Finite Generalized Gaussian Mixture Modeling and Applications to Image and Video Foreground Segmentation

Mohand Saïd Allili
Department of computer science,
University of Sherbrooke,
J1K 2R1, Sherbrooke (QC), Canada.
Email:ms.allili@usherbrooke.ca.

Nizar Bouguila
CIISE, Concordia University,
1515 St. Catherine Street West, EV007.632,
Montreal (QC), H3G 2W1, Canada.
Email: bouguila@ciise.concordia.ca.

Djemel Ziou
Department of computer science,
University of Sherbrooke,
J1K 2R1, Sherbrooke (QC), Canada.
Email:d.ziou@usherbrooke.ca.

Abstract

In this paper, we propose a finite mixture model of generalized Gaussian distributions (GDD) for robust segmentation and data modeling in the presence of noise and outliers. The model has more flexibility to adapt the shape of data and less sensibility for over-fitting the number of classes than the Gaussian mixture. In a first part of the present work, we propose a derivation of the Maximum-Likelihood estimation of the parameters of the new mixture model and we propose an information-theory based approach for the selection of the number of classes. In a second part, we propose some applications relating to image, motion and foreground segmentation to measure the performance of the new model in image data modeling with comparison to the Gaussian mixture.

keywords: *mixture of General Gaussians (MoGG), MML, image, motion, foreground segmentation.*

1. Introduction

Finite Gaussian mixture models are widely used in various fields of computer vision and image processing [8, 16, 17, 22]. The adoption of this model-based approach to data clustering and modeling brings important advantages: for instance, the selection of the number of classes or the assessment of the validity for a given model can be addressed in a formal way [16]. However, it is well-known that the Gaussian density has some drawbacks such as the rigidity of its shape which prevents it from having a good approximation to data with outliers [18]. For this reason, many researchers, especially in signal processing, start to use the generalized Gaussian density (GGD) for its flexibility to model data with different shapes. The GGD has been used recently in speech modeling [11], Wavelet-Based Texture Retrieval [9] and image and video coding [13, 19]. Recently, the authors in [2] used the GGD to estimate reliable location and regression coefficients in data containing noise or outliers. The majority of these works have focused on the one component version of the GGD where the data are uni-modal; i.e: modeled by using a single component. In applications where the data have many clusters, like image segmentation, a multi-component probabilistic model representation such as mixture modeling is required. The focus of the present paper is the utilization of the GGDs for a robust mixture modeling of multi-modal data in the presence of noise and outliers where the new mixture model brings accuracy and robustness for data modeling.

In the paper, a mixture model by using the formalism of General Gaussian distributions is proposed and we denote it by: MoGG (by reference to the commonly notation used in literature for the mixture of Gaussian distributions: MoG). Having the flexibility of fitting the shape of data better than a mixture of Gaussian distributions (MoG), the new model permits for having a robust mixture representation for noisy data. By robustness, we mean the ability of the model to represent accurately the shape of data with less sensibility to overfitting the number of classes in the presence of noise or outliers. In this regard, we show that the MoGG formalism offers a better performance than the MoG, which is demonstrated on applications related to image, motion and

Fourth Canadian Conference on Computer and Robot Vision(CRV'07)
0-7695-2786-8/07 $20.00

foreground segmentation for video object tracking.

This paper is organized as follows. In section (2), we propose the Maximum-Likelihood estimation of the parameters of the MoGG. In section (3), we propose the model selection by using the MML criterion. In section (4), we test the performance of the new model on examples of segmentation. We end the paper with conclusions and some perspectives.

2 Mixture of GGDs (MoGG)

The 1-dimensional GGD for a variable $X \in \Re$ is defined as follows [2]:

$$p(X/\mu,\sigma,\lambda) = A(\lambda)\exp\left(-B(\lambda)\left|\frac{X-\mu}{\sigma}\right|^{\lambda}\right) \qquad (1)$$

in which: $A(\lambda) = \frac{\lambda\sqrt{\frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)}}}{2\sigma\Gamma(1/\lambda)}$, $B(\lambda) = \left[\frac{\Gamma(3/\lambda)}{\Gamma(1/\lambda)}\right]^{\frac{\lambda}{2}}$ and $\Gamma(\cdot)$ denotes the gamma function given by: $\Gamma(z) = \int_0^\infty p^{z-1}e^{-p}dp$, where z and p are real variables. The parameters μ and σ are the pdf mean and standard deviation. Finally, the parameter $\lambda \geq 1$ controls the tails of the pdf and determines whether the latter is peaked or flat: The larger the value of λ is, the flatter is the pdf; the smaller is λ, the more peaked is the pdf . This gives a flexibility to the pdf to fit the shape of heavy-tailed data produced by the presence of noise or outliers. Note that the Laplacian and the Gaussian distributions are particular cases for the GGD where $\lambda = 1$ and 2 respectively.

2.1 MoGG for multi-dimensional data

A multi-dimensional generalization of the function (1) is not trivial. In the past, multivariate distributions where the Mahalanobis distance in the exponent is powered by a real parameter have been proposed [10, 15]. However, this 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 (see fig. (1)). Researches in the past showed the performance of non-linear regression models where the input variables have different powers in achieving a good approximation to data [7]. In our case, we seek to build a multi-dimensional GGD that has a different shape parameter for each dimension. In practice, however, this 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 [14]. Having a d-dimensional vector $\vec{X} = (X_1, ..., X_d)$, the probability of the vector $\vec{X}$ with a GGD is, then, given by:

$$p(\vec{X}/\vec{\mu},\vec{\sigma},\vec{\lambda}) = \prod_{k=1}^{d} A(\lambda_k)\exp\left(-B(\lambda_k)\left|\frac{X_k-\mu_k}{\sigma_k}\right|^{\lambda_k}\right) \qquad (2)$$

in which: $\vec{\mu} = (\mu_1, ..., \mu_d)$ and $\vec{\sigma} = (\sigma_1, ..., \sigma_d)$. 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 (see fig. (1)).

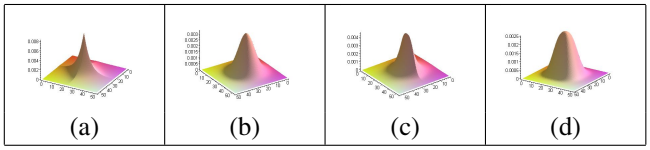

Figure 1. Different representations of a 2-dimensional GGD according to the parameter $\vec{\lambda}$. $\vec{\mu}$ and $\vec{\sigma}$ are fixed to $(23, 24)$ and $(7, 7)$ respectively. (a) $\vec{\lambda} = (1.1, 1.1)$, (b) $\vec{\lambda} = (2, 2)$, (c) $\vec{\lambda} = (1.1, 2.8)$ and (d) $\vec{\lambda} = (2.8, 2.8)$.

2.2 Maximum Likelihood estimation of the MoGG model

A generalized Gaussian mixture with M components is defined as:

$$p(\vec{X}/\Theta) = \sum_{j=1}^{M} p_j p(\vec{X}/\mu_j,\sigma_j,\lambda_j) \qquad (3)$$

with p_j, $j = 1, \ldots, M$, are the mixing parameters where $0 < p_j \leq 1$ and $\sum_{j=1}^{M} p_j = 1$, and $p(\vec{X}/\mu_j,\sigma_j,\lambda_j)$, $j = 1, \ldots, M$, are the conditional probabilities. The set of parameters of the mixture with M classes is defined by $\Theta = \bigcup_{i=1}^{4}\xi_i$, where $\xi_1 = \{\vec{\mu}_1, ..., \vec{\mu}_M\}$, $\xi_2 = \{\vec{\sigma}_1, ..., \vec{\sigma}_M\}$, $\xi_3 = \{\vec{\lambda}_1, ..., \vec{\lambda}_M\}$ and $\xi_4 = \{p_1, ..., p_M\}$ is the set of mixing parameters. Two important problems in the case of finite mixture models are the estimation of the vector 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 mostly via the Maximum-Likelihood estimation (MLE). Let us consider a set of data $\mathcal{X} = \{\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_N\}$. The MLE consists of getting the parameters of the mixture that maximizes the log-likelihood function: $\log(p(\mathcal{X}/\Theta)) = \log(\prod_{i=1}^{N} p(\vec{X}_i/\Theta))$. The problem of estimation, thus, becomes:

$$\hat{\Theta} = argmax_{\Theta}\big(\log(p(\mathcal{X}/\Theta))\big) \qquad (4)$$

with the constraint: $\sum_{j=1}^{M} p_j = 1$. For the moment, we suppose the number of mixture M is known. In the next

section, we will see how to estimate it automatically. The above constraint permit to take into consideration the property of the mixing parameters p_j to sum to 1. Using the Lagrange multipliers, we maximize the following function:

$$\Phi(\mathcal{X}, \Theta, \Lambda) = \log(p(\mathcal{X}/\Theta)) + \Lambda(1 - \sum_{j=1}^{M} p_j) \quad (5)$$

where Λ is the Lagrange multiplier. Then, minimizing according to the mixture parameters Θ and the Lagrange multiplier Λ yields $\forall i = 1, \ldots N, \forall j = 1, \ldots M, \forall k = 1, \ldots d$, we have:

$$\hat{p}_j = \frac{1}{N} \sum_{i=1}^{N} p(j/\vec{X}_i) \quad (6)$$

$$\hat{\mu}_{jk} = \frac{\sum_{i=1}^{N} p(j/\vec{X}_i) |X_{ik} - \mu_{jk}|^{\lambda_{jk}-2} X_{ik}}{\sum_{i=1}^{N} p(j/\vec{X}_i) |X_{ik} - \mu_{jk}|^{\lambda_{jk}-2}} \quad (7)$$

$$\hat{\sigma}_{jk} = \left[\frac{\lambda_{jk} A(\lambda_{jk}) \sum_{i=1}^{N} p(j/\vec{X}_i) |X_{ik} - \mu_{jk}|^{\lambda_{jk}}}{\sum_{i=1}^{N} p(j/\vec{X}_i)} \right]^{1/\lambda_{jk}} \quad (8)$$

where $p(j/\vec{X}_i)$ is the posterior probability of the component j given by: $p(j/\vec{X}_i) = \frac{p_j p(\vec{X}_i/j)}{\sum_{l=1}^{M} p_l p(\vec{X}_i/l)}$. For the parameter $\vec{\lambda}_j = (\lambda_{j1}, ..., \lambda_{jd})$, we use the Newton-Raphson method, which is based on developing to power series: $\frac{\partial \log(p(\mathcal{X}/\Theta))}{\partial \lambda_{jk}}$ according to λ_{jk} and use the following updating equation:

$$\hat{\lambda}_{jk} \simeq \lambda_{jk} - \left(\frac{\partial^2 \log(p(\mathcal{X}/\Theta))}{\partial \lambda_{jk}^2} \right)^{-1} \frac{\partial \log(p(\mathcal{X}/\Theta))}{\partial \lambda_{jk}} \quad (9)$$

The calculation of the terms $\frac{\partial \log(p(\mathcal{X}/\Theta))}{\partial \lambda_{jk}}$ and $\frac{\partial^2 \log(p(\mathcal{X}/\Theta))}{\partial \lambda_{jk}^2}$ can be obtained in the reference [1].

3 Model selection by using the MML

For the selection of the number of classes in mixture models, various approaches have been suggested in the past, such as the Akaike's information criterion (AIC), the Minimum Description Length (MDL) and the Laplace Empirical Criterion (LEC) [16]. However, several works have shown the performance of the Bayesian methods for the selection of mixture models [14]. Among the most used criteria, we find the Minimum Message Length (MML)[3, 6, 14] that we will adopt in the present work. By using the MML, the optimal number of classes of the mixture, M, is the one that minimizes the following equation:

$$MML(M) \simeq -\log(p(\Theta)) - \log(p(\mathcal{X}/\Theta)) + \frac{1}{2}\log|\mathbf{F}(\Theta)| - \frac{1}{2}\log(12) + \frac{N_p}{2} \quad (10)$$

where $p(\Theta)$ is the prior probability, $p(\mathcal{X}/\Theta)$ the data likelihood and $|\mathbf{F}(\Theta)|$ is the determinant of the Fisher information matrix of minus the log-likelihood. The symbol N_p designates the number of parameters to be estimated, which is equal to $M(3d + 1)$ in our case ($3d$ parameters for each mixture component and M mixing parameters). The estimation of the number of classes is carried out by finding the minimum with regard to Θ of the message length.

To calculate the prior distribution $p(\Theta)$, we assume the parameters of the different classes are *a priori* independent. We know that for the vector $(p_1, ..., p_M)$ we have $\sum_{j=1}^{M-1} p_j < 1$, then a natural choice as prior for this vector is the Dirichlet distribution:

$$p(\xi_4) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \quad (11)$$

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

$$p(\xi_4) = (M - 1)! \quad (12)$$

Note that this unform prior is defined over the $(M - 1)$-dimensional region of hyper-volume $1/(M - 1)!$. It represents the inverse of the hyper-volume so that it integrates to 1. For the parameters ξ_1, ξ_2 and ξ_3 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 doesn't provide any knowledge about the parameters of a class j, $j \neq i$. Suppose that $\vec{\sigma} = (\sigma_1, \ldots, \sigma_d)$ and $\vec{\mu} = (\mu_1, \ldots, \mu_d)$ are the standard deviation and the mean vectors of the entire population (all the set of data $\mathcal{X}$). For ξ_1, knowing that each μ_{jk} is chosen to be uniform in the region within one standard deviation of the population mean; i.e. $\mu_k - \sigma_k \leq \mu_{jk} \leq \mu_k + \sigma_k$, the prior is given by the following equation:

$$p(\xi_1) = \prod_{j=1}^{M} \prod_{k=1}^{d} p(\mu_{jk}) = \prod_{k=1}^{d} \frac{1}{(2\sigma_k)^M} \quad (13)$$

For ξ_2, knowing that $0 \leq \sigma_{jk} \leq \sigma_k$, $k = 1, \ldots, d$, then we have:

$$p(\xi_2) = \prod_{j=1}^{M} \prod_{k=1}^{d} \frac{1}{\sigma_k} = \prod_{k=1}^{d} \frac{1}{\sigma_k^M} \quad (14)$$

For the last parameter ξ_3, for each λ_{jk} we adopt a uniform distribution $\mathcal{U}[0, h]$ where h is the maximum value permitted for each parameter λ_{jk}, $j = 1, ..., M$, $k = 1, ..., d$, and we obtain:

$$p(\xi_3) = \prod_{j=1}^{M} \prod_{k=1}^{d} p(\lambda_{jk}) = \frac{1}{h^{M \cdot d}} \quad (15)$$

0-7695-2786-8/07 $20.00

Finally, by using eq. (12), (14), (13) and (15) we obtain the following prior or the MoGG parameters:

$$p(\Theta) = \frac{(M-1)!}{(2h)^{M \cdot d}} \prod_{k=1}^{d} \frac{1}{\sigma_k^{2M}} \tag{16}$$

To calculate the determinant of the Fisher information matrix $|\mathbf{F}(\Theta)|$, we use the following simplification:

$$|\mathbf{F}(\Theta)| \simeq |\mathbf{F}(\xi_4)| \prod_{j=1}^{M} |\mathbf{F}(\vec{\mu}_j)||\mathbf{F}(\vec{\sigma}_j)||\mathbf{F}(\vec{\lambda}_j)| \tag{17}$$

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 the component j of the mixture [1].

4 Experiments

In the following section, we show two applications of the proposed model relating to noisy image and video foreground segmentation. For each application, we developed an algorithm for the MOGG model selection.

4.1 Image and motion segmentation

In the past, Gaussian mixture models have been used for segmentation where the aim is to build a partition of the image with each region represented by a mixture component [17]. It remains, however, that with noisy data, the image histogram may be heavy-tailed and the mixture may lose accuracy by overt-fitting the number of regions in the image. This is undesirable in applications such as medical image processing, where precision is a strict requirement. Having a robust mixture model, which is less prone to over-fitting, is a solution that we propose in the present work. For the MoGG model selection, we use the following algorithm:

Algorithm1

a) For each candidate value of M do:

1) Initialization of the MoGG parameters.

2) Repeat

- E-step: Compute the posterior probabilities: $p(j/\vec{X}_i)$.

- M-step: Update the parameters by using eqs. (6), (7), (8) and (9).

Until: $\|\Theta_{n+1} - \Theta_n\| < \epsilon$.

3) Calculate the MML criterion using eq. (10).

b) Select M such that: $M = argmin_M MML(M)$.

The convergence of the algorithm is detected in step 2 where the distance between the resultant parameters between two successive iterations n and $n+1$ is smaller than a threshold ϵ, i.e: $\|\Theta_{n+1} - \Theta_n\| < \epsilon$, or a maximum number of iterations is reached. We note that the initialization step is performed for the mixture parameters by using the K-Means algorithm [4] where the shape parameters λ_{jk}, $j = 1, ..., M$, $k = 1, ..., d$, are initialized to 2. In the following experiments we set the value ϵ to 0.01.

In fig. (2), a comparison between a MoG and MoGG histogram fitting is shown for a synthetic image composed of 4 regions with an added noise. The noise has a normal distribution $\mathcal{N}(0, \sigma_n)$ with the standard deviation $\sigma_n = 3$. The original and noisy images are shown on the top row of the figure. In the bottom row, we show the histograms of these images. Notice the effect of adding noise on the histogram of the image which resulted in creating heavy tails and increasing the overlapping between the histogram modes. The value of the MML obtained for the MoG and the MoGG models is shown in the top row of fig. (3) for the original image of fig. (2) (left graph) and the noisy version of it (right graph). Notice that the MML has favored 5 regions for the MoG model and only 4 regions for the MoGG model. The over-fitting of the MoG is due to the accumulated noise in the tails of the histogram which resulted in a new component for the MoG (see the bottom row of fig. (3)).

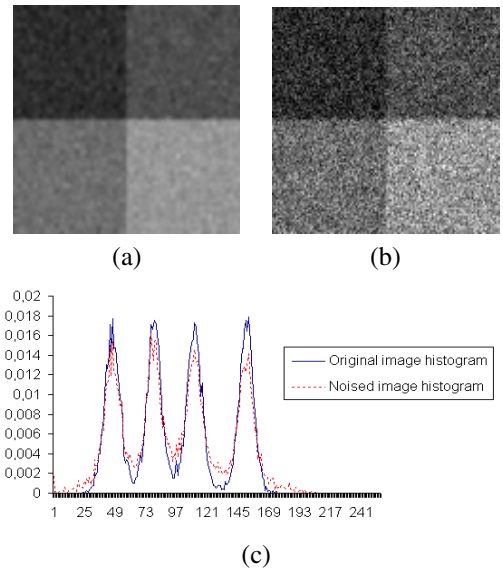

Figure 2. Effect of adding noise on the image histogram: (a) and (b) represent respectively a synthetic image and noisy version of it, (c) represents the histograms of the original image (continuous blue line) and the noisy version of it (dashed red line).

0-7695-2786-8/07 $20.00

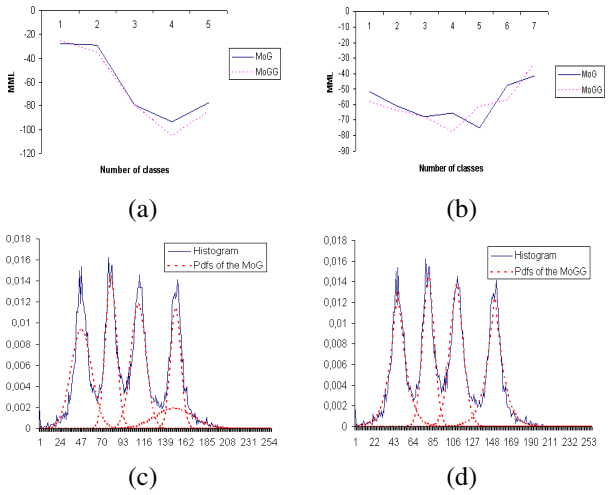

Figure 3. Example of histogram fitting by using MoG and MoGG models: (a) and (b) show respectively the value of the MML on the original and noisy version of it, (c) and (d) show the histogram fitting by using respectively a 5-component MoG and 4-component MoGG models.

In fig. (4), we show an example of real world image segmentation where we added a Gaussian noise, with $\sigma_n = 3$, to part of the image (hand image) and to the whole image (face image). We remark that the regions of interest (the hand in the first image and the head in the second image) have been severely over-segmented by using the MoG model which over-fitted the real number of regions. By using the MoGG model, the estimated number of components is the same as the real one and the image was less over-segmented.

In fig. (5), we show and example of motion field segmentation. The 2-D measurement of the motion field is described by a vector of two components $V(u, v)$. In motion segmentation, several objects may move differently in an image sequence and the aim is to localize each object by differentiating its motion from the rest of the objects and the background. The motion field [1] between the original pair of images is shown in fig. (5.c). Its segmentation by using a MoG model resulted in 3 regions (background, the tiger and the toy objects) as shown in fig. (5.g). The motion field obtained when we added partially a Gaussian noise with $\sigma_n = 3$ to the image is shown in fig. (5.f). The noisy image segmentation by using a MoG and MoGG models is shown respectively in fig. (5.h) and (5.i). After calculating the MML, the MoG and MoGG ended up with 4 and 3 classes respectively. The over-fitting of of the MoG is clearly apparent in fig. (5.h) where the region of the tiger has been more over-segmented than by using the MoGG model.

[1] We used the algorithm of Horn and Schunk [12] to calculate the motion field.

4.2 Foreground segmentation in video sequences

In the past, adaptive Gaussian mixture models have been used for modeling dynamically the foreground and background pixel distribution [8, 20]. This approach assumes a video sequence taken by a static camera and the problem is to segment the foreground (moving objects) by constructing a mixture model for the background, observed over a period of time, and to decide whether a pixel in an input frame belongs to the foreground or the background. The approach showed promising results in [8, 20]. However, major issues remain a real challenge like handling sudden illuminations changes, slow moving objects, shadows and other phenomenons that produce non-stationary backgrounds which may cause erroneous classification of pixels to the foreground.

The focus of this section is to compare the performance of the MoGG model versus the MoG's for foreground segmentation. Although we don't expect to resolve all the aforementioned issues, we aim to improve the accuracy and robustness of background modeling, and thereby reduce pixel misclassification. In [8], the authors proposed a flexible mixture model where the value of each pixel throughout a video sequence is modeled by learning online a MoG. In each mixture, the components that occur frequently in the sequence (with high *a priori* probability and small variance) model the background. Formally, having a mixture with M components ordered by the value of this term $p_j/||\vec{\sigma}_j||$, $j = 1, ..., M$, the first B components are chosen as a model for the background where:

$$B = argmin_b(\sum_{j=1}^{b} p_j > T) \tag{18}$$

with T is a threshold and $||\cdot||$ designates the norm of a vector. In what follows, we propose to build an online estimation for the parameters of MoGG model. Afterwards, we propose to measure the performance with the online version of MoG in [8]. Following the work of [6, 21, 23], we propose the following iterative formulas for the online estimation of the MoGG parameters for reach pixel (x, y):

$$p_j^{(n+1)} = p_j^{(n)} + \beta_n \cdot t_j(\vec{X}^{(n+1)}) \left(\vec{X}^{(n+1)} - p_j^{(n)}\right) \tag{19}$$

$$\theta_j^{(n+1)} = \theta_j^{(n)} + \beta_n \cdot t_j(\vec{X}^{(n+1)}) \frac{\partial log(p(\vec{X}^{(n+1)}/\theta_j))}{\partial \theta_j} \tag{20}$$

where $t_j(\vec{X}^{(n+1)}) = p(j/\vec{X}^{(n+1)})$ and β_n represents any sequence of positive numbers which decreases to zero. The

derivatives in the equation (20) with respect to the different parameters of the mixture distributions, $\vec{\mu}_j$, $\vec{\sigma}_j$ and $\vec{\lambda}_j$ are given in [1]. The algorithm for foreground segmentation works as following. Having an input frame of the sequence, for each pixel check if its new value match one of the components of the MoGG mixture. A match to a component occurs when the value of the pixel $\vec{X}$ falls within 2 standard deviation of the mean of the component. If no mach occurs, create a new component for the mixture with the mean equal to the new value of the pixel. Calculate the MML for the new mixture model: if (MML(M) $>$ MML(M+1)) then M $\leftarrow$ M+1, else: update the old mixture parameters with eqs. (19) and (20). If $\exists p_j < 0$, discard the component j of the mixture and put M $\leftarrow$ M-1.

The rationale behind introducing the MML for the MoGG model selection is to build a background model that is more robust to outliers caused by phenomenons like sudden illumination change in indoor scenes or non-stationary backgrounds produced by swaying tree branches or shadows in outdoor scenes. Having the GGDs which have the ability to adapting the shape of data more easily than the Gaussian can reduce over-fitting the mixture model, and thereby mitigate the impact of the above-mentioned phenomenons. The complete tracking algorithm is summed up in the following script:

Algorithm2
a) Mixture initialization for each pixel $\mathbf{x}$*=(x,y):*
1) Initialize: $M = 1$, $p_j = 1/M$, $\sigma_{jk} = 0.2$, $\vec{\mu}_j = \vec{X}^{(0)}(\mathbf{x})$ *and* $\lambda_{jk} = 2$, $\forall j = 1, ..., M$, $\forall k = 1, ..., d$.
b) *For a new frame* $(n+1)$ *of the sequence, for each pixel* $\boldsymbol{x}$, *having a new value of the pixel* $\vec{X}^{(n+1)}$:
- Verify if there is a match for pixel value $\vec{X}^{(n+1)}$.
- Update its MoGG parameters by using eq. (19) and (20).
- Extract the foreground object according to eq. (18).

In the following experiment, we show a comparison of the MoGG and the MoG models in an example of foreground segmentation. In fig. (6), the foreground is segmented for an indoor scene of a video sequence where the moving objects have shadows. The results were obtained by setting the parameter T to 0.3. In the first row of the figure we show the first frame of the video sequence containing 966 frames. In the second row, we show the 2D histogram of the Red and Green color of 3 typical pixels over a video sequence. The 3 pixels are characterized respectively by the following occurred events: 1) No event (background), 2) shadow, and 3) moving object (foreground). In the third and fourth rows, we show the result of foreground segmentation for the frames 120 and 210 of the sequence. We remark that the MoGG model showed more resilience than the MoG's for segmenting the shadow of the moving objects to the foreground. However, we noticed on other test sequences that, when the shadow is very pronounced, such as to create a separate mode in the histogram with high a priori probability, even the MoGG model failed to segment the shadow pixels to the background.

5 Conclusion

In this paper, we proposed a new mixture model based on the formalism of the general Gaussian distributions. We derived the maximum Likelihood Estimation and the MML criterion for the new model. Our experiments showed that the new model outperforms the MoG model in avoiding mixture over-fitting of noisy data. Tests performed on image, motion and foreground segmentation permitted to show the performance of the model. In future work, other applications of the proposed model will be investigated for problems based on mixture modeling.

References

[1] M.S. Allili, N. Bouguila and D. Ziou. Finite Generalized Gaussian Mixture Modeling and Applications to Segmentation and Tracking. *Technical Report*, September 2006.

[2] M. Baccar, L.A. Gee and M.A. Abidi. Reliable Location and Regression Estimates with Application to Range Image Segmentation. *J. of Math. Imag.*, 11:195-205, 1999.

[3] R.A. Baxter and J.J. Olivier. Finding Overlapping Components with MML. *Stat. and Computing*, 10(1):516, 2000.

[4] C.M. Bishop. Pattern Recognition and Machine Learning. *Springer*, 2006.

[5] N. Bouguila and D. Ziou. Online Clustering via Finite Mixtures of Dirichlet and Minimum Message Length. *Engineering Applications of Artificial Intelligence*, 19(4):371-379, 2006.

[6] N. Bouguila and D. Ziou. Unsupervised Selection of a Finite Dirichlet Mixture Model: An MML-Based Approach. *IEEE Trans. on KDE*, 18(8):993-1009, 2006.

[7] G.E.P. Box and P.W. Tidwell. Transformation of Independent Variables. *Technometrics*, 4(4):531-550, 1962.

[8] J. Cheng, J. yang, Y. Zhou and Y. Cui. Flexible Background Mixture Models for Foreground Segmentation. *Image and Vision Computing*, 24(5):473-482, 2006.

[9] M.N. Do and M. Vetterli. Wavelet-Based Texture Retrieval Using Generalized Gaussian Density

and Kullback-Leibler Distance. *IEEE Trans. on IP*, 11(2):146-158, 2002.

[10] K-T. Fang, S. Kotz and K-W. Ng. Symmetric Multivariate and Related Distributions. *Chapman and Hall*, 1990.

[11] K. Kokkinakis and A.K. Nandi. Exponent Parameter Estimation for Generalized Gaussian Probability Density Functions with Application to Speech Modelling. *Signal Processing*, 85(9):1852-1858, 2005.

[12] B.K.P. Horn and B. Schunck. Determining Optical Flow. Artificial Intelligence, 17:185-2003, 1981.

[13] R.L. Joshi and T.R. Fischer. Comparison of Generalized Gaussian and Laplacian Modelling in DCT Image Coding. *IEEE SPL*, 2(5):81-82, 1995.

[14] M. H. C. Law, M.A.T. Frigueiredo and A.K. Jain. Simultanuous Feature Selection and Clustering Using Mixture Models. *IEEE Trans. on PAMI*, 26(9):1154-1166, 2004.

[15] J.K. Lindsey. Multivariate Elliptically Contoured Distributions for Repeated Measurements. *Biometrics*, 55:1277-1280, 1999.

[16] G. McLachlan and D. Peel. Finite Mixture Models. *Wiley Series in Probability and Statistics*, 2000.

[17] J. Puzicha and T. Hofmann and J.M. Buhmann. Discrete Mixture Models for Unsupervised Image Segmentation. *DAGM-Symposium*, 135-142, 1998.

[18] A.E. Raftery and J.D. Banfield. Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics*, 49:803-821, 1993.

[19] K. Sharifi and A. Leon-Garcia. Estimation of Shape Parameter for Generalized Gaussian Distibution in Subband Decomposition of Video. *IEEE Trans. on CSVT*, 5(1):52-56, 1995.

[20] C. Stauffer and W.E.L. Grimson. Learning Patterns of Activity Using Real-Time Tracking. *IEEE Trans. on PAMI*, 22(8):747-757, 2000.

[21] D.M. Titterington. Recursive Parameter Estimation Using Incomplete Data. *Journal of Royal Statistical Society*, 46(2):257-267, 1984.

[22] X. Yang and S.M. Krishman. Image Segmentation Using Finite Mixtures and Spatial Information. *Image and Vision Computing*, 22(9):735-745, 2004.

[23] J-F. Yao. On Recursive Estimation in Incomplete Data Models. *Statistics*, 34:27-51, 2000.

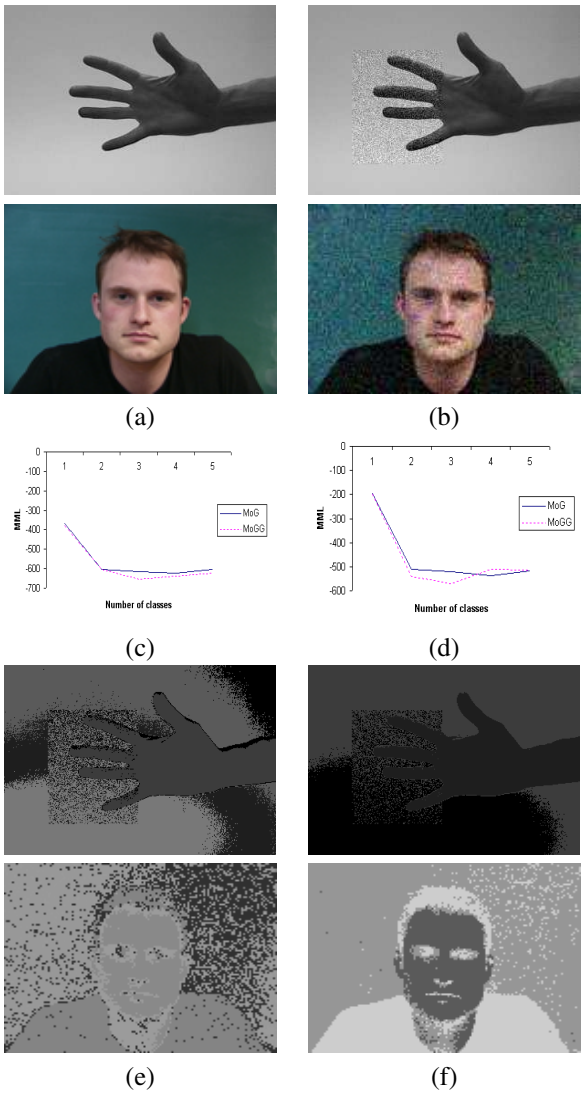

Figure 4. Example of real world image segmentation: In the first and second rows, (a) and (b) show respectively the original image and noisy version of it. In the third row, (c) shows the value of the MML calculated for different number of regions in the noisy image of the first row by using the MoG and MoGG models, (d) shows the same MML calculation for the noisy image of the second row. The two last rows show the segmentation of the noisy images by using: (e) the MoG model, (f) the MoGG model.

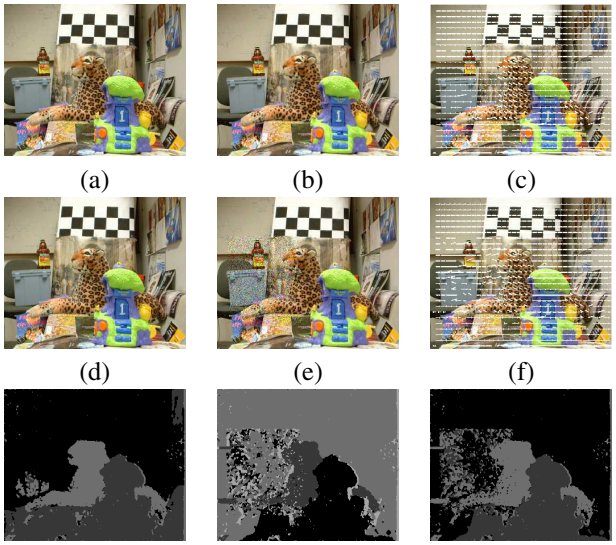

Figure 5. Motion segmentation by using mixture of pdfs. Fig. (5.a) and (5.d) represent the image a time t. Fig. (5.b) and (5.e) represent resp. the image at time $t+1$ and a noised version of it. Fig. (5.c) and (5.f) represent resp. the motion fields calculated for the original and noised images. Fig. (5.g) represents the segmentation of the motion field in fig. (5.c) by using a MoG. Fig. (5.h) and (i) represent the motion segmentation in fig. (5.f) by using resp. the MoG and MoGG models.

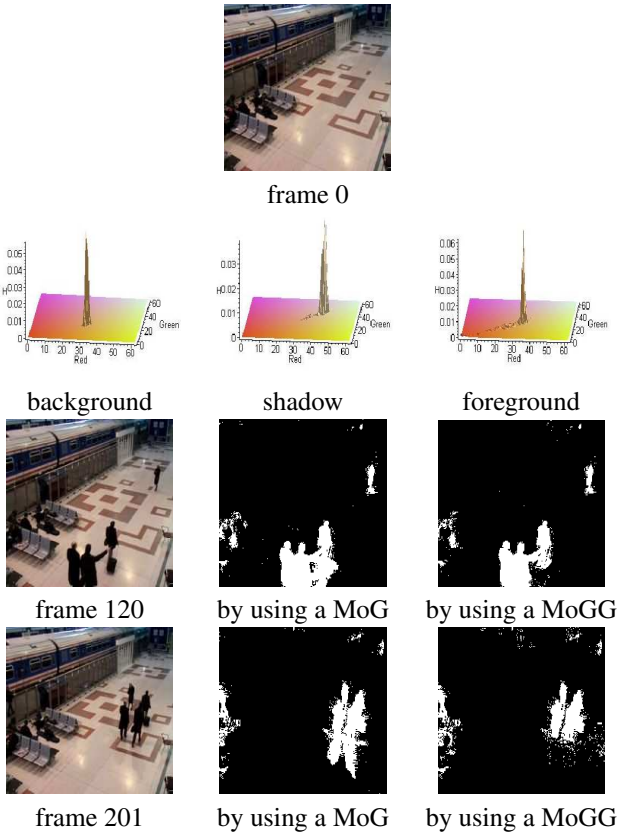

Figure 6. Example of foreground segmentation in a video sequence containing shadows. The first line shows the different profiles of pixel histograms in the sequence. The second and third rows show the result of foreground segmentation for two frames containing shadows by using the MoG and MoGG models.

Fourth Canadian Conference on Computer and Robot Vision(CRV'07)
0-7695-2786-8/07 $20.00 © 2007 IEEE