Computer Vision and Image Understanding 117 (2013) 1659–1671

Contents lists available at SciVerse ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Finite asymmetric generalized Gaussian mixture models learning for infrared object detection

Tarek Elguebaly [a], Nizar Bouguila [b,*]

[a] Electrical and Computer Engineering (ECE), Concordia University, Montreal, QC, Canada
[b] Concordia Institute for Information Systems Engineering (CIISE), Concordia University, Montreal, QC, Canada

ARTICLE INFO

Article history:
Received 18 July 2012
Accepted 11 July 2013
Available online 25 July 2013

Keywords:
Infrared
Multiple target tracking
Pedestrian detection
Foreground segmentation
Image fusion
Mixture models
Asymmetric generalized Gaussian
EM
MML

ABSTRACT

The interest in automatic surveillance and monitoring systems has been growing over the last years due to increasing demands for security and law enforcement applications. Although, automatic surveillance systems have reached a significant level of maturity with some practical success, it still remains a challenging problem due to large variation in illumination conditions. Recognition based only on the visual spectrum remains limited in uncontrolled operating environments such as outdoor situations and low illumination conditions. In the last years, as a result of the development of low-cost infrared cameras, night vision systems have gained more and more interest, making infrared (IR) imagery as a viable alternative to visible imaging in the search for a robust and practical identification system. Recently, some researchers have proposed the fusion of data recorded by an IR sensor and a visible camera in order to produce information otherwise not obtainable by viewing the sensor outputs separately. In this article, we propose the application of finite mixtures of multidimensional asymmetric generalized Gaussian distributions for different challenging tasks involving IR images. The advantage of the considered model is that it has the required flexibility to fit different shapes of observed non-Gaussian and asymmetric data. In particular, we present a highly efficient expectation–maximization (EM) algorithm, based on minimum message length (MML) formulation, for the unsupervised learning of the proposed model's parameters. In addition, we study its performance in two interesting applications namely pedestrian detection and multiple target tracking. Furthermore, we examine whether fusion of visual and thermal images can increase the overall performance of surveillance systems.

1. Introduction

Security of human lives and property has always been a major concern. Nowadays, developing video surveillance systems aimed at monitoring private and public areas has became one of the most active research fields due to the high amount of theft, accidents, terrorists attacks and riots. However, human attention is known to drop after just 30 min when engaged in monotonous and repetitive activities [1]. This is the case for security personnel tasked to monitor relatively vast environments where suspicious events are rare. Therefore, automatic video surveillance techniques were proposed to allow automatic processing of the data acquired by surveillance cameras without requiring the continuous attention of human operators. Automatic video surveillance systems are employed in controlled and uncontrolled environments [2]. In controlled or indoor environments (i.e. airports, warehouses, and production plants) monitoring is easier to implement as it does not depend on weather changes [3,4]. Uncontrolled environment is used to refer to outdoor scenes where illumination and temperature changes occur frequently, and where various atmospheric conditions can be observed [4,5].

Normally, when setting up a security system there are two major types of security cameras: visual-light, and infrared sensors. Visual-light or color cameras are employed vastly due to their lower cost compared to infrared sensors [6,7]. However, under low illumination sensing in visible spectrum becomes infeasible [8]. Thermal IR sensors measure the emitted heat energy from different objects, which make it invariant to changes in ambient illumination. Hence, IR imaging is a perfect choice for monitoring under low illumination conditions or even in darkness [9]. In order to show that thermal IR offers a promising alternative to visible imagery we will use it for pedestrian detection. Despite its robustness to illumination changes, IR has various drawbacks. One of its disadvantages is its sensitivity to outdoor temperature changes, which make it vulnerable to cold or warm air [10,11]. Some researchers decided to use both visible and infrared images together in order to increase the efficiency of surveillance systems [12,13]. It is widely known in the field of image fusion that the combination

* Corresponding author.
E-mail addresses: t_elgue@encs.concordia.ca (T. Elguebaly), nizar.bouguila@concordia.ca (N. Bouguila).

1077-3142/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2013.07.007

of thermal infrared and visible images is not trivial. Fusion techniques can be grouped into two classes: representative and analytical. Representative fusion uses both visible and infrared features together in order to generate a new image more informative or intuitive for a human observer. It is important to understand that the generation of such an image can be of a great importance in the case of human monitoring and is not required for automated video monitoring applications. On the other hand, analytical fusion combines available information from both sensors for a more robust analysis and interpretation of the image or video content. This method is based on the idea that combining both thermal and visible information can overcome the disadvantages of both visible-light images (i.e. shadows problem, sensitivity to variations in illumination and lights) and infrared images (i.e. sensitivity to outdoor temperature changes).

Discovering and finding valuable information and patterns in multidimensional data depends generally on the selection of an appropriate statistical model and the learning of its parameters. In recent years a lot of different algorithms were developed in the aim of automatically learning to recognize complex patterns, and to produce intelligent decisions based on observed data. Finite mixture models are now among the most widely used statistical approaches in many areas and applications and allow a formal approach for unsupervised learning. In such context, classic interest is often related to the determination of the number of clusters (i.e. model selection) and the estimation of the mixture's parameters. The isotropic nature of the Gaussian distribution, along with its capability to represent the data compactly by a mean vector and covariance matrix, has made Gaussian mixture (GM) decomposition a popular technique. However, Gaussian density has some drawbacks such as its symmetry around the mean and the rigidity of its shape, which prevent it from fitting accurately the data especially in the presence of outliers. Fig. 1 shows an example of an IR image. We can notice that its intensity distribution is not symmetrical. It is clear that using the GM to represent this distribution is not efficient. In order to overcome problems related to the Gaussian assumption, some researchers have shown that the generalized Gaussian distribution (GGD) can be a good choice to model non-Gaussian data [14,15]. Compared to the GD, the 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. In this article, we suggest the consideration of the asymmetric generalized Gaussian distribution (AGGD) capable of modeling non-Gaussian asymmetrical data. The AGGD uses two variance parameters for left and right parts of the distribution, which allow it not only to approximate a large class of statistical distributions (e.g. impulsive, Laplacian, Gaussian and uniform distributions) but also to include the asymmetry. As shown in Fig. 1(b) we can notice that the asymmetric generalized Gaussian mixture (AGGM) was able to accurately model the data and outperforms both the GM and the generalized Gaussian mixture (GGM).

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 maximizes a model selection criterion such as Akaike's information criterion (AIC) [16], minimum description length (MDL) [17] and Laplace empirical criterion (LEC) [18]. Stochastic methods such as Markov chain Monte Carlo (MCMC) can be used in order to sample from the full *a posteriori* distribution with M considered unknown [19]. Despite their formal appeal, MCMC methods are too computationally demanding, therefore cannot be applied efficiently for online applications such as automatic video surveillance. For this reason, we are interested in deterministic approaches. In our proposed method, we use K-means algorithm to initialize the asymmetric generalized Gaussian mixture parameters and successfully solve the initialization problem. The number of mixture components is automatically determined by implementing MML criterion [20] into an EM algorithm based on maximum likelihood (ML) estimation. Our learning method can integrate simultaneously parameter estimation and model selection in a single algorithm and is consequently totally unsupervised. It is noteworthy that the proposed work is completely different from recent efforts published for instance in [21–23]. In fact, [21] proposed the use of the gradient and the ML methods for estimating the parameters of only a one-dimensional AGGD. The work in [22] has been devoted to image segmentation using ML estimation of one-dimensional AGGM with known number of components. In [23] a Bayesian nonparametric 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 AGGM model and gives a complete learning algorithm. In Section 3, we assess the performance of the new model for pe-

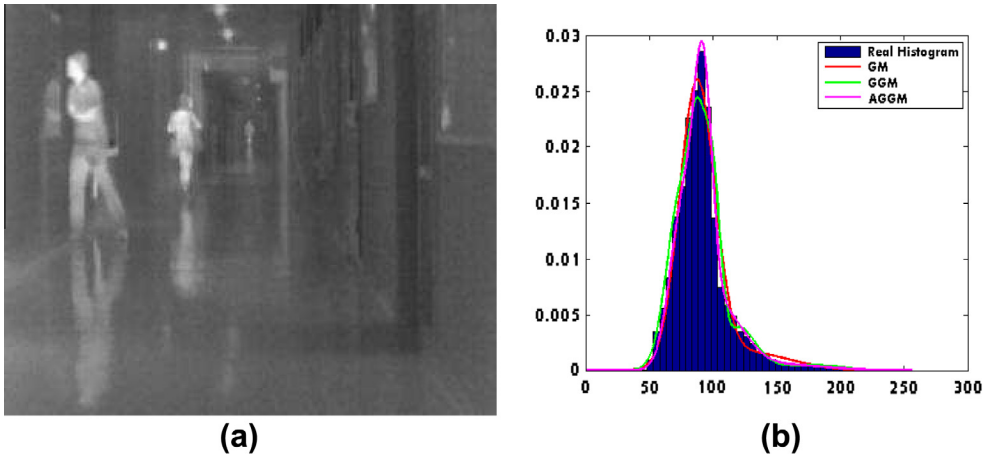

Fig. 1. (a) IR image. (b) Real and estimated (using GM, GGM and AGGM) histograms for the IR image.

destrian detection and multiple-target tracking; while comparing it to other models. Our last section is devoted to the conclusion and some perspectives.

2. Finite asymmetric generalized Gaussian mixture model

2.1. The finite mixture model

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

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

where ξ_j is the set of parameters of 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, $M \geqslant 1$ is number of components in the mixture. For the AGGM, each component density $p(\vec{X}|\xi_j)$ is an AGGD:

$$p(\vec{X}|\xi_j) = \prod_{k=1}^{d} \begin{cases} \frac{\beta_{jk}\left[\frac{\Gamma(3/\beta_{jk})}{\Gamma(1/\beta_{jk})}\right]^{1/2}}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})\Gamma(1/\beta_{jk})} \exp\left[-A(\beta_{jk})\left(\frac{\mu_{jk}-X_k}{\sigma_{l_{jk}}}\right)^{\beta_{jk}}\right] & \text{if } X_k < \mu_{jk} \\ \frac{\beta_{jk}\left[\frac{\Gamma(3/\beta_{jk})}{\Gamma(1/\beta_{jk})}\right]^{1/2}}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})\Gamma(1/\beta_{jk})} \exp\left[-A(\beta_{jk})\left(\frac{X_k-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}}\right] & \text{if } X_k \geqslant \mu_{jk} \end{cases} \tag{2}$$

where $A(\beta_{jk}) = \left[\frac{\Gamma(3/\beta_{jk})}{\Gamma(1/\beta_{jk})}\right]^{\beta_{jk}/2}$ and $\Gamma(.)$ is the Gamma function given by: $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt,\ x > 0$. Note that $\xi_j = (\vec{\mu}_j, \vec{\beta}_j, \vec{\sigma}_{l_j}, \vec{\sigma}_{r_j})$ is the set of parameters of component j where $\vec{\mu}_j = (\mu_{j1}, \ldots, \mu_{jd})$, $\vec{\sigma}_{lj} = (\sigma_{l_{j1}}, \ldots, \sigma_{l_{jd}})$, and $\vec{\sigma}_{rj} = (\sigma_{r_{j1}}, \ldots, \sigma_{r_{jd}})$ are the mean, the left standard deviation, and the right standard deviation of the d-dimensional AGGD, respectively. The parameter $\vec{\beta}_j = (\beta_{j1}, \ldots, \beta_{jd})$ controls the tails of the pdf and determines whether it is peaked or flat: the larger the value of $\vec{\beta}_j$, the flatter the pdf, and the smaller $\vec{\beta}_j$ is, the more peaked the pdf. The AGGD 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 $\mathcal{X} = (\vec{X}_1, \ldots, \vec{X}_N)$ be a set of N independent and identically distributed vectors, assumed to arise from a finite AGGM with M components. Thus, its corresponding likelihood can be expressed as follows:

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

where the set of parameters of the mixture with M classes is defined by $\Theta = (\vec{\mu}_1, \ldots, \vec{\mu}_M, \vec{\beta}_1, \ldots, \vec{\beta}_M, \vec{\sigma}_{l_1}, \ldots, \vec{\sigma}_{l_M}, \vec{\sigma}_{r_1}, \ldots, \vec{\sigma}_{r_M}, p_1, \ldots, p_M)$. We introduce membership vectors, $\vec{Z}_i = (Z_{i1}, \ldots, Z_{iM})$, one for each observation encoding to which component the observation belongs. In other words, Z_{ij} $j = 1, \ldots, M$ equals 1 if $\vec{X}_i$ belongs to class j and 0, otherwise. Taking into account $Z = \{\vec{Z}_1, \ldots, \vec{Z}_N\}$, the complete-data likelihood is given by:

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

2.2. Maximum likelihood estimation of the mixture parameters

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

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

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:

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

Using Eq. (6) we can assign each vector $\vec{X}_i$ to one of the M clusters. Now, using these expectations, the goal is to maximize the complete data log-likelihood with respect to our model parameters. This can be done by calculating the gradient of the log-likelihood with respect to $p_j, \vec{\mu}_j, \vec{\beta}_j, \vec{\sigma}_{lj}$, and $\vec{\sigma}_{rj}$. 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:

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

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

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

By calculating the gradients of the complete log-likelihood with respect to $\vec{\mu}_j, \vec{\beta}_j, \vec{\sigma}_{l_j}$, and $\vec{\sigma}_{r_j}$, we obtain the following for $k = 1, \ldots, d$:

$$\sum_{i=1, X_{ik}<\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})}{\sigma_{l_{jk}}^{\beta_{jk}}}(\mu_{jk} - X_{ik})^{\beta_{jk}-1} - \sum_{i=1, X_{ik}\geqslant\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})}{\sigma_{r_{jk}}^{\beta_{jk}}}(X_{ik} - \mu_{jk})^{\beta_{jk}-1} = 0 \tag{9}$$

$$\begin{aligned} &\sum_{i=1, X_{ik}<\mu_{jk}}^{N} Z_{ij}A(\beta_{jk})\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta jk}\left[\left(\frac{3\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{2\beta_{jk}}\right) - \log\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)\right] \\ &+ \sum_{i=1, X_{ik}\geqslant\mu_{jk}}^{N} Z_{ij}A(\beta_{jk})\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta jk}\left[\left(\frac{3\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{2\beta_{jk}}\right) - \log\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)\right] \\ &+ \sum_{i=1}^{N} Z_{ij}\left[\frac{1}{\beta_{jk}} - \frac{3}{2}\left(\frac{\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{\beta_{jk}^2}\right)\right] = 0 \end{aligned} \tag{10}$$

$$\sum_{i=1, X_{ik}<\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})\beta_{jk}}{\sigma_{l_{jk}}}\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta jk} - \sum_{i=1}^{N}\frac{Z_{ij}}{\sigma_{l_{jk}}+\sigma_{r_{jk}}} = 0 \tag{11}$$

$$\sum_{i=1, X_{ik}\geqslant\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})\beta_{jk}}{\sigma_{r_{jk}}}\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta jk} - \sum_{i=1}^{N}\frac{Z_{ij}}{\sigma_{l_{jk}}+\sigma_{r_{jk}}} = 0 \tag{12}$$

where $\Psi(x) = \frac{\partial \log[\Gamma(x)]}{\partial x}$. It is easy to notice that the equations from (9)–(12) related to all AGGD parameters are non linear. Thus, we decided to use the Newton–Raphson method to estimate these parameters:

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

$$\hat{\beta}_{jk} \simeq \beta_{jk} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \beta_{jk}^2}\right)^{-1}\left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \beta_{jk}}\right)\right] \tag{14}$$

$$\hat{\sigma}_{l_{jk}} \simeq \sigma_{l_{jk}} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk}}^2}\right)^{-1}\left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk}}}\right)\right] \tag{15}$$

$$\hat{\sigma}_{r_{jk}} \simeq \sigma_{r_{jk}} - \left[\left(\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma_{r_{jk}}^2}\right)^{-1}\left(\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \sigma_{r_{jk}}}\right)\right] \quad (16)$$

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

2.3. Model selection using MML criterion

Different model selection methods have been introduced to estimate the number of components of a mixture model. Among these methods the MML criterion has been shown to perform efficiently. The MML approach is based on evaluating statistical models according to their ability to compress a message containing the data (minimum coding length criterion). 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 its parameters 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 (i.e. the message length) [20,24]:

$$MessLen \approx -\log(p(\Theta)) - L(\Theta,Z,\mathcal{X}) + \frac{1}{2}\log|F(\Theta)| + \frac{N_p}{2} - \frac{1}{2} \times \log(12) \quad (17)$$

where $p(\Theta)$ is the prior probability, $-F(\Theta)-$ is the determinant of the Fisher information matrix of minus the log-likelihood of the mixture, and N_p is the number of parameters to be estimated and is equal to $M(4d+1)$ in our case. In the following sections, we develop both $p(\Theta)$ and $-F(\Theta)-$.

2.3.1. Derivation of $p(\Theta)$

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 our parameters $(\mu = \{\vec{\mu}_j\}, \beta = \{\vec{\beta}_j\}, \sigma_l = \{\vec{\sigma}_{lj}\}, \sigma_r = \{\vec{\sigma}_{rj}\}, P = (p_1,\ldots,p_M))$ are mutually independent, then:

$$p(\Theta) = p(\mu)p(\beta)p(\sigma_l)p(\sigma_r)p(P) \quad (18)$$

In what follows, we will compute each of these priors separately. Starting with $p(P)$, we know that P 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

$$p(P) = \frac{\Gamma\left(\sum_{j=1}^{M}\eta_j\right)}{\prod_{j=1}^{M}\Gamma(\eta_j)}\prod_{j=1}^{M} p_j^{\eta_j - 1} \quad (19)$$

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

$$p(P) = (M-1)! \quad (20)$$

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

$$p(\mu) = \prod_{j=1}^{M}\prod_{k=1}^{d} p(\mu_{jk}) = \prod_{k=1}^{d}\frac{1}{(\sigma_{l_k} + \sigma_{r_k})^M} \quad (21)$$

For the parameter β, we adopt a uniform distribution $\mathcal{U}[0,h]$ for each β_{jk}, where h is the maximum value permitted. Then the prior for β is given by

$$p(\beta) = \prod_{j=1}^{M}\prod_{k=1}^{d} p(\beta_{jk}) = \frac{1}{h^{Md}} \quad (22)$$

It is known that $(0 \leqslant \sigma_{l_{jk}} \leqslant \sigma_{l_k})$ and $(0 \leqslant \sigma_{r_{jk}} \leqslant \sigma_{r_k})$ for σ_l and σ_r, respectively. Then, for both parameters σ_l and σ_r we take a uniform prior for each $\sigma_{l_{jk}}$ and $\sigma_{r_{jk}}$

$$p(\sigma_l) = \prod_{j=1}^{M}\prod_{k=1}^{d} p(\sigma_{l_{jk}}) = \prod_{k=1}^{d}\frac{1}{\sigma_{l_k}^M} \quad (23)$$

$$p(\sigma_r) = \prod_{j=1}^{M}\prod_{k=1}^{d} p(\sigma_{r_{jk}}) = \prod_{k=1}^{d}\frac{1}{\sigma_{r_k}^M} \quad (24)$$

Finally, by replacing the priors in Eq. (18) by the expressions in Eqs. (20)–(24), we get

$$p(\Theta) = \frac{(M-1)!}{h^{Md}}\prod_{k=1}^{d}\frac{1}{\sigma_{l_k}^M \sigma_{r_k}^M (\sigma_{l_k} + \sigma_{r_k})^M} \quad (25)$$

2.3.2. Derivation of $|F(\Theta)|$

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 analytically the expected Fisher information matrix of a mixture. Therefore, we use the complete Fisher information matrix which determinant is equal to the product of the determinants of the information matrices with respect to the parameters of each mixture component:

$$|F(\Theta)| = |F(P)|\prod_{j=1}^{M}|F(\vec{\mu}_j)||F(\vec{\beta}_j)||F(\vec{\sigma}_{l_j})||F(\vec{\sigma}_{r_j})| \quad (26)$$

where $F(P)|, |F(\vec{\mu}_j)|, |F(\vec{\beta}_j)|, |F(\vec{\sigma}_{l_j})|$, and $|F(\vec{\sigma}_{r_j})|$ are the Fisher information with regards to P, $\vec{\mu}_j$, $\vec{\beta}_j$, $\vec{\sigma}_{l_j}$, and $\vec{\sigma}_{r_j}$, respectively. Regarding $-F(P)-$ it is straightforward to show that:

$$|F(P)| = \frac{N^{M-1}}{\prod_{j=1}^{M} p_j} \quad (27)$$

The Hessian matrices when we consider the vectors $\vec{\mu}_j, \vec{\beta}_j, \vec{\sigma}_{l_j}$, and $\vec{\sigma}_{r_j}$ are given by

$$F(\vec{\mu}_j)_{k_1,k_2} = \frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \mu_{jk_1}\partial \mu_{jk_2}} \quad (28)$$

$$F(\vec{\beta}_j)_{k_1,k_2} = \frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \beta_{jk_1}\partial \beta_{jk_2}} \quad (29)$$

$$F(\vec{\sigma}_{l_j})_{k_1,k_2} = \frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma_{l_{jk_1}}\partial \sigma_{l_{jk_2}}} \quad (30)$$

$$F(\vec{\sigma}_{r_j})_{k_1,k_2} = \frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma_{r_{jk_1}}\partial \sigma_{r_{jk_2}}} \quad (31)$$

where $(k_1,k_2) \in (1,\ldots,d)$. Using A to compute the derivatives in Eqs. (28)–(31), we obtain

$$|F(\vec{\mu}_j)| = \prod_{k=1}^{d} -A(\beta_{jk})\beta_{jk}(\beta_{jk}-1) \left[\sum_{i=1, X_{ik}<\mu_{jk}}^{N} Z_{ij}\frac{(\mu_{jk}-X_{ik})^{\beta_{jk}-2}}{\sigma_{l_{jk}}^{\beta_{jk}}} + \sum_{i=1, X_{ik}\geqslant\mu_{jk}}^{N} Z_{ij}\frac{(X_{ik}-\mu_{jk})^{\beta_{jk}-2}}{\sigma_{r_{jk}}^{\beta_{jk}}}\right] \quad j=1,\ldots,M \quad (32)$$

$$|F(\vec{\beta}_j)| = \prod_{k=1}^{d} - \left\{ \sum_{i=1}^{N} Z_{ij} \left[\frac{1}{\beta_{jk}^2} + \frac{3\Psi'(1/\beta_{jk})}{2\beta_{jk}^4} + 3\frac{\Psi(1/\beta_{jk}) - \Psi(3/\beta_{jk})}{\beta_{jk}^3} - \frac{9\Psi'(3/\beta_{jk})}{2\beta_{jk}^4} \right] \right.$$
$$+ (\beta_{jk}) \sum_{i=1}^{N}_{X_{ik}<\mu_{jk}} Z_{ij} \left(\frac{\mu_{jk} - X_{ik}}{\sigma_{l_{jk}}} \right)^{\beta_{jk}} \left[\left(\frac{9\Psi'(3/\beta_{jk}) - \Psi'(1/\beta_{jk})}{2\beta_{jk}^3} + \frac{3\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{2\beta_{jk}^2} \right) \right.$$
$$\left. + \left(\frac{3\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{2\beta_{jk}} - \log \left[\frac{\mu_{jk} - X_{ik}}{\sigma_{l_{jk}}} \right] \right)^2 \right]$$
$$+ (\beta_{jk}) \sum_{i=1}^{N}_{X_{ik}\geq\mu_{jk}} Z_{ij} \left(\frac{X_{ik} - \mu_{jk}}{\sigma_{r_{jk}}} \right)^{\beta_{jk}} \left[\left(\frac{9\Psi'(3/\beta_{jk}) - \Psi'(1/\beta_{jk})}{2\beta_{jk}^3} + \frac{3\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{2\beta_{jk}^2} \right) \right.$$
$$\left. \left. + \left(\frac{3\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{2\beta_{jk}} - \log \left[\frac{X_{ik} - \mu_{jk}}{\sigma_{r_{jk}}} \right] \right)^2 \right] \right\} \quad j = 1, \ldots, M \tag{33}$$

where $\Psi\prime(x) = \frac{\partial^2 \log[\Gamma(x)]}{\partial x^2}$.

$$|F(\vec{\sigma}_{l_j})| = \prod_{k=1}^{d} \left[\sum_{i=1}^{N} \frac{Z_{ij}}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})^2} - A(\beta_{jk})\beta_{jk}(\beta_{jk}+1) \sum_{X_{ik}<\mu_{jk}} \frac{Z_{ij}}{\sigma_{l_{jk}}^2} \left(\frac{\mu_{jk} - X_{ik}}{\sigma_{l_{jk}}} \right)^{\beta_{jk}} \right] \quad j = 1, \ldots, M \tag{34}$$

$$|F(\vec{\sigma}_{r_j})| = \prod_{k=1}^{d} \left[\sum_{i=1}^{N} \frac{Z_{ij}}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})^2} - A(\beta_{jk})\beta_{jk}(\beta_{jk}+1) \sum_{X_{ik}\geq\mu_{jk}} \frac{Z_{ij}}{\sigma_{r_{jk}}^2} \left(\frac{X_{ik} - \mu_{jk}}{\sigma_{r_{jk}}} \right)^{\beta_{jk}} \right] \quad j = 1, \ldots, M \tag{35}$$

2.4. Complete AGGM learning algorithm

In the following steps, we summarize the algorithm used for the learning of our AGGM [1]:

Algorithm 1.

Input: Data set $\mathcal{X}$ and M_{max}
Output: Θ_{M^*} (the values of Θ when M^* components are chosen) and M^*
Step 1: For $M = 1$: M_{max} **do**{
1. Initialization.
2. Repeat until convergence.
   (a) The expectation step using Eq. (6).
   (b) The maximization step using Eqs. (13)–(16).
3. Calculate the associated message length using Eq. (17).

} **END FOR**

Step 2: Select the model M^* with the smallest message length value.

In order 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, as for the values of the shape parameters we initialized them to 2. It is noteworthy that this is equivalent actually to reducing the AGGM to a simple GM at the initialization step. Concerning the convergence, we stop the iterations when the log-likelihood does not change much from one step to the next. More interesting and detailed information on the convergence properties of the EM algorithm can be found in [25].

3. Experimental results

In this section we report results on two interesting applications namely pedestrian detection and multiple-target tracking. We investigate the effectiveness of our algorithm by comparing it to other state of the art methods. In all our experiments M_{max} is set to 9.

3.1. Pedestrian detection

Pedestrian detection is an essential task in intelligent automatic video surveillance systems. In recent years, numerous approaches have been proposed in order to detect, track, and recognize human activity [26]. However, pedestrian detection still remains an active research area in computer vision. The pedestrian detection task is challenging from a computer vision perspective due to the great variety of human appearances (very high intraclass variability), background structure, partial occlusions, and lighting conditions. In this paper, we present an approach toward pedestrian detection for infrared imagery. Unlike visible images, thermal images characteristically have a low SNR, halos that appear around very hot or cold objects, and are vulnerable to climate and temperature changes. For these reasons, we decide to use the AGGM as data contain non-Gaussian characteristics impossible to model using rigid distributions. The AGGM model is used in this application to partition a given IR image into regions (each associated with one mixture component). The number of mixture components needed for the image and the parameters of each component are determined using the learning algorithm developed in the previous section.

It is known that warm objects (objects with high thermal inertia like water, animals, and people) appear lighter than cold objects (dark surfaces like cars and buildings) in thermal imagery. Thus, pedestrian detection can be done by choosing the distribution with the largest mean, but this will not be efficient due to the polarity switch phenomenon [8]. Polarity switch is known as the phenomenon that reverses the hot and cold ranges of thermal sensor (pedestrians that normally give rise to bright pixels became dark pixels). We have noticed that when segmenting the image using mixture models the class that contains pedestrians is characterized by its large left and right standard deviations. Our algorithm is very simple and can be summarized as follows:

1. Apply the AGGM introduced in Section 2.4 on each IR image to partition it into regions.
2. Check if the class with the largest mean has the largest variance as compared to other classes.
   (a) If true, then this is the distribution that models the pedestrians in the image.
   (b) If false, then choose the distribution with the largest variance.

To show the effectiveness of our method we compared it with four other methods: the GM learned via the MML criterion, the infinite Gaussian mixture model (IGM) [27], the GGM with MML [14], and the infinite generalized Gaussian mixture model (IGGM) [23]. We have used the OSU Thermal Pedestrian Database [28] for this application. We decided to use this dataset as it is taken on different days and under different weather conditions, which makes it vulnerable to climate and temperature changes. Fig. 2 shows an image taken in the presence of Haze for five pedestrians. Comparing the four outputs together we can notice that the GM and the IGM both have wrongly modeled the two white street parts and the street lamp, also the pedestrian behind the tree was not correctly represented. As for the GGM, it has taken the street lamp into consideration, and failed to represent the pedestrian behind the tree. Both IGGM and our method were able to recognize the five pedestrians without any problem. Fig. 3 shows an image taken on a very cloudy day for four pedestrians. We can notice from this image that the effect of a cloudy day is the same as the polarity switch phenomenon. From the methods' outputs, we can see that the GM and the IGM both have only identified two pedestrians and their outputs are very noisy. For the GGM, it has identified three pedestrians out of the four. As for the last two, they were able to recognize all of them clearly. Fig. 4 shows an image taken on a rainy day for six pedestrians where three are using umbrellas. Comparing all outputs together we can notice that the IGGM and our method outperformed the three other methods. In order to

[1] The complete source code is available upon request.

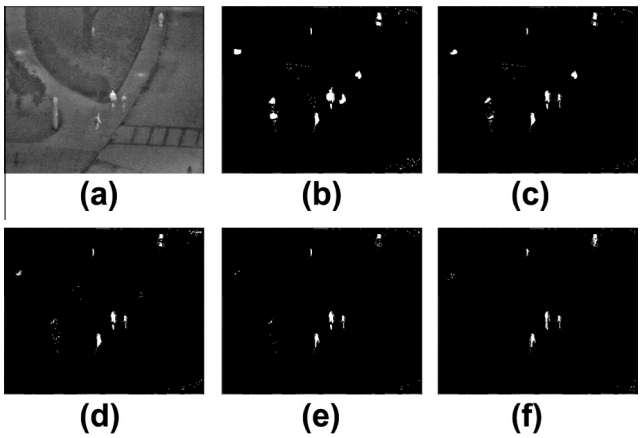

Fig. 2. (a) IR image for five pedestrians in the presence of Haze, (b) GM, (c) IGM, (d) GGM, (e) IGGM, and (f) AGGM.

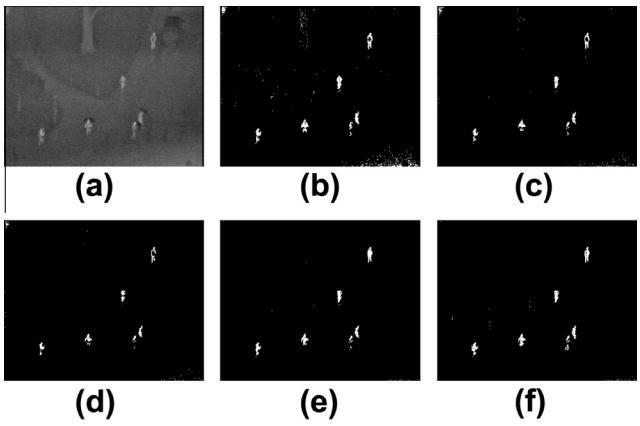

Fig. 4. (a) IR Image for six pedestrians on a rainy day, (b) MoG, (c) IMoG, (d) MoGG, (e) IMoGG, and (f) MoAGG.

have a quantitative evaluation of the performance, we have used two well-known metrics, precision and recall, to quantify how well each algorithm works in classifying the data [29]. Precision (Eq. (36)) represents the percentage of detected true positives (pedestrians) to the total number of items detected by the algorithm. Recall (Eq. (37)) is the percentage of number of detected true positives by the algorithm to the total number of true positives in the dataset:

$$Precision = \frac{TP}{TP + FP} \tag{36}$$

$$Recall = \frac{TP}{TP + FN} \tag{37}$$

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 pedestrians that were wrongly classified as background (false negatives). Table 1 represents the average recall and precision for our method (AGGM), the IGGM, the GGM, the IGM, the GM, as well as the methods introduced in [28,8]. From this table we can deduce that our model is capable of detecting pedestrians efficiently. According to quantitative and analytical analysis, it is clear that our method was able to identify all pedestrians in each image even under polarity switch phenomenon. From Table 1 we can see that our method outperformed all the other model based approaches except the IGGM, however, Bayesian methods are still far too computationally demanding since the learning is based on Markov Chain Monte Carlo (MCMC) techniques and cannot be applied efficiently for online applications like pedestrian detection. Furthermore, we can notice that the method introduced in [8] outperformed our method, however, this method was designed for pedestrian tracking and uses joint shape and appearance cues to resolve this problem which is not as simple as our method. Finally, when comparing our technique to the two stages approach introduced in [28] we found that our method has a higher recall which means that it is more effective in identifying pedestrians.

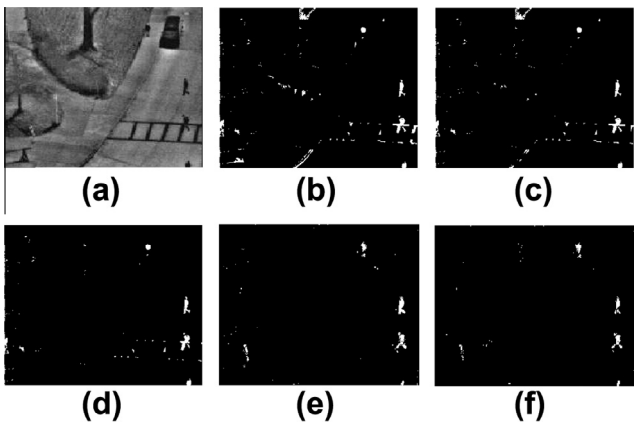

Fig. 3. (a) IR Image for six pedestrians on a very cloudy day, (b) GM, (c) IGM, (d) GGM, (e) IGGM, and (f) AGGM.

3.2. Multiple-target tracking

Multiple-target tracking (MTT) is a crucial task in video surveillance systems, as it aims at inferring trajectories for each target from a video sequence. MTT is challenging, especially when dealing with crowded scenes, due to similar appearance, inter and intra occlusions, low illumination conditions, outdoor temperature changes, and low resolution. Tracking can be achieved by bottom-up or top-down approaches. First mentioned approach, also known as low level tracker, consists of motion segmentation then a subsequent target association in order to detect each object size, position and velocity. Motion segmentation can be carried out either by optical flow, or background subtraction. Then, a prediction stage is applied using Kalman filter in order to provide better chances of tracking success. Top-down approach (high level tracker) is based on complex shape and motion modeling to deal with object appearance. Contour tracking has been widely used for tracking the boundary contour of a deforming object, however, it may be inappropriate in crowded scenes due to multiple target-occlusions. Comaniciu et al. [30] introduced another algorithm that performs a gradient-descent search on the region of interest in images but it was not effective for MTT. However, none of these two approaches alone is capable to deal simultaneously with the multiple target tracking problems such as environment occlusions, both total and partial, and collisions, such as grouping and splitting events. In this application, we extend the work presented in [31], where the authors introduced a new framework for MTT in visible spectrum that involves both low and high level approaches capable of overcoming the aforementioned problems.

Most solutions proposed for MTT have been proposed in the case of visible spectrum. In brightly illuminated scenes, standard color cameras provide the best information for object segmentation. However, in outdoor applications, darkness and other environmental conditions such as fog, rain and smoke strongly decrease the efficiency of standard cameras. In many applications, achievement of zero miss detection rate is a critical requirement and investment in more powerful imaging systems is justified. This

Table 1
Precision and recall.

	Davis and Keck [28]	Dai et al. [8]	GM	IGM	GGM	IGGM	AGGM
Prec. (%)	99.36	99.39	82.24	85.36	95.72	98.69	97.81
Rec. (%)	94.51	99.49	81.46	83.67	93.26	97.41	95.03

opens the way to video systems combining thermal and color cameras. This work is based on the hypothesis that the addition of LWIR cameras (8–12 μm) can significantly improve the robustness of MTT systems in uncontrolled environments. The used dataset for this application is the OSU Color-Thermal Database [32]. This dataset consists of two video sequences, one in the visible spectrum and the other in thermal infrared. Some sample frames taken from this data set are shown in Fig. 5.

Our method can be divided into three main components: Detection, Low level tracking, and high level tracking. Detection is used to detect every moving blob within the scene. Then, a low level tracker is introduced to track every isolated object. Finally, a high level tracker is applied to deal with occlusions. In this work, we are using the AGGM for blob detection as well as extending both the low and high level tracker represented in [31] to deal with fusion of both visible and infrared inputs.

3.2.1. Blob detection

The first step in any MTT system is the detection of every moving blob within the scene. This step, known as foreground segmentation, is often used as the primary step in video surveillance and optical motion capture in order to model the background and to detect the moving objects in the scene. Recently, adaptive GM models have been applied for segmenting video foregrounds [33–35] (a complete detailed survey can be found in [36]). The 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 [37,38]. However, these methods have some drawbacks. Modeling the background using the GM implies the assumption that the background and foreground distributions are Gaussians which is not always the case for uncontrolled environments as argued by [39].

Here, we try to overcome these problems related to GM by using AGGM to enhance the robustness of mixture modeling. Our approach is built on the method of [38] in which an online learning of a GM model for each pixel in the video frames is presented. This algorithm is based on the idea that the components that occur frequently in the mixture (i.e., with high prior probability and small variance) are used to model the background. Thus, in our case, every pixel at a given time frame t is represented by a vector of four values: $\vec{X}^{(t)} = [R, G, B, I]$, where R, G, B are the Red, Green, and Blue values, respectively, taken from the color camera. I is the intensity value taken from the thermal sensor. In order to segment the foreground, the components are first ordered by the value of $\frac{p_j}{\|\vec{\sigma}_{l_j}\|+\|\vec{\sigma}_{r_j}\|}$, where p_j is the mixing proportion for cluster j, $\|\vec{\sigma}_{l_j}\|$ and $\|\vec{\sigma}_{r_j}\|$ are the norm of the left and right standard deviations of the j component, respectively. Then, the first A components are chosen to model the background, such that

$$A = \arg\min_a \sum_{j=1}^{a} p_j > T, \tag{38}$$

where T is a measure of the minimum portion of the data that represents the background. In this application, we set $T = 0.3$ as the background model does not include any repetitive background motion. In order to update the model parameter, assume that a new value $\vec{X}^{(t+1)}$ is introduced while the parameters of the model $\mathcal{M}^t$ are known then

$$p_j^{(t+1)} = p_j^{(t)} + B_{(t)}[p(j|\vec{X}^{(t+1)}) - p_j^{(t)}] \tag{39}$$

$$\xi_j^{(t+1)} = \xi_j^{(t)} + B_{(t)}\left[p(j|\vec{X}^{(t+1)})\frac{\partial L(\Theta, Z, \mathcal{X}^{(t+1)})}{\partial \xi_j}\right] \tag{40}$$

where $B_{(t)}$ represents any sequence of positive numbers that decreases to zero. The derivatives in Eq. (40) are given in A. The MML criterion is used for the selection of the number of classes in the mixture model. For each pixel in the input frame of the sequence, we check whether its new value matches one of the components of its AGGM mixture, if true then this value is assigned to this component else a new component with the mean equal to the new value of the pixel is created for the mixture. Note that, a match is identified when the value of the pixel falls within two 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). For each iteration, we update the number of components of the mixture depending on the MML. The complete algorithm for foreground segmentation can be summarized in the following steps:

1. Mixture initialization for each pixel X.
   (a) Set $M = 1$, $p_1 = 1$.
   (b) $\forall\ k = 1, \ldots, d$: set $\sigma_{l_{1k}} = \sigma_{r_{1k}} = 0.2$, $\mu_{1k} = X_k^{(0)}$, and $\beta_{jk} = 2$.

Fig. 5. Some sample frames from the OSU Color-Thermal Database.

2. For each pixel $\vec{X}^{(t+1)}$ in a new frame do
   (a) verify if there is a match that exists for the new pixel value.
      i. if true assign this pixel to this component
      ii. else create a new component with the mean equal to the new value of the pixel.
   (b) Update the new pixel model parameters using Eqs. (39) and (40).
   (c) Order the pixel mixture components by the values of $\frac{p_j}{\|\vec{\sigma}_{l_j}\|+\|\vec{\sigma}_{r_j}\|}$.
   (d) Extract the foreground objects by identifying the first A components that represent the background using Eq. (38) .

In order to evaluate our algorithm for foreground segmentation we compared it with the well-known GM method introduced in [37], the GGM [14], and the IGGM introduced in [23]. Figs. 6–8 show some results of applying the four approaches on the OSU Color-Thermal Database. In order to demonstrate the robustness of our method we have used the background subtraction evaluation approach for a dataset without ground truth introduced in [40]:

$$D = cD_{color} + hD_{hist} + mD_{motion} \tag{41}$$

where the parameters c, h, and m can be adjusted according to the characteristics of the video sequence and are restricted to be one. In this application, we consider the straight arithmetic averaging of the three measures $c = h = m = 1/3$. D_{color} is used to measure the color difference between the color of pixels across the estimated object boundary. D_{hist} is used to assess the changes in the color histogram of the segmented object by calculating the pairwise color histogram differences of the video object planes (VOP) at time t and $t-1$. In order to quantify how well the estimated object boundaries coincide with actual motion boundaries, we use D_{motion}. Note that, D value is between zero and one where zero is the best value and one is the worst (see Table 2). From both quantitative and analytical analysis, we can find that our method performed as good as the IGGM. However, the AGGM is naturally preferred given the huge difference concerning computational time. We can notice from Figs. 6–8 the existence of noise. In order to remove noise in the background subtraction, we perform the following procedure: We remove isolated pixels defined as every pixel which is part of a 1-pixel thick object, then apply an opening morphological operator to fill the blobs, finally, we apply a Minimum-area filter.

In order to evaluate the efficiency of our algorithm for change detection we used the change detection dataset introduced in CVPR2012 [41]. We applied our method on five thermal video sequences from this dataset to assess its performance. Fig. 9(a) shows some samples taken from this dataset. In order to have a quantitative evaluation of the performance, we have used three "pixel wise" metrics, precision, recall, and F-measure [29]. Furthermore, we have chosen to compare our method with three state of the art methods: Zivkovic [33], KaewTraKulPong and Bowden [42], and Evangelio et al. [43]. From Table 3 we can deduce that our model is capable of detecting changes throughout the videos.

3.2.2. Low level tracker

After foreground segmentation and blob detection a low level tracker is applied to track every isolated object. Our idea in this step is to extract the contour of each blob and to compute the ellipse that represents it [31]. We have chosen this method as it is good in identifying each object and will be useful when dealing with object collision. Thus, the l-observed blob at time t is represented by $z_l^t = \left(x_l^t, y_l^t, h_l^t, w_l^t, \theta_l^t\right)$, where x_l^t, y_l^t represent the ellipse centroid, h_l^t and w_l^t are the major and minor axes, respectively, and θ_l^t is the ellipse orientation.

Knowing the components identifying each blob, the next step is to estimate the target state by filtering the sequence of noisy measures. The Kalman filter is widely used in tracking systems, it is an algorithm that uses a series of measurements observed over time, containing noise and other inaccuracies, and produces estimates that tend to be more precise than those that would be based on a single measurement alone. The Kalman filter uses a recursive algorithm in order to predict the position, in other words it works in two steps: predict the position, then use the observed measurements to correct the filter. We adopt a first order dynamics, used by [31] in order to predict the target state. Then, the target state for an ellipse representation is represented by $z_l^t = \left(x_l^t, \dot{x}_l^t, y_l^t, \dot{y}_l^t, h_l^t, \dot{h}_l^t, w_l^t, \dot{w}_l^t, \theta_l^t\right)$ where $\dot{v}_l^t$, $v \in \{x, y, h, w\}$, represents the velocity of each component. Note that, the velocity of θ_l^t is not measured as it is considered as noise. The measures used in this application are both the square root of the innovation covariance matrix S_k determinant and the Mahalanobis square distance (MSD). Fig. 10 shows an example of tracking using the ellipse centroid metric.

3.2.3. High level tracker

The Kalman filter introduced above is able to predict the state for multiple targets. However, it cannot deal with collision, grouping events, or non-smooth changes in position or shape. In order to address these issues we implemented a high level tracker which

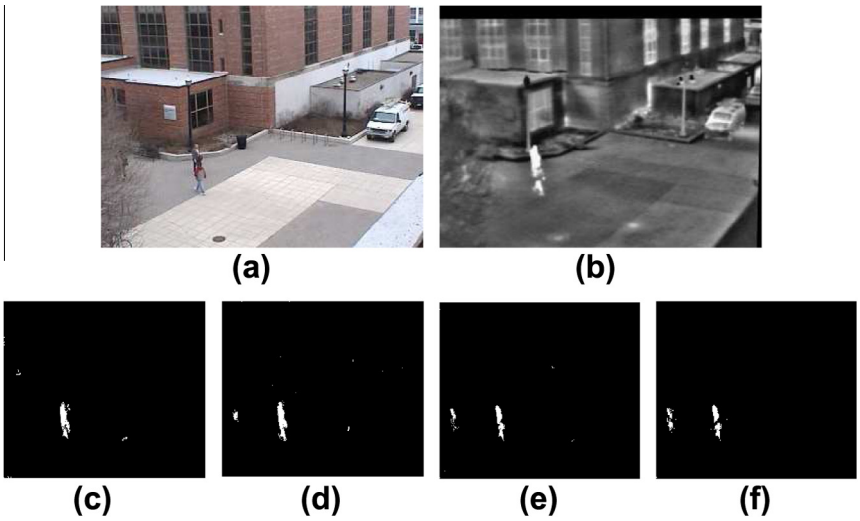

Fig. 6. (a) Color image, (b) IR image, (c) GM, (d) GGM, (e) IGGM, and (f) AGGM.

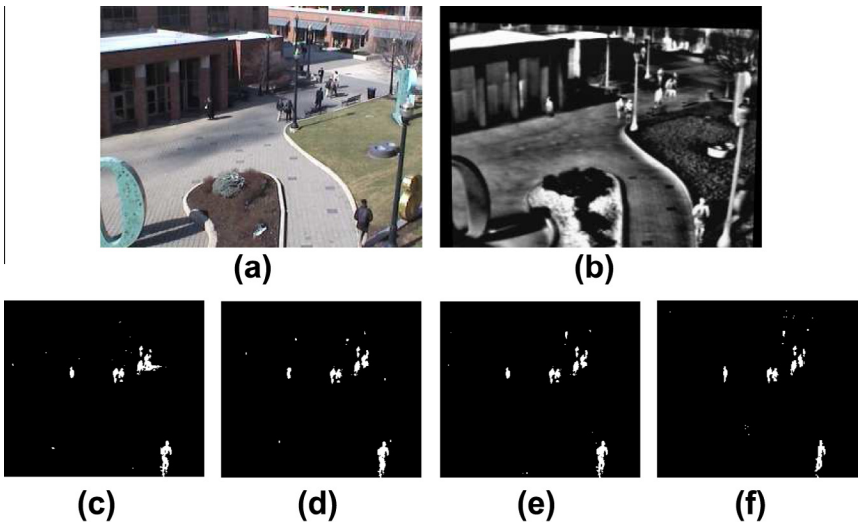

Fig. 7. (a) Color image, (b) IR image, (c) GM, (d) GGM, (e) IGGM, and (f) AGGM.

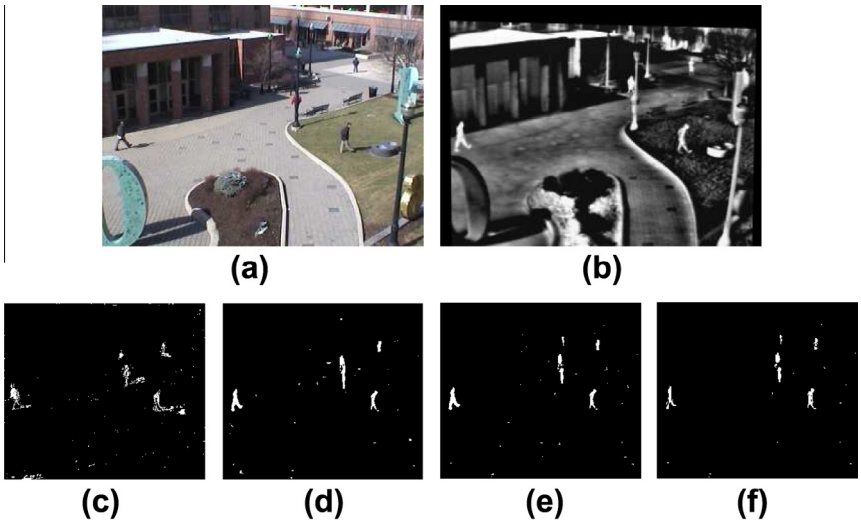

Fig. 8. (a) Color image, (b) IR image, (c) GM, (d) GGM, (e) IGGM, and (f) AGGM.

Table 2
Combined performance measure [40].

GM	GGM	IGGM	AGGM
0.43	0.31	0.26	0.28

deals with object appearance. Collins et al. [44] introduced an appearance based method for tracking found on the idea that the features which best discriminate between object and background are also efficient for tracking the object. They used multiple color histograms as they are less sensitive to rotations or target deformation. For each object, they have chosen two regions: the first containing the object itself and the other consisting of the background wrapping the object. They collect a pool of 49 different histograms by using different combinations of the RGB color space for each of the two regions. Then, using a log likelihood metric between the object and the background histograms, they select a pool of best features to determine if the object corresponds to a previously tracked object. However, this method does not work effectively in outdoor environment due to high illumination changes. In order to solve this problem we used a linear combination of both RGB and infrared features. Thus, the set of candidate features is composed of linear combinations of camera R, G, B pixel values with infrared intensity values I. In particular, we have chosen the following set of feature-space candidates for our experiments:

$$h = w_1 \times I + w_2 \times R + w_3 \times G + w_4 \times B \tag{42}$$

where $w_a \in (-1,0,1)$; $a = (1,\ldots,4)$. Eq. (42) represents linear combinations composed of integer coefficients between -1 and 1. The total number of such candidates would be 3^4, however, by excluding redundant coefficients and by disallowing $(w_1; w_2; w_3; w_4) = (0; 0; 0; 0)$, we are left with a pool of 43 features. Features are then normalized and discretized into 64 bits, and their log-likelihood ratios are calculated. The log-likelihood ratio of the ith feature can be computed as:

$$L^i(k) = \log \frac{\max(p_k^i, \in)}{\max(q_k^i, \in)} \tag{43}$$

where $\in$ is chosen as the minimum histogram value to prevent dividing by zero or taking the logarithm of zero, p_k^i and q_k^i are the kth bin of the ith feature of the target and background histogram,

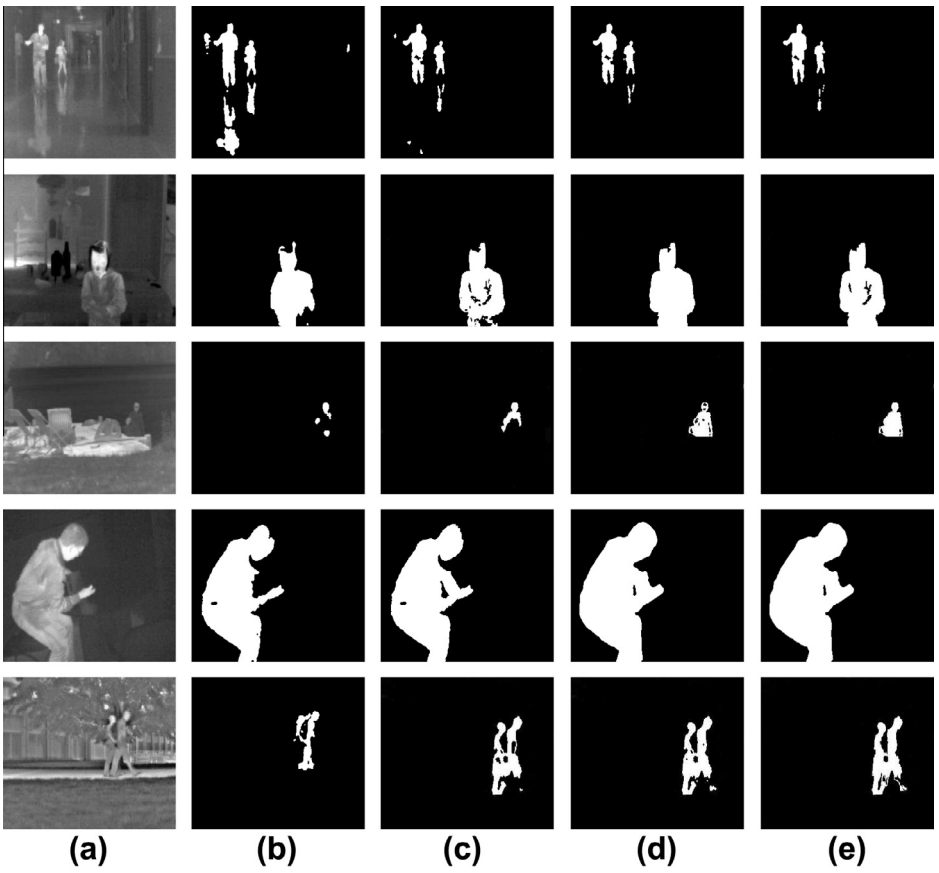

Fig. 9. (a) IR image, (b) GM, (c) GGM, (d) IGGM, and (e) AGGM.

Table 3
Precision, recall, and *F*-measure.

	Zivkovic [33]	KaewTraKulPong and Bowden [42]	Evangelio et al. [43]	GM	GGM	IGGM	AGGM
Corridor							
Prec. (%)	83.93	96.20	85.84	79.75	83.96	87.19	86.93
Rec. (%)	83.26	65.71	84.68	76.51	82.34	84.26	83.68
F-measure	0.84	0.78	0.85	0.78	0.83	0.86	0.85
DiningRoom							
Prec. (%)	92.31	98.54	94.03	92.03	94.32	97.54	97.18
Rec. (%)	69.43	43.16	77.45	70.42	77.21	81.01	82.16
F-measure	0.79	0.61	0.85	0.80	0.85	0.89	0.89
Lakeside							
Prec. (%)	92.23	94.10	96.86	92.04	93.23	96.58	97.86
Rec. (%)	36.41	20.59	35.80	39.88	41.88	45.91	42.13
F-measure	0.52	0.34	0.52	0.56	0.58	0.62	0.59
Library							
Prec. (%)	81.76	96.68	93.88	84.76	93.86	95.30	96.51
Rec. (%)	28.68	24.03	30.23	25.03	26.56	31.38	28.32
F-measure	0.42	0.38	0.46	0.39	0.41	0.47	0.44
Park							
Prec. (%)	85.07	99.95	92.57	80.07	85.07	86.53	86.57
Rec. (%)	59.30	16.24	39.68	47.38	59.96	63.30	61.98
F-measure	0.70	0.28	0.56	0.60	0.70	0.73	0.72

respectively. Then, features are evaluated according to the variance-ratio of the log-likelihood:

$$VR^i(k) = var\frac{\max\left(p_k^i, \in\right)}{\max\left(q_k^i, \in\right)} \tag{44}$$

Thus, features are ranked according to their variance ratio (the higher is the better). In this application, opposed to the method of [44], long-run features are kept and smoothed to be used in the case of target loss recovery. For each time t, the best M features are chosen and kept for N long-run in order to recursively compute the mean appearance histogram for each of the M features. Then, the mean appearance histogram of the ith feature at time t can be computed by:

$$m_t^i = m_{t-1}^i + \frac{1}{n_i}\left(p_t^i - m_t^i\right) \tag{45}$$

Fig. 10. Low level tracking example using the ellipse centroid metric.

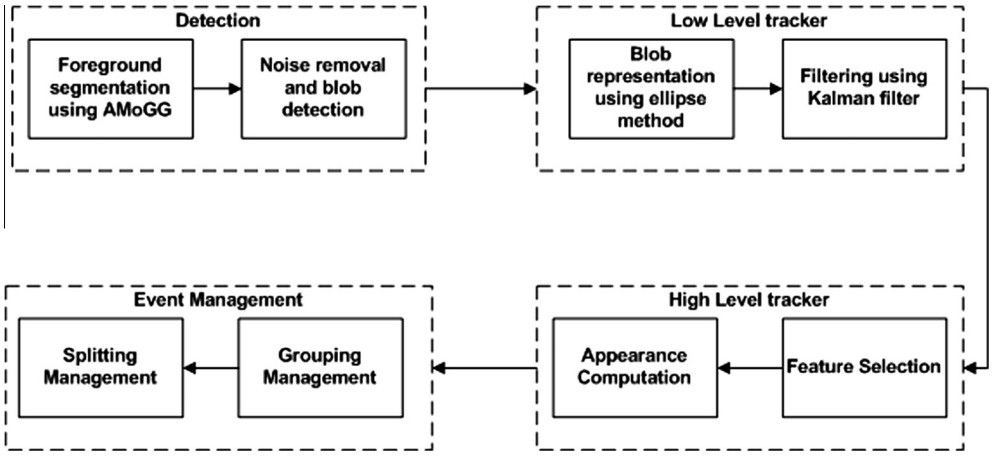

Fig. 11. System architecture.

where n_i is the number of times the histogram has been updated. Later, similarity between the two histograms (of the two different frames) is computed using the Bhattacharyya distance $d_B = \sqrt{1 - \sum_{k=1}^{K} \sqrt{p_k q_k}}$. Then, the mean and the variance of d_B between the smoothed histogram and the new one are computed and updated in order to recognize if the two histograms are close enough:

$$\mu_t^i = \mu_{t-1}^i + \frac{1}{n-1}\left(d_{B_t}^i - \mu_t^i\right) \tag{46}$$

$$\sigma_t^2 = \frac{n-3}{n-2}\sigma_{t-1}^2 + (n-1)(\mu_t^i - \mu_{t-1}^i) \tag{47}$$

3.2.4. Complete framework

From the aforementioned sections, two trackers are used in this method, each with its merits and demerits. The low level tracker is better when dealing with isolated objects, on the contrary, high level tracker is better when facing occlusion or object collision. Therefore, we fused both tracker together to improve tracking performance. Fig. 11 shows our system architecture. First, our method try to detect collision (splitting/grouping), then the low level detector is used for single tracking, finally the high level tracker is applied to find the tracking object that were occluded in previous frames. Grouping is identified if two or more different ellipse centroids fit within a new ellipse in the scene. And splitting is detected if two or more new ellipses fit within a tracking object predicted ellipse. Fig. 12 shows some collision detection examples.

3.2.5. Results

Our approach performance has been evaluated using the OSU Color-Thermal Database [32]. This benchmark contains six videos that where shot using both thermal and color cameras. Videos are taken in outdoor environment and are widely applied for persistent object detection in urban settings. These video sequences include isolated objects, occlusions, and splitting/grouping events. We have compare our method with the one proposed by Rowe et al. [31]. Table 4 represents the performance of the two methods with only visible camera as input and with both thermal and visible used as input. The performance here is based on the capability to identify and track objects, grouping and splitting events, and to recover occlusions. From this table we can conclude that the use of AGGM in foreground segmentation with the fusion of infrared and color cameras increased the performance greatly.

4. Conclusion

In this paper, we have proposed the consideration of AGGM models for applications involving multidimensional non-Gaussian asymmetric data. In particular, we have developed a principled learning approach to fit this kind of data. Our learning technique

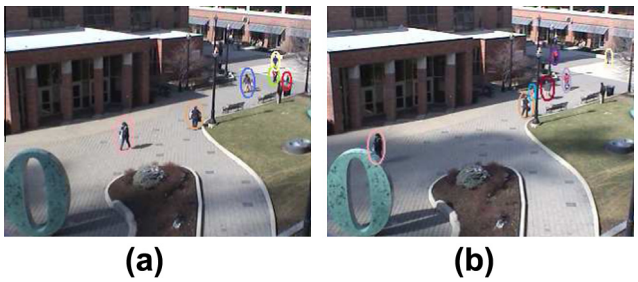

Fig. 12. Collision detection example. (a) Object before collision. (b) Grouping and splitting events identified.

Table 4
Multiple target tracking results. The performance here is based on the capability to identify and track objects, grouping and splitting events, and to recover occlusions.

	Total	Our Method		Rowe et al. [31]	
		Visible	Fusion	Visible	Fusion
1st Video					
Tracked objects	4	4	4	4	4
Grouping events	0	0	0	0	0
Splitting events	0	0	0	0	0
Occlusions recovered	0	0	0	0	0
2nd Video					
Tracked objects	2	2	2	2	2
Grouping events	1	1	1	1	1
Splitting events	1	1	1	1	1
Occlusions recovered	0	0	0	0	0
3rd Video					
Tracked objects	4	4	4	4	4
Grouping events	1	1	1	0	0
Splitting events	1	1	1	1	1
Occlusions recovered	1	1	1	1	1
4th Video					
Tracked objects	8	7	7	7	7
Grouping events	2	2	2	1	2
Splitting events	1	1	1	1	1
Occlusions recovered	1	0	1	1	1
5th Video					
Tracked objects	18	18	18	15	15
Grouping events	4	3	4	3	3
Splitting events	3	3	3	1	1
Occlusions recovered	5	4	5	2	2
6th Video					
Tracked objects	15	13	14	12	13
Grouping events	5	5	5	5	5
Splitting events	4	3	4	2	4
Occlusions recovered	8	8	8	4	4

is based on an EM algorithm which goal is to minimize a message length objective in order to estimate and select simultaneously the mixture's parameters and its model order (i.e. number of components), respectively. Extensive experiments involving challenging applications namely pedestrian detection and MTT have shown the merits of the proposed statistical framework. We have also demonstrated the importance of the fusion of both visible and infrared images for MTT. Future works can be devoted, for instance, to the incorporation of a feature selection step into the estimation of the mixture in order to speed up learning and to improve model accuracy and generalization capabilities.

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

$$\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \mu_{jk}} = A(\beta_{jk})\beta_{jk}\left[\sum_{i=1,X_{ik}\geq\mu_{jk}}^{N} Z_{ij}\frac{(X_{ik}-\mu_{jk})^{\beta_{jk}-1}}{\sigma_{r_{jk}}^{\beta_{jk}}} - \sum_{i=1,X_{ik}<\mu_{jk}}^{N} Z_{ij}\frac{(\mu_{jk}-X_{ik})^{\beta_{jk}-1}}{\sigma_{l_{jk}}^{\beta_{jk}}}\right] \tag{A.1}$$

$$-\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \mu_{jk}^2} = A(\beta_{jk})\beta_{jk}(\beta_{jk}-1)\left[\sum_{i=1,X_{ik}<\mu_{jk}}^{N} Z_{ij}\frac{(\mu_{jk}-X_{ik})^{\beta_{jk}-2}}{\sigma_{l_{jk}}^{\beta_{jk}}} + \sum_{i=1,X_{ik}\geq\mu_{jk}}^{N} Z_{ij}\frac{(X_{ik}-\mu_{jk})^{\beta_{jk}-2}}{\sigma_{r_{jk}}^{\beta_{jk}}}\right] \tag{A.2}$$

$$\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \mu_{jk_1}\mu_{jk_2}} = 0 \tag{A.3}$$

$$\begin{aligned}\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \beta_{jk}} &= \sum_{i=1}^{N} Z_{ij}\left[\frac{1}{\beta_{jk}} - \frac{3}{2}\left(\frac{\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{\beta_{jk}^2}\right)\right] \\ &+ \sum_{i=1,X_{ik}<\mu_{jk}}^{N} Z_{ij}A(\beta_{jk})\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta_{jk}}\left[\left(\frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}}\right) - \log\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)\right] \\ &+ \sum_{i=1,X_{ik}\geq\mu_{jk}}^{N} Z_{ij}A(\beta_{jk})\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}}\left[\left(\frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}}\right) - \log\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)\right]\end{aligned} \tag{A.4}$$

$$\begin{aligned}-\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \beta_{jk}^2} &= \sum_{i=1}^{N} Z_{ij}\left[\frac{1}{\beta_{jk}^2} + \frac{3\Psi'(1/\beta_{jk})}{2\beta_{jk}^4} + 3\frac{\Psi(1/\beta_{jk})-\Psi(3/\beta_{jk})}{\beta_{jk}^3} - \frac{9\Psi'(3/\beta_{jk})}{2\beta_{jk}^4}\right] \\ &+ A(\beta_{jk})\sum_{i=1,X_{ik}<\mu_{jk}}^{N} Z_{ij}\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta_{jk}}\left[\left(\frac{9\Psi'(3/\beta_{jk})-\Psi'(1/\beta_{jk})}{2\beta_{jk}^3} + \frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}^2}\right)\right. \\ &\left. + \left(\frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}} - \log\left[\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right]\right)^2\right] \\ &+ A(\beta_{jk})\sum_{i=1,X_{ik}\geq\mu_{jk}}^{N} Z_{ij}\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}}\left[\left(\frac{9\Psi'(3/\beta_{jk})-\Psi'(1/\beta_{jk})}{2\beta_{jk}^3} + \frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}^2}\right)\right. \\ &\left. + \left(\frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}} - \log\left[\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right]\right)^2\right]\end{aligned} \tag{A.5}$$

$$\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \beta_{jk_1}\beta_{jk_2}} = 0 \tag{A.6}$$

$$\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \sigma_{l_{jk}}} = \sum_{i=1,X_{ik}<\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})\beta_{jk}}{\sigma_{l_{jk}}}\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta_{jk}} - \sum_{i=1}^{N}\frac{Z_{ij}}{\sigma_{l_{jk}}+\sigma_{r_{jk}}} \tag{A.7}$$

$$\begin{aligned}-\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma_{l_{jk}}^2} &= \sum_{i=1,X_{ik}<\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})\beta_{jk}(\beta_{jk}+1)}{\sigma_{l_{jk}}^2}\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta_{jk}} \\ &- \sum_{i=1}^{N}\frac{Z_{ij}}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2}\end{aligned} \tag{A.8}$$

$$\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma_{l_{jk_1}}\sigma_{l_{jk_2}}} = 0 \tag{A.9}$$

$$\frac{\partial L(\Theta,Z,\mathcal{X})}{\partial \sigma_{r_{jk}}} = \sum_{i=1,X_{ik}\geq\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})\beta_{jk}}{\sigma_{r_{jk}}}\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}} - \sum_{i=1}^{N}\frac{Z_{ij}}{\sigma_{l_{jk}}+\sigma_{r_{jk}}} \tag{A.10}$$

$$\begin{aligned}-\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma_{r_{jk}}^2} &= \sum_{i=1,X_{ik}\geq\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})\beta_{jk}(\beta_{jk}+1)}{\sigma_{r_{jk}}^2}\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}} \\ &- \sum_{i=1}^{N}\frac{Z_{ij}}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2}\end{aligned} \tag{A.11}$$

$$\frac{\partial^2 L(\Theta,Z,\mathcal{X})}{\partial \sigma_{r_{jk_1}}\sigma_{r_{jk_2}}} = 0 \tag{A.12}$$

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

References

[1] E.W. Kerce, Boredom at work: implications for the design of jobs with variable requirements, Navy Research (1985).
[2] L. St-Laurent, X. Maldague, D. Prevost, Combination of colour and thermal sensors for enhanced object detection, in: The 10th International Conference on Information Fusion, 2007, pp. 1–8.
[3] F. De la Torre, E. Martinez, M.E. Santamaria, J.A.Moran. Moving object detection and tracking system: a real-time implementation, in: Symposium on Signal and Image Processing, 1997, pp. 375–378.
[4] L.M. Fuentes, S.A. Velastin, People tracking in surveillance applications, Image and Vision Computing 24 (11) (2006) 1165–1171.
[5] I. Haritaoglu, D. Harwood, L.S. Davis, W4: real-time surveillance of people and their activities, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 809–830.

[6] D.A. Socolinsky, A. Selinger, J.D. Neuheisel, Face recognition with visible and thermal infrared imagery, Computer Vision and Image Understanding 91 (1–2) (2003) 72–114.
[7] X. Chen, P.J. Flynn, K.W. Bowyer, IR and visible light face recognition, Computer Vision and Image Understanding 99 (3) (2005) 332–358.
[8] C. Dai, Y. Zheng, X. Li, Pedestrian detection and tracking in infrared imagery using shape and appearance, Computer Vision and Image Understanding 106 (2–3) (2007) 288–299.
[9] I. Pavlidis, P. Symosek, The imaging issue in an automatic face/disguise detection system, in: IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications (CVBVS), 2000, pp. 15–24.
[10] D.B. Rensch, R.K. Long, Comparative studies of extinction and backscattering by aerosols, fog, and rain at 10.6μ and 0.63μ, Applied Optics 9 (7) (1970) 1563–1573.
[11] M.A. Naboulsi, H. Sizun, F. de Fornel, Fog attenuation prediction for optical and infrared waves, Optical Engineering 43 (2) (2004) 319–329.
[12] A. Torabi, G. Masse, G.-A. Bilodeau, Feedback scheme for thermal-visible video registration, sensor fusion, and people tracking, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010, pp. 15–22.
[13] A. Leykin, R. Hammoud, Pedestrian tracking by fusion of thermal-visible surveillance videos, Machine Vision and Applications 21 (4) (2008) 587–595.
[14] M.S. Allili, N. Bouguila, D. Ziou, Finite general Gaussian mixture modeling and application to image and video foreground segmentation, Journal of Electronic Imaging 17 (1) (2008) 1–13.
[15] T. Elguebaly, N. Bouguila, Bayesian learning of finite generalized Gaussian mixture models on images, Signal Processing 91 (4) (2011) 801–820.
[16] H. Akaike, A new look at the statistical model identification, IEEE Transaction on Automatic Control 19 (6) (1974) 716–723.
[17] J. Rissanen, Modeling by shortest data description, Automatica 14 (1987) 465–471.
[18] G.J. McLachlan, D. Peel, Finite Mixture Models, Wiley, New York, 2000.
[19] N. Bouguila, D. Ziou, A Dirichlet process mixture of generalized Dirichlet distributions for proportional data modeling, IEEE Transaction on Neural Networks 21 (1) (2010) 107–122.
[20] C.S. Wallace, D.M. Boulton, An information measure for classification, The Computer Journal 11 (2) (1968) 195–209.
[21] J.-Y. Lee, A.K. Nandi, Parameter estimation of the asymmetric generalised Gaussian family of distributions, in: IEE Colloquium on Statistical Signal Processing, 1999, pp. 9/1–9/5.
[22] N. Nacereddine, S. Tabbone, D. Ziou, L. Hamami, Asymmetric generalized Gaussian mixture models and EM algorithm for image segmentation, in: The 2010 International Conference on Pattern Recognition, 2010, pp. 4557–4560.
[23] T. Elguebaly, N. Bouguila, A nonparametric bayesian approach for enhanced pedestrian detection and foreground segmentation, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2011, pp. 21–26.
[24] R.A. Baxter, J.J. Olivier, Finding overlapping components with MML, Statistics and Computing 10 (1) (2000) 5–16.
[25] G.J. McLachlan, T. Krishnan, The EM Algorithm and Extensions, Wiley-Interscience, New York, 1997.
[26] O. Tuzel, F.M. Porikli, P. Meer, Pedestrian detection via classification on riemannian manifolds, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (10) (2008) 1713–1727.
[27] C.E. Rasmussen, The infinite Gaussian mixture model, Advances in Neural Information Processing Systems (NIPS) (2000) 554–560.
[28] J. Davis, M. Keck, A two-stage approach to person detection in thermal imagery, in Workshop on Applications of Computer Vision, IEEE OTCBVS WS Series Bench, 2005, pp. 364–369.
[29] L. Maddalena, A. Petrosino, A self-organizing approach to background subtraction for visual surveillance application, IEEE Transactions on Image Processing 17 (7) (2008) 1168–1177.
[30] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–577.
[31] D. Rowe, I. Reid, J. Gonz'alez, J.J. Villanueva, Unconstrained multiplepeople tracking, In Lecture Notes in Computer Science (2006) 505–514.
[32] J. Davis, V. Sharma, Background-subtraction using contour-based fusion of thermal and visible imagery, Computer Vision and Image Understanding 106 (2–3) (2007) 162–182.
[33] Z. Zivkovic, Improved adaptive Gaussian mixture model for background subtraction, in: The 17th International Conference on Pattern Recognition, 2004, pp. 28–31.
[34] A. Shimada, D. Arita, R. Taniguchi, Dynamic control of adaptive mixture-of-Gaussians background model, in: IEEE International Conference on Video and Signal Based Surveillance, 2006, p. 5.
[35] R. Tan, H. Huo, J. Qian, T. Fang, Traffic video segmentation using adaptive-K Gaussian mixture model, in: International Conference on Intelligent Computing in Pattern Analysis/Synthesis, 2006, pp. 125–134.
[36] T. Bouwmans, F. El Baf, B. Vachon, Background modeling using mixture of Gaussians for foreground detection – a survey, Recent Patents on Computer Science 1 (3) (2008) 219–237.
[37] C. Stauffer, E. Grimson, Adaptive background mixture models for real-time tracking, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1999, pp. 252–258.
[38] J. Cheng, J. Yang, Y. Zhou, Y. Cui, Flexible background mixture models for foreground segmentation, Image and Vision Computing 24 (5) (2006) 473–482.
[39] D. Wang, W. Xie, J. Pei, Z. Lu, Moving area detection based on estimation of static background, Journal of Information and Computing Science 2 (1) (2005) 129–134.
[40] C. Erdem, B. Sankur, A.M. Tekalp, Performance measures for video object segmentation and tracking, IEEE Transactions on Image Processing 13 (7) (2004) 937–951.
[41] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, P. Ishwar, changedetection.net: A new change detection benchmark dataset, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, pp. 1–8.
[42] P. KaewTraKulPong, R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection, in: Workshop on Advanced Video Based Surveillance Systems, 2001.
[43] R.H. Evangelio, M. Pätzold, and T Sikora, Splitting Gaussians in mixture models, in: IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance (AVSS), 2012, pp. 300–305.
[44] R. Collins, Y. Liu, M. Leordeanu, Online selection of discriminative tracking features, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1631–1643.