he served as lecturer at several universities in France. During the same period, he was a researcher in the Centre de Recherche en Informatique de Nancy (CRIN), and the Institut National de Recherche en Informatique et Automatique (INRIA) in France. He is currently a full professor in the Department of Computer Science at the University of Sherbrooke in Canada. He has served on numerous conference committees as member or chair. He heads the MOIVRE laboratory and the CoRIMedia consortium, which he founded. His research interests include image processing, information retrieval, computer vision, and pattern recognition.