Unsupervised Selection and Estimation of Non-Gaussian Mixtures for High Dimensional Data Analysis

Tarek Elguebaly

A Thesis
in
The Department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of PhD (Electrical and Computer Engineering) at
Concordia University
Montréal, Québec, Canada

June 2014

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Tarek Elguebaly

Entitled: Unsupervised Selection and Estimation of Non-Gaussian Mixtures for High Dimensional Data Analysis

and submitted in partial fulfillment of the requirements for the degree of

PhD (Electrical and Computer Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Adel M. Hanna Chair

Dr. Fakhreddine Karray External Examiner

Dr. Hoi Dick Ng External to Program

Dr. Abdessamad Ben Hamza Examiner

Dr. Walaa Hamouda Examiner

Dr. Nizar Bouguila Thesis Supervisor

Approved by

Chair of Department or Graduate Program Director

Dean of Faculty

Abstract

Unsupervised Selection and Estimation of Non-Gaussian Mixtures for High Dimensional Data Analysis

Tarek Elguebaly, Ph.D.
Concordia University, 2014

Lately, the enormous generation of databases in almost every aspect of life has created a great demand for new, powerful tools for turning data into useful information. Therefore, researchers were encouraged to explore and develop new machine learning ideas and methods. Mixture models are one of the machine learning techniques receiving considerable attention due to their ability to handle efficiently and effectively multidimensional data. Generally, four critical issues have to be addressed when adopting mixture models in high dimensional spaces: (1) choice of the probability density functions, (2) estimation of the mixture parameters, (3) automatic determination of the number of components M in the mixture, and (4) determination of what features best discriminate among the different components. The main goal of this thesis is to summarize all these challenging interrelated problems in one unified model.

In most of the applications, the Gaussian density is used in mixture modeling of data. Although a Gaussian mixture may provide a reasonable approximation to many real-world distributions, it is certainly not always the best approximation especially in computer vision and image processing applications where we often deal with non-Gaussian data. Therefore, we propose to use three highly flexible distributions: the generalized Gaussian distribution (GGD), the asymmetric Gaussian distribution (AGD), and the asymmetric generalized Gaussian distribution (AGGD). We are motivated by the fact that these distributions are able to fit many distributional shapes and then can be considered as a useful class of flexible models to address several problems and applications involving measurements and features having well-known marked deviation from the Gaussian shape.

Recently, researches have shown that model selection and parameter learning are highly dependent and should be performed simultaneously. 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 a set of candidate models using the Expectation-Maximization (EM) framework, then choose the model that maximizes a model selection criterion. 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. Hence, in this thesis, we propose three learning techniques capable of automatically determining model complexity while learning its parameters. First, we incorporate a Minimum Message Length (MML) penalty in the model learning step performed using the EM algorithm. Our second approach employs the Rival Penalized EM (RPEM) algorithm which is able to select an appropriate number of densities by fading out the redundant densities from a density mixture. Last but not least, we incorporate the nonparametric aspect of mixture models by assuming a countably infinite number of components and using Markov Chain Monte Carlo (MCMC) simulations for the estimation of the posterior distributions. Hence, the difficulty of choosing the appropriate number of clusters is sidestepped by assuming that there are an infinite number of mixture components.

Another essential issue in the case of statistical modeling in general and finite mixtures in particular is feature selection (i.e. identification of the relevant or discriminative features describing the data) especially in the case of high-dimensional data. Indeed, feature selection has been shown to be a crucial step in several image processing, computer vision and pattern recognition applications not only because it speeds up learning but also because it improves model accuracy and generalization. Moreover, the learning of the mixture parameters (i.e. both model selection and parameters estimation) is greatly affected by the quality of the features used. Hence, in this thesis, we are trying to solve the feature selection problem in unsupervised learning by casting it as an estimation problem, thus avoiding any combinatorial search. Finally, the effectiveness of our approaches is evaluated by applying them to different computer vision and image processing applications.

Acknowledgements

I owe my deepest gratitude to my supervisor, Dr. Nizar Bouguila, for his continuous support and encouragement throughout my graduate studies. It was an honor for me to work with such a wonderful advisor.

I would like to thank the members of my committee for their encouragement, insightful comments, and advices. Also I am thankful to my fellow lab-mates at Concordia University for their support, motivation and all the good time we had together. I am also grateful to Concordia University and to the Natural Sciences and Engineering Research Council (NSERC) of Canada for supporting my research during my graduate studies.

Finally, I would like to thank my family for unconditional support throughout my studies, your endless love and care always encourage me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Over the last decade, technological advances have brought an explosion of enormous data not only in size but also in dimension. These data pose a challenge to standard statistical methods and comparatively recently have received much attention. The importance of finding a way to model and analyze multidimensional data lies in their usefulness in wide range of applications such as image processing and computer vision. Modeling and finding valuable information in multidimensional data depend on recognizing complex patterns, regularities, and relationships in data. In recent years various algorithms were developed in the aim of automatically learning to recognize complex patterns, and to produce intelligent decisions based on observed data. Machine learning is the branch of artificial intelligence that offers a principled approach for developing and studying automatic techniques capable of learning models and their parameters based on training data [8–11]. Machine learning and statistical pattern recognition have seen dramatic growth over the past few years, this explosion is ascribable to the fact that they can be applied in diverse areas such as engineering, medicine, computer science, psychology, neuroscience, physics, and mathematics [12, 13]. Recent advances in machine learning fascinated researchers from different fields because they offer promise for the development of novel supervised and unsupervised methods that can help in modeling and analyzing different data.

A broad range of tasks in computer vision may be viewed as unsupervised partitioning of data. Image and video segmentation and multimedia database categorization are two problems with two different application objectives that use low and high level visual information, respectively. However, they are all built on the same idea, which is the partitioning of the visual entities (pixels or images) into clusters or parts similar in their own composition and different when it comes to comparison to each others. The practice of classifying objects and patterns according to perceived similarities is the basis of most image processing and computer vision applications. This task is known as Clustering or cluster analysis and is one of the most fundamental modes of understanding and learning for humans and machines. Clustering is the task of grouping various objects into different groups where objects in the same group are more similar to each other than to those in other groups. Clustering approaches can be categorized based on their cluster model into hierarchical, relocation, probabilistic, density based, and grid based. Hierarchical techniques create the clusters gradually and exploit the connectivity matrix which express the similarity between data items. Two main directions to hierarchical clustering exist: the agglomerative approach which starts with a set of singleton clusters containing only one element and iteratively merge pairs of clusters and the divisive approach which begins with a single cluster containing all objects and iteratively splits it to different clusters. Relocation algorithms do not build the clusters gradually, but they start with a randomly generated partition, then, relocate data items among existing clusters in order to improve them. Usually these methods require an apriori-fixed number of clusters. The most used method in this category is the K-means approach which uses an iterative procedure of two alternate steps: the data assignment, and the update of the centroids values. Probabilistic methods were built on the idea that the data set corresponds to a sample independently drawn from a mixture of several populations. Density-based clustering methods regard clusters as high density regions in the feature space separated by low density regions. This interpretation has the superiority of detecting clusters of arbitrary shapes. These methods use two main concepts density and connectivity which take into account the local distribution in data and necessitate the definition of neighborhood in data and nearest neighbors computations. Grid-based clustering algorithms segment the feature space and then aggregate dense neighbor segments. A segment is a multi-rectangular region in the feature space that results from the Cartesian product of individual feature sub-ranges. Thus, data partitioning is practically achieved through space partitioning. There are some grid-based methods that prune the attribute space in an apriori manner, thus, performing subspace clustering which can be critical in case of high-dimensional data, when irrelevant features can mask the grouping tendency. Therefore, It can be considered as an extension of traditional clustering that seeks to find clusters in different subspaces within a data set. In this thesis, we are interested with probabilistic approaches and especially mixture models.

Mixture models are one of the machine learning techniques receiving considerable attention in different applications. Mixture models are normally used to model complex data sets by assuming that each observation has arisen from one of the different groups or components [14]. Moreover, mixture models have been successfully applied in different tasks such as clustering and density estimation.

1.1 Mixture Models

A mixture model is formed by taking linear combinations of a number of basic distributions. These basic distributions are called components of the mixture model. For instance, a mixture model with M components is given by:

$$p(X) = \sum_{j=1}^{M} p_j p(X|\theta_j) \tag{1.1}$$

where each component has a probability distribution $p(X|\theta_j)$ with parameters θ_j and a given weight p_j. The sum of the weights of all components is equal to one and M represents the total number of components. Mixture models can be finite or infinite [14, 15] depending on the number of components M in the model. Finite mixture models deal with a countably finite number of components. On the other hand, infinite mixture models allow M to increase to infinity. In order to use mixture models, three main points have to be identified: the choice of the probability density function (PDF), the approaches used for parameters estimation, and the selection of the number of components. Another essential issue in the case of statistical modeling in general and mixture models in particular is feature selection (i.e. identification of the relevant or discriminative features describing the data) especially in the case of high-dimensional data.

1.1.1 Probability Density Function Selection

Mixture models are convex combinations of two or more PDFs. By combining the properties of the individual PDFs, mixture models are capable of approximating any arbitrary distribution. Therefore, selecting the most accurate PDF that best represent the mixture components is of a crucial importance, because it affects the capability of the mixture to represent the data shape. Furthermore, the wrong selection of PDF may force the mixture model to increase the number of components in order to model the data (i.e overfitting). One of the most fundamental and widely used statistical models is the mixture of Gaussians which is generally justified for asymptotic reasons (i.e. the sample is supposed to be sufficiently large) [16]. However, it has been observed that the Gaussian distribution is generally an inappropriate choice to model data in complex real life applications [17]. For instance, it is well-known that natural image clutter is generally non-Gaussian. Many studies have shown that the generalized Gaussian distribution (GGD), that we consider in the second Chapter of this thesis, can be a good alternative to the Gaussian thanks to its shape flexibility which allows the modeling of a large number of non-Gaussian signals [18–21]. The GGD contains the Laplacian, the Gaussian and asymptotically the uniform distribution as special cases [22] and has been used in many challenging problems (see, for instance [21, 23, 24]). However, the GGD is still a symmetrical distribution inappropriate to model non-symmetrical data. Therefore, in the rest of this thesis, we suggest the consideration of two non-symmetrical distributions: the asymmetric Gaussian and the asymmetric generalized Gaussian.

1.1.2 Parameters Learning

Parameter learning approaches are used in order to estimate the model parameters. This problem is not straightforward and many deterministic as well as Bayesian approaches have been proposed. In deterministic approaches, parameters are assumed as fixed and unknown, and inference is founded on the likelihood of the data. Normally, the EM algorithm is employed to find maximum likelihood solutions for mixture models. However, the EM algorithm needs an appropriate predefined number of components, otherwise, it will lead to a poor result. Furthermore, many works have proved that deterministic methods have severe problems such as convergence to local maxima, and the tendency to complicate the resulted models. On the other hand, Bayesian Markov Chain Monte Carlo (MCMC) methods consider parameters to be random, and to follow different probability distributions (prior distributions). These distributions describe our knowledge before considering the data, as for updating our prior beliefs the likelihood is used. Despite the fact that MCMC techniques have revolutionized Bayesian statistics by accommodating situations characterized by uncertainty of the statistical model structure [16], their use is often limited to small-scale problems in practice because of their high computational cost and the difficulty in tracking convergence [16, 25].

1.1.3 Selection of the number of components

Another crucial issue when using mixture models is the selection of the number of components or model complexity. The usual tradeoff in model complexity determination problem arises: with too many components, the mixture may overfitt the data, while a mixture with too few components may not be flexible enough to approximate the true underlying model. Lack of knowledge about the number of clusters is a challenging problem in mixture modeling and considerable efforts already have been made to investigate this important aspect. In the past decades, a lot of research has been devoted to the automatic selection of the number of clusters which best describe a given data set (see, for instance, [26–28]). Most of the literature on model selection can be broadly divided into deterministic and Stochastic.

Deterministic approaches can be further partitioned into two groups. The first category estimates the model parameters for different ranges of M then chooses the value that maximizes a model selection criterion such as Akaike's information criterion (AIC) [29], minimum description length (MDL) [30] and Laplace empirical criterion (LEC) [14]. Despite the popularity of these approaches, these conventional criteria may overestimate or underestimate the number of clusters due to the difficulty of choosing an appropriate penalty function. Furthermore, they may be time-consuming and lead to a sub-optimal solution because model selection and parameters estimation are determined in two separate steps. In contrast, the other direction is to introduce algorithms capable of automatically estimating the model parameters and selecting the number of components simultaneously. Hence, this category generally gives a promising way to develop a robust clustering algorithm in terms of number of clusters. One of the widely used methods in this category is to incorporate a Minimum Message Length (MML) penalty in the model learning step [31, 32]. This can be done by choosing a large initial value for M and deriving the structure of the mixture by letting the estimates of some of the mixing probability to be zero. Therefore, this method aims at finding the best overall model in the entire set of available models rather than selecting one among a set of candidate models [32]. Furthermore, the work in [33] introduced the rival penalized competitive learning (RPCL) algorithm which can automatically select the number of clusters during learning via penalizing the rival in competition. The basic idea of the RPCL is that for each input not only the winner of the input sample is updated to adapt to the input, but also its rival is de-learned by a smaller de-learning rate. Many experiments have shown that the RPCL can indeed automatically select the correct cluster number by gradually driving extra seed points far away from the input data set. However, its performance is sensitive to the selection of the de-learning rate, such that if it is not well selected, the RPCL may completely break down. In order to overcome this problem, the rival penalized controlled competitive learning (RPCCL) was introduced in [34]. This algorithm sets the de-learning rate at the same value as the learning rate, then dynamically adjust it based on the relative distance of the winner to the rival and the current input, respectively. In [35], the Rival Penalized EM (RPEM) algorithm was proposed for density mixture clustering. The RPEM learns the model parameters by making the mixture components compete with each other at each time step; this can be done by not only updating the winning density component parameters to adapt to the input but also all rivals parameters are penalized with the strength proportional to the corresponding posterior density probabilities. Therefore, the RPEM is able to automatically select an appropriate number of densities by fading out the redundant densities from a density mixture which can save computing time.

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 [36]. Despite their formal appeal, MCMC methods are too computationally demanding, therefore can't be applied efficiently in several complex applications.

1.1.4 Feature Selection

Another essential issue in the case of statistical modeling in general and finite mixtures in particular is feature selection (i.e. identification of the relevant or discriminative features describing the data) especially in the case of high-dimensional data which analysis has been the topic of extensive research in the past. This is actually an important problem, since the main goal is not only the determination of clusters and their parameters but also to provide the most parsimonious model that can accurately describe the data. Moreover, it is noteworthy that the way employed by humans in clustering and recognition is based on formulating few selected features (i.e. humans pick up just the relevant information and ignore the irrelevant [37]) and clustering the data on the basis of these features [38]. Furthermore, feature selection can speed up learning and improve model accuracy and generalization. Therefore, feature selection has been shown to be a crucial step in several image processing, computer vision and pattern recognition applications such as object detection [39], handwriting separation [40], image retrieval, categorization and recognition [41]. However, the majority of research in mixture models assumes that all features have the same weight and uses a pre-processing step such as principal components analysis to transform the original features into a new dimension-reduced space. The main drawback of that approach is that the physical meaning of the original features is generally lost [42]. Moreover, the learning of the mixture parameters (i.e. both model selection and parameters estimation) is greatly affected by the quality of the features used as shown for instance in [43] which has given renewed attention to the feature selection problem especially in unsupervised settings. Like many other model-based feature selection approaches (see, for instance, [44]) this work has been based on the Gaussian assumption by assuming diagonal covariance matrices [44] for all clusters (i.e. all the features are assumed independent). In this thesis, and following recent approaches (see, for instance [41, 43]), we are trying to solve the feature selection problem in unsupervised learning by casting it as an estimation problem, thus avoiding any combinatorial search. For each feature, we associate a relevance weight which measures the degree of its dependence on class labels.

1.2 Contributions

The aim of this thesis is to propose several novel approaches for high-dimensional non-Gaussian data modeling and clustering. The overall contributions of this thesis are as follows

☞ **A Bayesian Approach for Infinite Generalized Gaussian Mixture Models Learning:**

We extend the finite generalized Gaussian mixture model introduced in [20] to the infinite case through a nonparametric Bayesian framework namely Dirichlet process. The infinite assumption is used to avoid problems related to model selection (i.e. determination of the number of clusters) and allows simultaneous separation of data into similar clusters and selection of relevant features.

☞ **Background Subtraction using Finite Mixtures of Asymmetric Gaussian distributions:**

We implement a method for foreground segmentation of moving regions in image sequences by using a mixture of asymmetric Gaussians to enhance the robustness and flexibility of mixture modeling, and a shadow detection scheme to remove unwanted shadows from the scene.

☞ **A Framework for Finite Asymmetric Generalized Gaussian Mixture Models learning:**

We propose the consideration of asymmetric generalized Gaussian mixture models for applications involving multidimensional non-Gaussian asymmetric data. In particular, we develop a principled learning approach to fit this kind of data. Our learning technique 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.

☞ **Simultaneous High-Dimensional Clustering and Feature Selection using Asymmetric Mixture Models:**

We propose two approaches for clustering high dimensional data using two asymmetric mixture models, namely AGM and AGGM. Furthermore, we tackle the problem of noisy and uninformative features by determining a set of relevant features for each data cluster. For model learning, the RPEM is used to allow simultaneous parameters estimation and model selection for the first approach and the expectation-maximization is used with the minimum message length criterion for the second approach.

1.3 Thesis Overview

The organization of this thesis is as follows:

❑ The first Chapter contains an introduction to mixture models.

❑ In Chapter 2, we propose a hierarchical infinite mixture model of generalized Gaussian distributions for visual learning based on non-parametric Bayesian estimation. We also introduce an unsupervised feature selection approach to determine a set of relevant features for each data cluster. Furthermore, we demonstrate the effectiveness of the proposed approach via a set of challenging applications namely image categorization, image and video segmentation, and infrared facial expression recognition. This work is published in [45].

❑ In Chapter 3, we tackle the problem of foreground segmentation of moving regions in image sequences by using a mixture of asymmetric Gaussians to enhance the robustness and flexibility of mixture modeling, and a shadow detection scheme to remove unwanted shadows from the scene. The results of comparing our method to different state of the art background subtraction methods on real image sequences of both indoor and outdoor scenes show the efficiency of our model for real-time segmentation. This work is published in [46].

- In Chapter 4, we present a highly efficient expectation-maximization (EM) algorithm, based on minimum message length (MML) formulation, for the unsupervised learning of the AGGM models parameters. Extensive experiments involving challenging applications namely pedestrian detection and Multiple Target Tracking are performed to verify the effectiveness of the proposed approach. This work is published in [47].

- In Chapter 5, we propose two unified statistical learning frameworks based on finite AGM and AGGM models. The first learning algorithm is based on the optimization of a message length objective and the second one learns the models via an RPEM technique which allows simultaneous parameters estimation and model selection. Also, for both algorithms, we tackle the problem of noisy and uninformative features by determining a set of relevant features for each data cluster. The merits of the proposed work have been shown through a complicated computer vision applications, involving high-dimensional feature vectors and large number of classes, namely scenes categorization and facial expression recognition. Part of this work is published in [48].

- In Conclusions, we summarize our contributions and present some potential future works.

Chapter 2

Generalized Gaussian mixture models as a nonparametric Bayesian approach for clustering using class-specific visual features

In this chapter, we address the problem of modeling non-Gaussian data which are largely present, and occur naturally, in several computer vision and image processing applications via the learning of a generative infinite generalized Gaussian mixture model. The proposed model, which can be viewed as a Dirichlet process mixture of generalized Gaussian distributions, takes into account the feature selection problem, also, by determining a set of relevant features for each data cluster which provides better interpretability and generalization capabilities. We propose then an efficient algorithm to learn this infinite model parameters by estimating its posterior distributions using Markov Chain Monte Carlo (MCMC) simulations. We show how the model can be used, while comparing it with other models popular in the literature, in several challenging applications involving photographic and painting images categorization, image and video segmentation, and infrared facial expression recognition.

2.1 Introduction

The problem of clustering data into homogenous groups is widely studied and has many applications in a variety of areas such as image processing, data mining, computer vision and bioinformatics [49]. Given its importance many approaches have been proposed in the past. Finite mixture models have become increasingly popular as a formal approach to clustering by assuming that the data are originated from different sources where the data arising from each particular source are modeled by a certain probability density function [14]. Such an approach to clustering raises, however, several fundamental problems: Which distribution should be considered to model the data? What order (i.e. number of clusters) should be selected? Should we consider all the features? How we should estimate the mixture parameters? The main goal of this chapter is to summarize all these challenging interrelated problems in one unified model.

One of the most fundamental and widely used statistical models is the mixture of Gaussians which is generally justified for asymptotic reasons (i.e. the sample is supposed to be sufficiently large) [16]. However, it has been observed that the Gaussian distribution is generally an inappropriate choice to model data in complex real life applications [17] and especially in the case of image processing problems where we often deal with small samples [50]. For instance, the distribution of intensity levels in natural images is well-known to be far from Gaussian [51–54]. Many studies have shown that the generalized Gaussian distribution (GGD) can be a good alternative to the Gaussian thanks to its shape flexibility which allows the modeling of a large number of non-Gaussian signals [19, 20, 55, 56]. The GGD contains the Laplacian, the Gaussian and asymptotically the uniform distribution as special cases [22] and has been used in many challenging problems (see, for instance, [21, 23, 24]). A standard method to learn finite mixture models is maximum likelihood which generally estimates the parameters through the expectation maximization (EM) framework. The EM algorithm enables us to update the mixture parameters with respect to a data set. The EM, however, is not guaranteed to lead to the best global optimal solution, depends heavily on the choice of initial parameters, and produces models that generally overfits the data which leads to suboptimal generalization performances [14, 57]. A solution to these problems can be provided by Bayesian approaches which consider the average result computed over several models by taking into account model uncertainty [58–60] and then enhances generalization performance [16, 25]. Bayesian methods have been extensively used in machine learning and signal processing because they provide a strong theoretical framework to design clustering algorithms as well as a formal approach to incorporate prior knowledge about the problem at hand (see, for instance, [20, 61–63]). A lot of research has been devoted also to the automatic selection of the number of clusters which best describe a given data set (see [14, 26, 64], for instance, and references therein).

Mixture models are parametric since a particular form has to be chosen for the components densities. At the same time, mixture models can be viewed as nonparametric, since it is possible to increase the number of components as new data arrive. The number of components can be actually supposed to increase to infinity [65]. Thus, mixtures models provide actually the best of both worlds (i.e. parametric and nonparametric approaches). In this chapter, we are interested in the nonparametric aspect of mixture models and in particular Bayesian nonparametric approaches for modeling and selection using mixture of Dirichlet processes [66] which have been shown to be a powerful alternative to determine the number of clusters [67–69]. In contrast with classic Bayesian approaches which suppose an unknown finite number of mixture components, nonparametric Bayesian approaches assume infinitely complex models (i.e. an infinite number of components) and have witnessed considerable theoretical and computational advances in recent years [36, 65, 68, 70–74]. Reviews and in-depth coverage of nonparametric Bayesian approaches can be found in [75, 76]. Thus, our approach builds on and extends our work on finite generalized Gaussian mixtures [20] to the infinite case. To our knowledge, there has been no previous consideration of nonparametric Bayesian learning for the generalized Gaussian mixture.

The majority of research in mixture models has been primarily concerned with the estimation of parameters and the selection of the number of clusters. Such an approach has several limitations because a priori all features are typically assumed to have the same weight. This is actually an important problem, since the main goal is not only the determination of clusters and their parameters but also providing the most parsimonious model to accurately describe the data which are typically highly dimensional in the number of variables. Moreover, it is noteworthy that the way employed by humans in clustering and recognition is based on formulating few selected features (i.e. humans pick up just the relevant information and ignore the irrelevant [38]) and cluster the data on the basis of these features [37]. Hence, a crucial preprocessing step is usually feature selection which generally provides more comprehensible parsimonious statistical models. Indeed, some studies have shown that two completely different patterns can be made similar by increasing the number of redundant features that encode them [77]. In conventional approaches, feature selection is treated as a separate preprocessing step. It is important to differentiate between feature selection and feature extraction. Unlike feature selection techniques, feature extraction approaches such as principal components analysis, transform the original features into a new dimension-reduced space. The main drawback of feature extraction approaches is that the physical meaning of the original features is generally lost [42]. In our case, and following recent approaches (see, for instance, [27, 41, 43, 78]), feature selection is performed simultaneously with the learning of clusters by incorporating the notion of feature relevancy into our infinite model.

The remainder of this chapter is structured as follows: First, we introduce the main formalism of our model; then we derive the posterior distributions over the model parameters and we provide a detailed description of the learning approach. Next, a simulation study is conducted to demonstrate the effectiveness of the proposed approach via a set of challenging applications. Finally, we discus the merits and demerits of our approach.

2.2 A Hierarchical Bayesian Model for Clustering and Feature Selection

We first introduce our simultaneous feature selection and clustering approach and then we show how it can be represented as a Bayesian Hierarchical model.

2.2.1 The Modeling Approach

Let $\mathcal{X} = \{\vec{X}_1, \dots, \vec{X}_N\}$ be an unlabeled data set where each vector $\vec{X}_i$ is composed of a set of continuous features representing a given object (e.g. image, video, document, etc.). It is common to assume that the data contains signals from various sources and generated from a finite mixture model:

$$p(\vec{X}_i|\Theta_M) = \sum_{j=1}^{M} p_j p(\vec{X}_i|\theta_j) \tag{2.1}$$

where M is the number of components (i.e. sources) which determines the structure of the model, $\Theta_M = (\vec{P}, \vec{\theta})$, $\vec{\theta} = (\theta_1, \dots, \theta_M)$, $\vec{P} = (p_1, \dots, p_M)$ is the vector of the components weights which are positive and sum to one, and $p(\vec{X}_i|\theta_j)$ are the components distributions which we take as multidimensional generalized Gaussians. In dimension d, by supposing that the features are conditionally independent, the generalized Gaussian density can be defined by [79]:

$$p(\vec{X}_i|\vec{\mu},\vec{\sigma},\vec{\lambda}) = \prod_{k=1}^{d} p(X_{ik}|\mu_k,\sigma_k,\lambda_k) = \prod_{k=1}^{d} \frac{\lambda_k\sqrt{\frac{\Gamma(3/\lambda_k)}{\Gamma(1/\lambda_k)}}}{2\sigma_k\Gamma(1/\lambda_k)} \exp\left(- A(\lambda_k)\left|\frac{X_{ik}-\mu_k}{\sigma_k}\right|^{\lambda_k}\right) \quad (2.2)$$

in which $A(\lambda_k) = \left(\frac{\Gamma(3/\lambda_k)}{\Gamma(1/\lambda_k)}\right)^{\lambda_k/2}$, $\Gamma(.)$ denotes the Gamma function, $\vec{\mu} = (\mu_1,\dots,\mu_d)$, $\vec{\sigma} = (\sigma_1,\dots,\sigma_d)$ and $\vec{\lambda} = (\lambda_1,\dots,\lambda_d)$. μ_k and σ_k are the pdf location and standard deviation parameters in the k^{th} dimension. The generalized Gaussian has been shown to efficiently take into account the non-Gaussian character of the natural image ensemble [80] thanks to the flexibility of its shape. The parameter λ_k controls the tails of the pdf and determines whether it is peaked or flat. Smaller values of λ_k correspond to heavy tailed distributions, when $\lambda = 2$ we have the Gaussian distribution, when $\lambda = 1$, we have the Laplacian pdf, when $\lambda >> 1$ the distribution tends to a uniform pdf, and when $\lambda < 1$ the pdf tends to be more peaked around the mean and to have heavier tails [79]. Notice that by selecting generalized Gaussians for the mixture components, the generic parameter θ_j in Eq. 2.1 becomes $(\vec{\mu}_j,\vec{\sigma}_j,\vec{\lambda}_j)$.

It is noteworthy that the model in Eq. 2.1 supposes actually that the d features have the same importance and carry pertinent information which is not generally the case, since many of which can be irrelevant for the targeted application. This is especially true in the case of image processing and computer vision applications which generally generate high-dimensional feature vectors and thus grew out the need to have efficient feature weighting and selection procedures [54, 81–84]. Examples include the challenging problems of object detection and visual scenes categorization where an important step is to determine which are the relevant features that express structure common to a given object or visual scene class [85, 86]. In fact, using all the dimensions in general will not only result in poor modeling, but also incurs excessive costs for estimating an excessive number of model parameters, some of which are potentially irrelevant [42]. It is natural, then, to assume that different features may have different weights according to each data cluster [87, 88] which can be expressed as following in the case of mixture models [78]

$$p(\vec{X}_i|\Theta) = \sum_{j=1}^{M} p_j \prod_{k=1}^{d} \left(\rho_{jk}p(X_{ik}|\theta_{jk}) + (1-\rho_{jk})p(X_{ik}|\theta_{jk}^{irr})\right) \quad (2.3)$$

where $\Theta = (\Theta_M,\vec{\rho},\vec{\theta}^{irr})$, $\vec{\theta}^{irr} = (\theta_1^{irr},\dots,\theta_M^{irr})$, $\theta_{jk} = (\mu_{jk},\sigma_{jk},\lambda_{jk})$, $\theta_{jk}^{irr} = (\mu_{jk}^{irr},\sigma_{jk}^{irr},\lambda_{jk}^{irr})$, and $\vec{\rho} = (\vec{\rho}_1,\dots,\vec{\rho}_M)$ such that $\vec{\rho}_j = (\rho_{j1},\dots,\rho_{jd})$ where each $0 \le \rho_{jk} \le 1$ represents the saliency of feature k for component j (i.e. the probability that feature k is relevant for component j). The previous model has actually a sound interpretation. Indeed, it considers that the features are not with equal importance and makes a distinction between those that are relevant and those which are irrelevant. Conceptually we assume that relevant features have been generated from $p(X_{ik}|\theta_{jk})$ and irrelevant features have been generated from another distribution $p(X_{ik}|\theta_{jk}^{irr})$ taken also as a generalized Gaussian. We finally note that the previous model is reduced to the one in Eq. 2.1 when all the feature are considered as relevant.

2.2.2 Bayesian Hierarchical Model

In the context of Bayesian inference, the most important step is the determination of the posterior which is actually proportional to the model joint distribution [16, 25] which is given by the following in our case

$$\begin{aligned} p(\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, \mathcal{X}) &= p(\vec{P})p(Z|\vec{P})p(\vec{\rho}|\vec{P}, Z)p(z|\vec{\rho}, \vec{P}, Z)p(\vec{\theta}|\vec{P}, Z, \vec{\rho}, z) \\ &\times p(\vec{\theta}^{irr}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta})p(\mathcal{X}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}) \end{aligned} \tag{2.4}$$

where $Z = (Z_1, \ldots, Z_N)$ represents the missing allocation variables and $z = (z_1, \ldots, z_N)$ are missing binary vectors to identify if a given feature is relevant or not. Each Z_i indicates from which cluster each vector $\vec{X}_i$ arose (i.e. $Z_i = j$ means that $\vec{X}_i$ comes from component j). Each $p_j = p(Z_i = j)$ represents the *a priori* probability that the vector $\vec{X}_i$ was generated by component j, and it follows from Bayes' theorem [16, 25] that $p(Z_i = j|\vec{X}_i)$, the probability that vector i is in cluster j, conditional on having observed $\vec{X}_i$ is given by

$$\begin{aligned} p(Z_i = j|\vec{X}_i) &= \frac{p_j \prod_{k=1}^{d} \left(\rho_{jk} p(X_{ik}|\theta_{jk}) + (1-\rho_{jk}) p(X_{ik}|\theta_{jk}^{irr})\right)}{\sum_{j=1}^{M} p_j \prod_{k=1}^{d} \left(\rho_{jk} p(X_{ik}|\theta_{jk}) + (1-\rho_{jk}) p(X_{ik}|\theta_{jk}^{irr})\right)} \\ &\propto p_j \prod_{k=1}^{d} \left(\rho_{jk} p(X_{ik}|\theta_{jk}) + (1-\rho_{jk}) p(X_{ik}|\theta_{jk}^{irr})\right) \end{aligned} \tag{2.5}$$

As for z, we have $z_i = (\vec{z}_{i1}, \ldots, \vec{z}_{iM})$, $\vec{z}_{ij} = (z_{ij1}, \ldots, z_{ijd})$ where each z_{ijk} indicates if feature k in vector $\vec{X}_i$ is relevant for cluster j or not (i.e. $z_{ijk} = 1$, if the feature k is relevant for cluster j and $z_{ijk} = 0$, otherwise). Each $\rho_{jk} = p(z_{ijk} = 1)$ represents the *a priori* probability that the feature k is relevant for component j, and it is straightforward to show that $p(z_{ijk} = 1, Z_i = j|\vec{X}_i)$, the probability that a feature k is relevant for cluster j, conditional on having observed $\vec{X}_i$ is given by

$$\begin{aligned} p(z_{ijk} = 1, Z_i = j|\vec{X}_i) &= \frac{\rho_{jk} p(X_{ik}|\theta_{jk})}{\rho_{jk} p(X_{ik}|\theta_{jk}) + (1-\rho_{jk}) p(X_{ik}|\theta_{jk}^{irr})} p(Z_i = j|\vec{X}_i) \\ &\propto \rho_{jk} p(X_{ik}|\theta_{jk}) p(Z_i = j|\vec{X}_i) \qquad (2.6) \end{aligned}$$

and we can deduce that

$$\begin{aligned} p(z_{ijk} = 0, Z_i = j|\vec{X}_i) &= \frac{(1-\rho_{jk}) p(X_{ik}|\theta_{jk}^{irr})}{\rho_{jk} p(X_{ik}|\theta_{jk}) + (1-\rho_{jk}) p(X_{ik}|\theta_{jk}^{irr})} p(Z_i = j|\vec{X}_i) \\ &\propto (1-\rho_{jk}) p(X_{ik}|\theta_{jk}^{irr}) p(Z_i = j|\vec{X}_i) \qquad (2.7) \end{aligned}$$

It is worth mentioning that if we condition on Z and z, the distribution of $\mathcal{X}$ is simply given by

$$\begin{aligned} p(\mathcal{X}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}) &= p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \\ &= \prod_{i=1}^{N} \prod_{k=1}^{d} \left[\left(p(X_{ik}|\theta_{Z_i k}) \right)^{z_{ik}} \left(p(X_{ik}|\theta_{Z_i k}^{irr}) \right)^{1-z_{ik}} \right] \qquad (2.8) \end{aligned}$$

We impose further common conditional independencies, so that: $p(\vec{\rho}|\vec{P}, Z) = p(\vec{\rho})$, $p(z|\vec{\rho}, \vec{P}, Z) = p(z|\vec{\rho})$, $p(\vec{\theta}|\vec{P}, Z, \vec{\rho}, z) = p(\vec{\theta})$, $p(\vec{\theta}^{irr}|\vec{P}, Z, \vec{\rho}, z, \vec{\theta}) = p(\vec{\theta}^{irr})$, $p(\vec{\theta}|Z, \vec{P}) = p(\vec{\theta})$, thus

$$p(\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, \mathcal{X}) = p(\vec{P}) p(Z|\vec{P}) p(\vec{\rho}) p(z|\vec{\rho}) p(\vec{\theta}) p(\vec{\theta}^{irr}) p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \qquad (2.9)$$

A serious practical problem now is to choose the prior distributions which describe our prior opinion about the model parameters. In our case, we suppose that $\vec{\theta}$, $\vec{\theta}^{irr}$, $\vec{\rho}$ and $\vec{P}$ follow priors depending on hyperparameters, drawn from independent hyperpriors, Λ, Λ^{irr}, δ and η, respectively. In addition, our prior distributions are that $\vec{\mu}_j$, $\vec{\sigma}_j$, $\vec{\lambda}_j$, $\vec{\mu}_j^{irr}$, $\vec{\sigma}_j^{irr}$ and $\vec{\lambda}_j^{irr}$ are all drawn independently:

$$p(\vec{\theta}|\Lambda) = \prod_{j=1}^{M} p(\vec{\sigma}_j|\Lambda_{j|\sigma}) p(\vec{\mu}_j|\Lambda_{j|\mu}) p(\vec{\lambda}_j|\Lambda_{j|\lambda}) \qquad (2.10)$$

$$p(\vec{\theta}^{irr}|\Lambda^{irr}) = \prod_{j=1}^{M} p(\vec{\sigma}_j^{irr}|\Lambda_{j|\sigma}^{irr}) p(\vec{\mu}_j^{irr}|\Lambda_{j|\mu}^{irr}) p(\vec{\lambda}_j^{irr}|\Lambda_{j|\lambda}^{irr}) \qquad (2.11)$$

where $\Lambda = (\Lambda_1, \ldots, \Lambda_M)$, $\Lambda_j = (\Lambda_{j|\sigma}, \Lambda_{j|\mu}, \Lambda_{j|\lambda})$, $\Lambda^{irr} = (\Lambda_1^{irr}, \ldots, \Lambda_M^{irr})$ and $\Lambda_j^{irr} = (\Lambda_{j|\sigma}^{irr}, \Lambda_{j|\mu}^{irr}, \Lambda_{j|\lambda}^{irr})$. To add more flexibility to the model, it is common to assume that the hyperparameters η, δ, Λ^{irr}

and Λ themselves follow distributions $p(\eta)$, $p(\delta)$, $p(\Lambda^{irr})$ and $p(\Lambda)$, respectively, that we shall develop in the next section. The joint distribution of all our model's variables is then expressed by the following factorization

$$\begin{aligned} & p(\vec{P}, Z, \vec{\rho}, z, \vec{\theta}, \vec{\theta}^{irr}, \eta, \delta, \Lambda, \Lambda^{irr}, \mathcal{X}) = p(\eta)p(\delta)p(\Lambda)p(\Lambda^{irr}) \\ \times \quad & p(\vec{P}|\eta)p(Z|\vec{P})p(\vec{\rho}|\delta)p(z|\vec{\rho})p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z)\prod_{j=1}^{M}\Big[p(\vec{\sigma}_j|\Lambda_{j|\sigma})p(\vec{\mu}_j|\Lambda_{j|\mu}) \\ \times \quad & p(\vec{\lambda}_j|\Lambda_{j|\lambda})p(\vec{\sigma}_j^{irr}|\Lambda_{j|\sigma}^{irr})p(\vec{\mu}_j^{irr}|\Lambda_{j|\mu}^{irr})p(\vec{\lambda}_j^{irr}|\Lambda_{j|\lambda}^{irr})\Big] \end{aligned} \tag{2.12}$$

2.3 Nonparametric Bayesian Learning

We first present the priors of our model. After specifying prior distributions, it is important to consider how to update these priors with information brought by the data to obtain the posterior distributions. After developing these posteriors, we describe the nonparametric approach by extending the model to the infinite case. The MCMC posterior inference and the complete learning algorithm are also given.

2.3.1 Priors and Posteriors

Conditional Posterior Distributions of $\vec{\mu}_j$ and $\vec{\mu}_j^{irr}$

We consider independent Normal priors with common hyperparameters ψ and ε^2 as the mean and variance, respectively, for the different μ_{jk} and μ_{jk}^{irr}:

$$p(\vec{\mu}_j|\psi, \varepsilon^2) = \prod_{k=1}^{d} \frac{1}{\sqrt{2\pi\varepsilon}} \exp\big(\frac{-(\mu_{jk}-\psi)^2}{2\varepsilon^2}\big) \tag{2.13}$$

And $p(\vec{\mu}_j^{irr}|\psi, \varepsilon^2)$ has the same form as $p(\vec{\mu}_j|\psi, \varepsilon^2)$. So, the generic hyperparameters $\Lambda_{j|\mu}$ and $\Lambda_{j|\mu}^{irr}$ become (ψ, ε) and according to the previous equation and our joint distribution in Eq. 2.12, the full conditional posterior distributions for $\vec{\mu}_j$ and $\vec{\mu}_j^{irr}$, giving the rest of the parameters, are:

$$p(\vec{\mu}_j|\ldots) \propto p(\vec{\mu}_j|\psi, \varepsilon^2)p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \qquad p(\vec{\mu}_j^{irr}|\ldots) \propto p(\vec{\mu}_j^{irr}|\psi, \varepsilon^2)p(\mathcal{X}|\vec{\theta}, \vec{\theta}^{irr}, Z, z) \tag{2.14}$$

The hyperparameters ψ and ε are given Normal and inverse Gamma priors, respectively:

$$p(\psi|\epsilon,\chi^2) = \frac{1}{\sqrt{2\pi}\chi}\exp\big(\frac{-(\psi-\epsilon)^2}{2\chi^2}\big) \qquad p(\varepsilon^2|\varphi,\varrho) = \frac{\varrho^{\varphi}\exp(-\varrho/\varepsilon^2)}{\Gamma(\varphi)\varepsilon^{2(\varphi+1)}} \tag{2.15}$$

Thus, according to Eqs. 2.12, 2.13 and 2.15, we obtain the following posteriors

$$p(\psi|\ldots) \propto p(\psi|\epsilon,\chi^2)\prod_{j=1}^{M} p(\vec{\mu}_j|\psi,\varepsilon^2)p(\vec{\mu}_j^{irr}|\psi,\varepsilon^2) \qquad p(\varepsilon^2|\ldots) \propto p(\varepsilon^2|\varphi,\varrho)\prod_{j=1}^{M} p(\vec{\mu}_j|\psi,\varepsilon^2)p(\vec{\mu}_j^{irr}|\psi,\varepsilon^2) \tag{2.16}$$

Conditional Posterior Distributions of $\vec{\sigma}_j$ and $\vec{\sigma}_j^{irr}$

Independent Gamma priors with common hyperpriors ι, υ, as the shape and rate parameters, are given for $\vec{\sigma}_j$ and $\vec{\sigma}_j^{irr}$

$$p(\vec{\sigma}_j|\iota,\upsilon) \sim \prod_{k=1}^{d}\frac{\sigma_{jk}^{\iota-1}\upsilon^{\iota}\exp\big(-\upsilon\sigma_{jk}\big)}{\Gamma(\iota)} \tag{2.17}$$

And $p(\vec{\sigma}_j^{irr}|\iota,\upsilon)$ has the same form as $p(\vec{\sigma}_j|\iota,\upsilon)$. So, the generic hyperparameters $\Lambda_{j|\sigma}$ and $\Lambda_{j|\sigma}^{irr}$ become (ι,υ) and according to the previous equation and our joint distribution in Eq. 2.12, the full conditional posterior distributions for $\vec{\sigma}_j$ and $\vec{\sigma}_j^{irr}$, giving the rest of the parameters, are:

$$p(\vec{\sigma}_j|\ldots) \propto p(\vec{\sigma}_j|\iota,\upsilon)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \qquad p(\vec{\sigma}_j^{irr}|\ldots) \propto p(\vec{\sigma}_j^{irr}|\iota,\upsilon)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \tag{2.18}$$

The hyperparameters ι and υ are given inverse Gamma and Gamma priors, respectively:

$$p(\iota|\vartheta,\varpi) \sim \frac{\varpi^{\vartheta}\exp(-\varpi/\iota)}{\Gamma(\vartheta)\iota^{\vartheta+1}} \qquad p(\upsilon|\tau,\omega) \sim \frac{\upsilon^{\tau-1}\omega^{\tau}\exp\big(-\omega\upsilon\big)}{\Gamma(\tau)} \tag{2.19}$$

Thus, according to Eqs. 2.12, 2.17 and 2.19, we obtain the following posteriors

$$p(\iota|\ldots) \propto p(\iota|\vartheta,\varpi)\prod_{j=1}^{M} p(\vec{\sigma}_j|\iota,\upsilon)p(\vec{\sigma}_j^{irr}|\iota,\upsilon) \qquad p(\upsilon|\ldots) \propto p(\upsilon|\tau,\omega)\prod_{j=1}^{M} p(\vec{\sigma}_j|\iota,\upsilon)p(\vec{\sigma}_j^{irr}|\iota,\upsilon) \tag{2.20}$$

Conditional Posterior Distributions of $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$

For the parameters $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$ we placed independent Gamma priors with common hyperparameters κ and ς:

$$p(\vec{\lambda}_j|\kappa,\varsigma) = \prod_{k=1}^{d}\frac{\lambda_{jk}^{\kappa-1}\varsigma^{\kappa}\exp\big(-\varsigma\lambda_{jk}\big)}{\Gamma(\kappa)} \tag{2.21}$$

And $p(\vec{\lambda}_j^{irr}|\kappa,\varsigma)$ has the same form as $p(\vec{\lambda}_j|\kappa,\varsigma)$. So, the generic hyperparameters $\Lambda_{j|\lambda}$ and $\Lambda_{j|\lambda}^{irr}$ become (κ,ς) and according to the previous equation and our joint distribution in Eq. 2.12, the full conditional posterior distributions for $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$, giving the rest of the parameters, are:

$$p(\vec{\lambda}_j|\ldots) \propto p(\vec{\lambda}_j|\kappa,\varsigma)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \qquad p(\vec{\lambda}_j^{irr}|\ldots) \propto p(\vec{\lambda}_j^{irr}|\kappa,\varsigma)p(\mathcal{X}|\vec{\theta},\vec{\theta}^{irr},Z,z) \tag{2.22}$$

The hyperparameters κ and ς are given inverse Gamma and Gamma priors, respectively:

$$p(\kappa|\alpha,\phi) \sim \frac{\phi^{\alpha}\exp(-\phi/\kappa)}{\Gamma(\alpha)\kappa^{\alpha+1}} \qquad p(\varsigma|\nu,\beta) \sim \frac{\varsigma^{\nu-1}\beta^{\nu}\exp\left(-\beta\varsigma\right)}{\Gamma(\nu)} \tag{2.23}$$

Thus, according to Eqs. 2.12, 2.21 and 2.23, we obtain the following posteriors

$$p(\kappa|\ldots) \propto p(\kappa|\alpha,\phi)\prod_{j=1}^{M} p(\vec{\lambda}_j|\kappa,\varsigma)p(\vec{\lambda}_j^{irr}|\kappa,\varsigma) \qquad p(\varsigma|\ldots) \propto p(\varsigma|\nu,\beta)\prod_{j=1}^{M} p(\vec{\lambda}_j|\kappa,\varsigma)p(\vec{\lambda}_j^{irr}|\kappa,\varsigma) \tag{2.24}$$

Conditional Posterior Distribution of $\vec{\rho}$

We know that each ρ_{jk} is defined in the compact support [0,1], thus we consider for it a Beta distribution, with parameters δ_1 and δ_2 common to all classes and all dimensions, as a prior, which give us

$$p(\vec{\rho}|\delta) = \left[\frac{\Gamma(\delta_1+\delta_2)}{\Gamma(\delta_1)\Gamma(\delta_2)}\right]^{Md}\prod_{j=1}^{M}\prod_{k=1}^{d}\rho_{jk}^{\delta_1-1}(1-\rho_{jk})^{\delta_2-1} \tag{2.25}$$

So, the generic hyperparameter δ become (δ_1,δ_2). Recall that $\rho_{jk} = p(z_{jk}=1)$ and $1-\rho_{jk} = p(z_{jk}=0)$, $k=1,\ldots,d$, $j=1,\ldots,M$ thus each z_{jk} follows a d-variate Bernoulli distribution and we have

$$p(z|\vec{\rho}) = \prod_{i=1}^{N}\prod_{j=1}^{M}\prod_{k=1}^{d}\rho_{jk}^{z_{ijk}}(1-\rho_{jk})^{1-z_{ijk}} = \prod_{j=1}^{M}\prod_{k=1}^{d}\rho_{jk}^{f_{jk}}(1-\rho_{jk})^{N-f_{jk}} \tag{2.26}$$

where $f_{jk} = \sum_{i=1}^{N}\mathbb{I}_{z_{ijk}=1}$. Then, according to Eqs. 2.12, 2.25 and 2.26, we have

$$p(\vec{\rho}|\ldots) \propto p(\vec{\rho}|\delta)p(z|\vec{\rho}) \propto \prod_{j=1}^{M}\prod_{k=1}^{d}\rho_{jk}^{f_{jk}+\delta_1-1}(1-\rho_{jk})^{N-f_{jk}+\delta_2-1} \tag{2.27}$$

The hyperparameters δ_1 and δ_2 are given Gamma priors with common hyperparameters $(\varphi_\delta,\varrho_\delta)$ which give us the following posteriors

$$p(\delta_1|\ldots) \propto p(\delta_1|\varphi_\delta,\varrho_\delta)p(\vec{\rho}|\delta) \qquad p(\delta_2|\ldots) \propto p(\delta_2|\varphi_\delta,\varrho_\delta)p(\vec{\rho}|\delta) \tag{2.28}$$

2.3.2 The Infinite Model

An important issue now is the determination of the number of clusters which has been widely studied from both Bayesian and deterministic perspectives by supposing that the number of components is bounded (see, for instance, [14, 26]). An alternative approach that has attracted a lot of attention recently is to define mixture of distributions with a countably infinite number of components [65]. The attractive features of nonparametric Bayesian approaches, which can incorporate infinitely many parameters, have been widely exploited and are well documented and will not be repeated here (see, for instance, [36, 65, 68, 76]). In the following, we explain the main idea behind this approach in the case of mixture models.

According to Eq. 2.12, the only terms that involve $\vec{P}$ whose dimensionality is M are $p(Z|\vec{P})$ and $p(\vec{P}|\eta)$. Recall that $p_j = p(Z_i = j), j = 1, \ldots, M$, thus

$$p(Z|\vec{P}) = \prod_{j=1}^{M} p_j^{n_j} \tag{2.29}$$

where $n_j = \sum_{i=1}^{N} \mathbb{I}_{Z_i=j}$ is the number of vector in cluster j. The distribution $p(\vec{P}|\eta)$ is taken as a symmetric Dirichlet with parameters $\frac{\eta}{M}$. Because the Dirichlet is a conjugate prior to the multinomial, we can marginalize out $\vec{P}$ from Eq. 2.12:

$$\begin{aligned} p(Z|\eta) &= \int_{\vec{P}} p(Z|\vec{P})p(\vec{P}|\eta)d\vec{P} = \frac{\Gamma(\eta)}{\prod_{j=1}^{M}\Gamma(\frac{\eta}{M})} \int_{\vec{P}} \prod_{j=1}^{M} p_j^{n_j+\frac{\eta}{M}-1} d\vec{P} \\ &= \frac{\Gamma(\eta)}{\Gamma(\eta+N)} \prod_{j=1}^{M} \frac{\Gamma(\frac{\eta}{M}+n_j)}{\Gamma(\frac{\eta}{M})} \end{aligned} \tag{2.30}$$

which can be considered as a prior on Z. We have also

$$p(\vec{P}|Z,\eta) = \frac{p(Z|\vec{P})p(\vec{P}|\eta)}{p(Z|\eta)} = \frac{\Gamma(\eta+N)}{\prod_{j=1}^{M}\Gamma(\frac{\eta}{M}+n_j)} \prod_{j=1}^{M} p_j^{n_j+\frac{\eta}{M}-1} \tag{2.31}$$

which is a Dirichlet distribution with parameters $(n_1 + \frac{\eta}{M}, \ldots, n_M + \frac{\eta}{M})$ from which we can show that:

$$p(Z_i = j|\eta, Z_{-i}) = \frac{n_{-i,j} + \frac{\eta}{M}}{N - 1 + \eta} \tag{2.32}$$

where $Z_{-i} = \{Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_N\}$, $n_{-i,j}$ is the number of vectors, excluding $\vec{X}_i$, in cluster j. The main idea behind countably infinite mixture models relies on observing that by taking the limit of $p(Z_i = j|\eta, Z_{-i})$ as $M \to \infty$ gives us [65, 68]

$$p(Z_i = j|\eta, Z_{-i}) = \begin{cases} \frac{n_{-i,j}}{N-1+\eta} & \text{if } n_{-i,j} > 0 \text{ (cluster } j \text{ is represented)} \\ \frac{\eta}{N-1+\eta} & \text{if } n_{-i,j} = 0 \text{ (cluster } j \text{ is not represented)} \end{cases} \tag{2.33}$$

Thus, a vector $\vec{X}_i$ is allocated to an existing (i.e. represented) cluster with a certain probability proportional to the number of vectors already assigned to this cluster and it is affected to a new (i.e. not represented) cluster with probability proportional to the hyperparameter η. It is noteworthy that infinite mixture models takes implicitly into account the notion of online learning and then the fact that features relevancy may change as new data arrive which is crucial in several machine vision applications for instance [89]. Having the conditional priors in Eq. 2.33, the conditional posteriors are obtained by combining these priors with the likelihood of the data [65, 68]

$$\begin{aligned} &p(Z_i = j| \dots) \\ &= \begin{cases} \frac{n_{-i,j}}{N-1+\eta} \prod_{k=1}^{d} \left(\rho_{jk} p(X_{ik}|\theta_{jk}) + (1-\rho_{jk}) p(X_{ik}|\theta_{jk}^{irr})\right) & \text{if } j \text{ is represented} \\ \int \frac{\eta p(\vec{X}_i|\vec{\theta}_j, \vec{\theta}_j^{irr}, z_i) p(\theta_j|\Lambda_j) p(\theta_j^{irr}|\Lambda_j^{irr}) p(\vec{\rho}_j|\delta)}{N-1+\eta} d\theta_j d\theta_j^{irr} d\vec{\rho}_j & \text{if } j \text{ is not represented} \end{cases} \end{aligned} \tag{2.34}$$

Concerning the hyperparameters η[1], we have chosen an inverse gamma prior with parameters (χ_η, κ_η) for it:

$$p(\eta|\chi_\eta, \kappa_\eta) \sim \frac{{\kappa_\eta}^{\chi_\eta} \exp(-\kappa_\eta/\eta)}{\Gamma(\chi_\eta)\eta^{\chi_\eta+1}} \tag{2.35}$$

which gives with Eq. 2.33 the following posterior (for more details, see [65])

$$p(\eta| \dots) \propto \frac{{\kappa_\eta}^{\chi_\eta} \exp(-\kappa_\eta/\eta)}{\Gamma(\chi_\eta)\eta^{\chi_\eta+1}} \frac{\eta^M \Gamma(\eta)}{\Gamma(N+\eta)} \tag{2.36}$$

2.3.3 Complete Algorithm

Several Monte carlo methods for sampling mixture posteriors have been developed in the past [16, 25]. The most widely applied approach is the Gibbs sampling (see [91], for instance, for interesting discussions) that we will use to sample from the obtained model posteriors as follows:

- Generate Z_i from Eq. 2.34 and then update n_j, $j = 1, \dots, M$, $i = 1, \dots, N$.
- Update the number of represented components M.

[1] This parameter plays an important role in controlling the weights of the mixture components and then the number of clusters. Indeed, it is possible to show that the number of clusters increase at a rate proportional to $\eta \log N$ and then it is crucial to suppose that it is unknown and then follows a prior distribution [90].

- $p_j = \frac{n_j}{N+\eta}$, $j = 1, \ldots, M$ and the mixing parameters of unrepresented components are given by $p_U = \frac{\eta}{\eta+N}$.
- Generate the $\vec{z}_{ij}$ from a d-variate Bernoulli distribution with parameters $p(z_{ijk} = 1, Z_i = j|\vec{X}_i)$.
- Generate ρ_k from Eq. 2.27, $k = 1, \ldots, d$.
- Generate $\vec{\mu}_j$ and $\vec{\mu}_j^{irr}$ from the posteriors in Eq. 2.14, $\vec{\sigma}_j$ and $\vec{\sigma}_j^{irr}$ from the posteriors in Eq. 2.18, $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$ from the posteriors in Eq. 2.22, $j = 1, \ldots, M$.
- Update the hyperparameters: Generate ψ, ε^2, ι, υ, κ, ς, δ_1, δ_2 and η according to the posteriors in Eqs. 2.16, 2.20, 2.24, 2.28 and 2.36, respectively.

Note that, for the initialization step we start by assuming that all the vectors are in the same cluster and that all the features are relevant, and we generate the parameters by sampling from their prior distributions. It is noteworthy also that although an infinite model appears complex because of the number of involved parameters, it allows actually straightforward posterior inference with MCMC simulation as it is clear from the previous algorithm. The above algorithm can be viewed actually as a self-refinement process that starts with an initial set of data and feature relevancy. From this initial state, the process strives to find features that are discriminative for each cluster, and then refine the clusters by determining the cluster label of each vector using these relevant features. Via this self-refinement process, the accuracy of the whole data representation is gradually improved. All the conditional posterior distributions are straightforward to sample from (especially the posteriors of ρ_d and $\vec{z}_{ij}$ which have known forms). Indeed, sampling from Eqs. 2.16, 2.20, 2.24, 2.28 and 2.36 is based on adaptive rejection sampling (ARS) [92]. The sampling of the vectors Z_i (Eq. 2.34) is based on an approach, originally proposed in [68]. For simulations from the posteriors of $\vec{\mu}_j$, $\vec{\mu}_j^{irr}$, $\vec{\sigma}_j$, $\vec{\sigma}_j^{irr}$, $\vec{\lambda}_j$ and $\vec{\lambda}_j^{irr}$, we apply the well known random walk Metropolis-Hastings (M-H) algorithm (i.e. log-normal proposals with scale ζ^2). An important problem when using MCMC techniques is the convergence assessment which has been the topic of extensive rigorous studies in the past (see, for instance, [93–95]). Several systematic approaches for establishing convergence of MCMC have been proposed and one of them that we follow is the diagnostic approach proposed by Raftery and Lewis [96, 97], that has been shown to often work well in practice. This approach is based on a single long-run of the Gibbs sampler.

2.4 Experimental Results

Performance evaluation for our model is conducted using a set of challenging experiments involving distinguishing paintings from photographs, image and video segmentation, and infrared facial expression recognition. The main goal of these experiments is to compare our infinite model (IGGM+FS) with other models that have been used in the literature namely finite Gaussian mixture (GM) [28], finite Gaussian mixture with feature selection (GM+FS) [43], finite generalized Gaussian (GGM) [21], finite generalized Gaussian with feature selection (GGM+Fs) [27], infinite Gaussian mixture (IGM) [65], infinite Gaussian mixture with feature selection (IGM+FS) and infinite generalized Gaussian (IGGM) [74]. In these applications our specific choice for the hyperparameters is $\varphi_\delta = 2, \varrho_\delta = 0.5, \chi_\eta = 2, \kappa_\eta = 1, \epsilon = 0, \chi^2 = 1, \varphi = 2, \varrho = 0.5, \vartheta = 2, \varpi = 1, \tau = 2, \omega = 0.5, \alpha = 2, \phi = 1, \nu = 0.5$ and $\beta = 2$. In order to conduct a sensitivity test we have used different values of hyperparameters in the following intervals :$\varphi_\delta \in [1.8, 2.2], \varrho_\delta \in [0.3, 0.7], \chi_\eta \in [1.8, 2.2], \kappa_\eta \in [0.8, 1.2], \epsilon \in [0, 0.4], \chi^2 \in [0.8, 1.2], \varphi \in [1.8, 2.2], \varrho \in [0.3, 0.7], \vartheta \in [1.8, 2.2], \varpi \in [0.8, 1.2], \tau \in [1.8, 2.2], \omega \in [0.3, 0.7], \alpha \in [1.8, 2.2], \phi \in [0.8, 1.2], \nu \in [0.3, 0.7]$ and $\beta \in [1.8, 2.2]$. Having the outputs of our algorithm when changing the hyperparameters in hand we applied a student t test to determine the robustness of the posteriors results with respect to our hyperparameters.

2.4.1 Distinguishing Paintings from Photographs

Image Description

Distinguishing paintings from real photographs is an important and challenging (even for a human observer) problem in several applications such as content-based image retrieval, web site filtering (e.g. distinguishing pornographic images from nude paintings) [98, 99], categorization [100] and content-based access to art paintings [101]. However, very few works have been proposed in the past [98, 99, 101, 102] as compared, for instance, to the problem of distinguishing photographs and computer-generated graphics [103–105]. In particular, the authors in [98, 99] found that an important step is the extraction of the right visual features derived from the edge, color and gray-scale-texture information. In particular, the following distinguishing features have been derived and found efficient. Four scalar-valued, called visual features, were defined namely color edges vs. intensity edges (E_g), spatial variation of color (R), number of unique colors (U) and pixel saturation (S). Pixel distribution in RGBXY space ($\vec{s}$) and gray-scale-texture have been considered, also.

Color edges vs. intensity edges feature is defined as $E_g = \frac{\text{\#pixels: intensity, not color edge}}{\text{total number of edge pixels}}$ [98, 99]. The spatial variation of color, R, is defined as the average over all image pixels of the sums of the areas of the facets of the pyramids determined by three normals at each pixel. These normals are obtained by determining at each pixel the orientation of the plane that best fits a 5×5 neighborhood centered on that pixel in the RGB domain. The number of unique colors, U, is defined as the number of unique colors of an image normalized by the total number of pixels. The pixel saturation, S, is defined as the ratio between the count in the highest bin (20) and the lowest (1) of the mean saturation histogram derived from the image represented in the HSV color space. The pixel distribution in RGBXY space represents an image by a five dimensional vectors $\vec{s}$ of the singular values of its RGBXY pixel covariance matrix (i.e. the image representation in the RGB color space enhanced by adding the two spatial coordinates, x and y to the RGB vector of each pixel). Finally, the gray-scale-texture feature is a description of the image by a feature vector of 32 dimensions which represent the mean and standard deviation of the Gabor responses across image locations for filtered images obtained by considering four different scales and four orientations (0, 90, 45, 135 degrees). Using all these features images can be represented by 41-dimensional vectors which can be used for the categorization task (i.e. paintings vs. photographs).

Results

The performance of our infinite mixture model was evaluated on the data set considered in [98, 99] which contains 6000 photographs, with 568 × 506 pixels mean size and standard deviation equal to 144 × 92 pixels, and 6000 paintings with mean size and standard deviation equal to 534 × 497 and 171 × 143 pixels, respectively. In this data set, the painting class includes conventional canvas paintings, murals and frescoes, but excludes line drawings and computed-generated images. On the other hand, the class photographs includes exclusively three-dimensional real-world scenes color images. Figure 2.1 shows examples of images from both classes. From these images, and like [98, 99], we have generated 36 training sets where each set consists of 1000 paintings and 1000 photographs and the corresponding testing sets consist of the remaining images (i.e. 5000 paintings and 5000 photographs). Having the training data in hand, we apply our algorithm, presented in

Figure 2.1: Sample images from each group. Row 1: Paintings, Row 2: Photographs.

Section 2.3.3, to the training vectors in both classes. After this stage, each class in the database is represented by an infinite generalized Gaussian mixture. Finally, in the classification stage each unknown image is assigned to the class increasing more its loglikelihood. Table 2.1 summarizes the classification results when considering different learning approaches and scenarios. According to this table, we can see clearly that both considered infinite models (IGM and IGGM) outperform the classification approach used in [98, 99] and based on neural networks. Moreover, the results are improved further when feature selection is considered. It is noteworthy that we have applied a sensitivity test using different values of the hyperparameters with the student t test and found that the difference was not statistically significant (The minimum P-value for photographs and paintings are 0.527497 and 0.515763 respectively)

2.4.2 Image and Video Segmentation

Image and video segmentation is one of the major and basic steps in digital multimedia processing which has the objective of extracting information from an image or a sequence of images (video). It consists in partitioning the given image (or video) into homogeneous spatial (spatiotemporal in the case of videos) regions enjoying similar properties such as texture, color, boundary, and intensity. It is via segmentation that regions of interest are extracted for subsequent processing such as object detection and recognition and for further applications such as content-based image retrieval. Various methods have been proposed in the literature and tremendous advancements have been made within the past decade (see, for instance, [106, 107]). Contributions, however, continue to be made in the formulation of new mathematical and statistical approaches and in the

Table 2.1: Average classification accuracies (%) (± standard deviation) obtained using different approaches for distinguishing paintings from photographs. IGM: infinite Gaussian mixture, IGM + FS: infinite Gaussian mixture with feature selection, IGGM: infinite generalized Gaussian mixture, IGGM + FS: infinite generalized Gaussian mixture with feature selection.

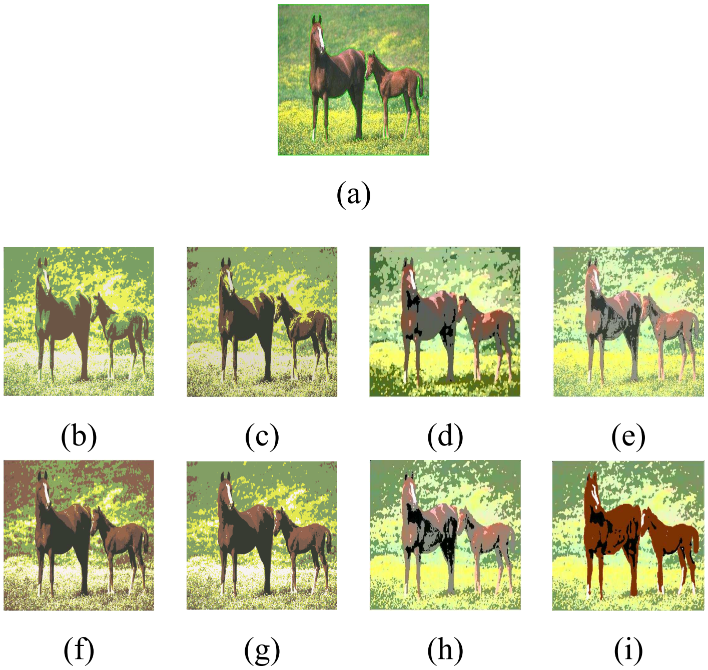
Approach	Photographs	Paintings
[98, 99] using $\{E_g, U, R, S\}$	71.00 (±4.00)	72.00 (±5.00)
[98, 99] using RGBXY space	81.00 (±3.00)	81.00 (±3.00)
[98, 99] using gray-scale-texture feature	78.00 (±4.00)	79.00 (±4.00)
[98, 99] using all features	92.00 (±2.00)	94.00 (±3.00)
IGM using $\{E_g, U, R, S\}$	69.00 (±5.50)	70.50 (±6.25)
IGM + FS using $\{E_g, U, R, S\}$	73.00 (±4.50)	74.25 (±6.00)
IGM using RGBXY space	80.75 (±4.25)	80.50 (±4.00)
IGM + FS using RGBXY space	83.00 (±3.75)	83.75 (±3.25)
IGM using gray-scale-texture feature	76.25 (±3.75)	77.00 (±3.25)
IGM + FS using gray-scale-texture feature	82.75 (±3.25)	82.00 (±3.50)
IGM using all features	89.00 (±4.75)	90.00 (±5.25)
IGM + FS using all features	93.75 (±3.25)	93.50 (±3.00)
IGGM using $\{E_g, U, R, S\}$	72.50 (±3.50)	73.50 (±4.00)
IGGM + FS using $\{E_g, U, R, S\}$	77.50 (±2.50)	78.50 (±3.00)
IGGM using RGBXY space	82.75 (±3.50)	81.50 (±3.50)
IGGM + FS using RGBXY space	87.25 (±2.25)	86.75 (±2.75)
IGGM using gray-scale-texture feature	80.25 (±2.75)	80.00 (±3.25)
IGGM + FS using gray-scale-texture feature	84.25 (±2.75)	84.75 (±1.75)
IGGM using all features	92.00 (±3.75)	93.00 (±3.25)
IGGM + FS using all features	97.25 (±1.50)	96.75 (±1.75)

developments of new algorithms. The discussion of all these previous approaches is clearly beyond the scope of this chapter. As a formal well-established approach to clustering, finite mixture models have been widely used for image segmentation. In particular finite generalized Gaussian mixture models have provided excellent segmentation results [20, 21]. The effectiveness of the segmentation to yield meaningful regions depends greatly on the choice of the number of clusters. This problem has been tackled in [21] and [20] using minimum message length (MML) and Bayes factor criteria. Here we propose the application of our infinite model in order to reduce further over/under-segmentation problems. Our model takes also into account the fact that, usually, in video/image segmentation, some features are noisy, redundant, or irrelevant for the segmentation. The presence of these irrelevant features introduces a bias to distances between objects which may effect the homogeneity of regions as discussed in some recent works that have shown that feature weighting and selection generally improve segmentation results [27, 108]. Color and texture are important segmentation cues [109, 110], thus we have used for each pixel a 27 features vector that combines both information. For color information, we have chosen the RGB color space, as for texture information 24 features calculated from the color correlogram of the pixel neighborhood, as defined in [111], have been considered.

The images used in our experiments are from the Berkeley benchmark [112]. We have chosen this dataset because a ground truth (GT) (i.e., segmentation performed manually) is provided for each image in the dataset. We have compared the segmentation results obtained using our proposed approach (IGGM+FS) with those obtained using: 1) The Gaussian mixture with MML and without feature selection (GM); 2) The infinite Gaussian mixture without feature selection (IGM); 3) The generalized Gaussian mixture with MML and without FS (GGM); 4) The infinite generalized Gaussian mixture without feature selection (IGGM); 5) The Gaussian mixture with MML and feature selection (GM+FS); 6) The infinite Gaussian mixture with feature selection (IGM+FS); 7) The generalized Gaussian mixture with MML and with FS (GGM+FS). In order to have a quantitative evaluation of the performance, we have used two objective criteria namely Boundary localization error (E_1) and the over/under-segmentation error (E_2). E_1 measures the misalignment of regions between a tested segmentation (TS) and the GT and is defined as [112]:

$$E_1 = \frac{1}{N} \sum_{(u,v)} \min\{E_{(u,v)}(TS, GT), E_{(u,v)}(GT, TS)\} \tag{2.37}$$

where $E_{(u,v)}(TS, GT) = \frac{|S_{TS} - S_{GT}|}{|S_{TS}|}$ and $E_{(u,v)}(GT, TS) = \frac{|S_{GT} - S_{TS}|}{|S_{GT}|}$. $S_{TS}(u, v)$ and $S_{GT}(u, v)$ are the segments (where the segment is defined as a connected set of 5 pixels or more) containing the pixel (u, v) in the TS and the GT, respectively. Note that the symbol (–) means the set difference operator and N is the number of pixels in the image. E_2 measures the amount of over/under-segmentation produced by each TS when compared to the GT. E_2 is defined as the sum of the number of segments in the GT that are over-segmented in the TS, and the number of segments in the TS that are over-segmented in the GT as suggested in [111].


(a)

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

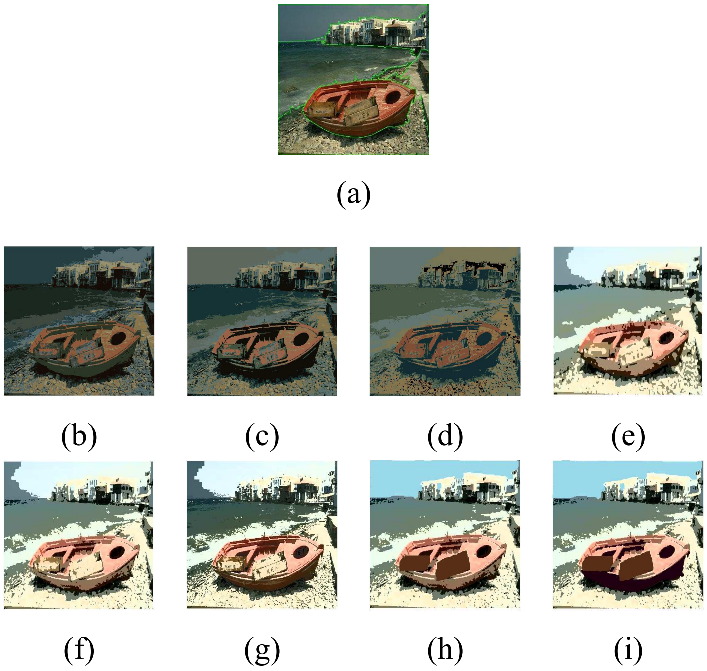
(f) (g) (h) (i)

Figure 2.2: Segmentation results for the first image from the Berkeley dataset. (a) GT, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

Figures 2.2, 2.3, and 2.4 show the segmentation results for 3 different images from the Berkeley dataset when applying the 8 different methods. Table 2.2 shows the different values of E_1 and E_2 for each model when considering the whole dataset. We can conclude that the IGGM+FS outperformed all other methods in both performance criteria. This can be explained by the fact that the generalized Gaussian is more flexible and by the fact that adopting Bayesian approach allows to account for the effect of uncertainty in the modeling parameters on the subsequent segmentation. We can see clearly also that using feature selection in both generalized Gaussian and Gaussian mixture models yields better performance than without using feature selection which is actually expected and meets the conclusions reached in some previous works [27].

Table 2.2: Errors (E_1 and E_2) calculation for the Berkeley dataset.

Errors (E_1; E_2)							
GM	IGM	GGM	IGGM	GM+FS	IGM+FS	GGM+FS	IGGM+FS
(0.21; 23)	(0.21; 21)	(0.19; 20)	(0.17; 18)	(0.14; 15)	(0.13; 13)	(0.09; 11)	(0.08; 9)



(a)

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

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

Figure 2.3: Segmentation results for the second image from the Berkeley dataset. (a) GT, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

In the case of videos, the segmentation problem is more challenging and depends on different factors such as lighting conditions, partial occlusion, rotation in depth and scale changes [113]. Moreover, the segmentation model has to be adapted in time to take into account the dynamical nature of the video scenes. Indeed, the number of regions, the saliency of the used features and the segmentation model's parameters can change from one frame to another. The majority of the previous works that have used mixture models for video segmentation assume a fixed number of components and just update the component's parameters. Here we use our infinite model, which takes implicitly into account the updating problem, by considering the same visual feature described in the image segmentation part and by adopting the formulation proposed in [27]. We have investigated our model via two widely used videos (Akiyo and Suzie). In order to demonstrate the robustness of our method we have used the two objective criteria (E_1 and E_2) introduced above. Figures 2.5 and 2.6 show two examples of video segmentation using different tested approaches. From each video, we show a frame drawn randomly from the sequence. Table 2.3 shows the segmentation results in terms of E_1 and E_2. We can see clearly the improvement brought by the

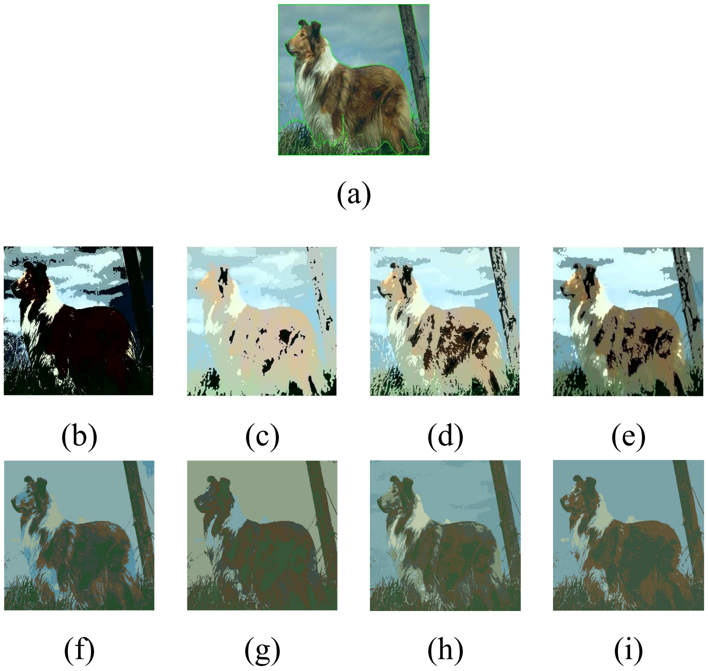
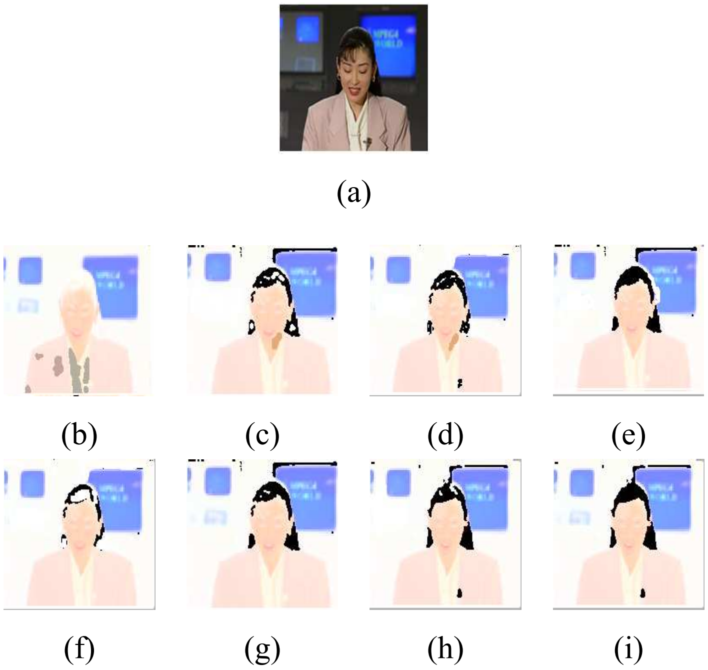


Figure 2.4: Segmentation results for the third image from the Berkeley dataset. (a) GT, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

proposed model against the compared ones. These results are confirmed visually in the segmentations shown in figures 2.5 and 2.6, where the quality of object segmentation is clearly improved using the proposed approach.

2.4.3 Infrared Facial Expression Recognition

Face expression analysis and recognition has been one of the fastest growing areas of computer vision over the last few years [114, 115], due primarily to the rapidly increasing demand for emotion analysis, biometrics, and image retrieval to ensure security and safety. Face expression recognition (FER) is interested in applying machine vision and pattern recognition algorithms on both still images and/or image video sequences in order to extract emotional content from visual patterns of a person's face. Most of FER systems rely on videos as their input [116, 117], however, video sequences are not always available in every real world situation. Thus, different FER image based approaches have been developed [116, 118] in order to offer a reliable alternative. FER process



(a)

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

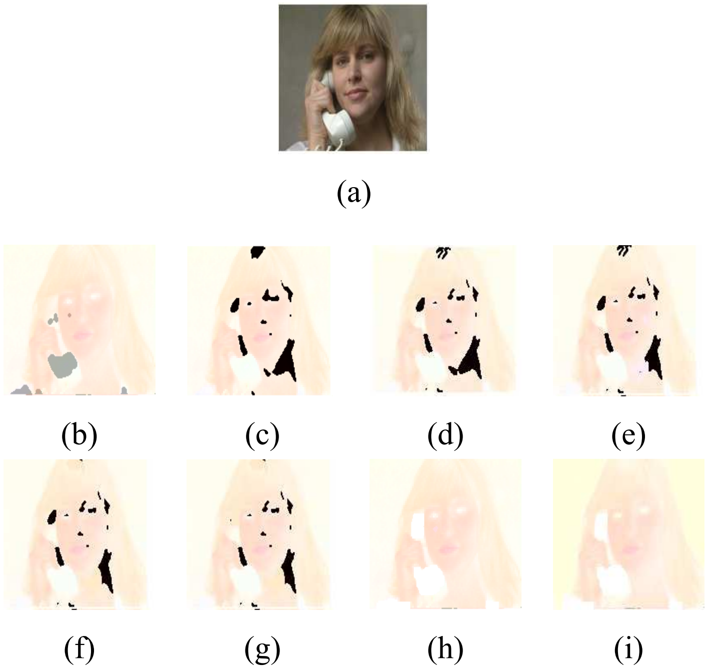
(f) (g) (h) (i)

Figure 2.5: Sample image from Akiyo video. (a) Sample frame, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

can be divided into three main tasks: region of interest selection, feature extraction, and image classification. Region of interest (ROI) selection is used to identify areas where feature extraction will take place (e.g. the entire face [118], eyes, and mouth [119]). In order to decrease the dimensionality of different ROI, they are usually represented in terms of low-level feature vectors in lower dimensional feature space [118, 119]. Image classification task identifies the emotional state of the input person face by searching a database of known different emotional expressions.

Facial analysis systems relying on visual spectrum have received relatively more attention compared to the thermal infrared one. This was justified by both the higher cost of thermal sensors, the lack of widely available IR image databases and the quality of the produced images (lower resolution and higher image noise). Recently, however, thermal imagery of human faces has been established as a valid biometric signature and several approaches have been proposed thanks to the advances of infrared imaging technology [120, 121].

In this section, the aim is to implement an infrared multi-class face expression recognition algorithm based on our infinite model (IGGM+FS). The proposed FER system can be divided into



(a)

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

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

Figure 2.6: Sample image from Suzie video. (a) Sample frame, (b) GM, (c) IGM, (d) GGM, (e) IGGM, (f) GM+FS, (g) IGM+FS, (h) GGM+FS, (i) IGGM+FS.

three steps: face localization, facial feature estimation, feature selection and emotions identification. Normally, the position of the face is not centered within the image and can change greatly. Figure 2.7 shows examples of faces with different expressions taken from different poses. Face localization is used to identify the approximate position of each subject's face within the image. Infrared face localization is based on the idea that higher image intensities correspond to region with higher temperature which correspond to the face in our case. First, we have used a thresholding operation over the entire image which makes facial pixel intensities more prominent. Now for the n pixels remaining (with values over the threshold value) taken as position vectors p = (p_x,p_y) we can compute the geometric centroid μ=(μ_x, μ_y) and μ_j=$\frac{\sum_{i=1}^{n}\mu_{j,i}}{n}$. In order to calculate μ_x we look for the facial pixels with the lowest thermal content along μ_y which will be centered at the person nose [121]. Figure 2.8 shows how by applying thresholding on the thermal image of a person's face we can easily locate the center even under different poses. Most of the research done until now considers that emotional information is centered around the eyes and mouth areas on the face. So in order to localize our facial features we have chosen an area of 120×130 centered

Table 2.3: Errors (E_1 and E_2) calculation for the 2 tested videos when using our infinite model (IGGM+FS), finite Gaussian mixture (GM), finite Gaussian mixture with feature selection (GM+FS), finite generalized Gaussian (GGM), finite generalized Gaussian with feature selection (GGM+FS), infinite Gaussian mixture (IGM), infinite Gaussian mixture with feature selection (IGM+FS) and infinite generalized Gaussian (IGM).

Errors (E_1; E_2)		
Video	Akiyo	Suzie
Size	300 frames	150 frames
GM	(0.23; 22.50)	(0.25; 27.40)
IGM	(0.22; 21.70)	(0.24; 24.90)
GGM	(0.22; 20.50)	(0.24; 23.70)
IGGM	(0.20; 19.40)	(0.21; 20.40)
GM+FS	(0.16; 17.80)	(0.18; 19.10)
IGM+FS	(0.15; 16.20)	(0.16; 15.60)
GGM+FS	(0.12; 13.20)	(0.12; 11.30)
IGGM+FS	(0.11; 11.70)	(0.11; 10.80)

around the image centroid μ which should be appropriate to our dataset as argued in [122]. In order to estimate the important features in this area we used the method of [123] that allows us to get 75 key points. Figure 2.8 shows an image from our dataset with the 75 interest points detected on the person's face. After detecting interest points we have applied the K mean approach as implemented in [122] in order to identify eyes and mouth areas on the face. From figure 2.8 we can see clearly that this method was able to identify these areas effectively. The texture information in these area has been then represented using the texture descriptors proposed and used in [124].

In our experiments, we have performed face recognition using images from the Iris thermal face which is a subset of the Object Tracking and Classification Beyond the Visible Spectrum (OTCBVS) database. Images are gray-scale infrared of 320×240 each and represent different persons under different expressions and poses. We used 756 images for 28 different persons with three different expressions: surprise, happy, and angry. Figure 2.7 shows images from different classes (with different emotions). We have used 9 images for each person as training set and the rest as testing set. This gave us 252 and 504 images for training and testing, respectively. We have also applied the 7 other methods introduced above. Tables 2.4.a, 2.4.b, 2.4.c, 2.4.d, 2.4.e, 2.4.f, 2.4.g, and 2.4.h are the corresponding confusion matrices.

In order to evaluate the quality of clustering we have used two different criteria: accuracy and normalized mutual information. Accuracy is a simple and transparent evaluation measure that computes the percentage of images correctly clustered to the total number of images. Normalized mutual information (NMI) [125] is a novel criterion used for classifier evaluation based on information theory that is always between 0 and 1 (The larger this value, the better the clustering performance):

$$NMI(\Omega, \mathtt{C}) = \frac{I(\Omega, \mathtt{C})}{[H(\Omega) + H(\mathtt{C})]/2} \tag{2.38}$$

where $\mathtt{C} = (\mathtt{C}_1, \ldots, \mathtt{C}_M)$ are the classes that represent the data (in this application M=3) and $\Omega = (\Omega_1, \ldots, \Omega_K)$ are the clusters identified by the classification algorithm. I is the mutual information given by:

$$I(\Omega, \mathtt{C}) = \sum_{m=1}^{M} \sum_{k=1}^{K} P(\Omega_k \cap \mathtt{C}_m) \log \frac{P(\Omega_k \cap \mathtt{C}_m)}{P(\Omega_k) P(\mathtt{C}_m)} \tag{2.39}$$

where $P(\Omega_k)$, $P(\mathtt{C}_m)$, and $P(\Omega_k \cap \mathtt{C}_m)$ are the probabilities of an image being in cluster Ω_k, class $\mathtt{C}_m$, and in the intersection of Ω_k and $\mathtt{C}_m$, respectively. H is the entropy where

$$\begin{aligned} H(\Omega) &= -\sum_{k=1}^{K} P(\Omega_k) \log P(\Omega_k) \\ H(\mathtt{C}) &= -\sum_{m=1}^{M} P(\mathtt{C}_m) \log P(\mathtt{C}_m) \end{aligned} \tag{2.40}$$

Table 2.5 shows the different accuracies and NMI for the dataset when applying the eight methods. According to this table it is clear that the IGGM+FS outperformed all other methods. Note that, the student t test has shown that the difference between the categorization accuracies of our infinite model when changing its hyperparameters is not statistically significant (The minimum P-value = 0.4548)

2.5 Discussion

In this chapter we have proposed a hierarchical infinite mixture model of generalized Gaussian distributions for visual learning based on non-parametric Bayesian estimation. The specific choice

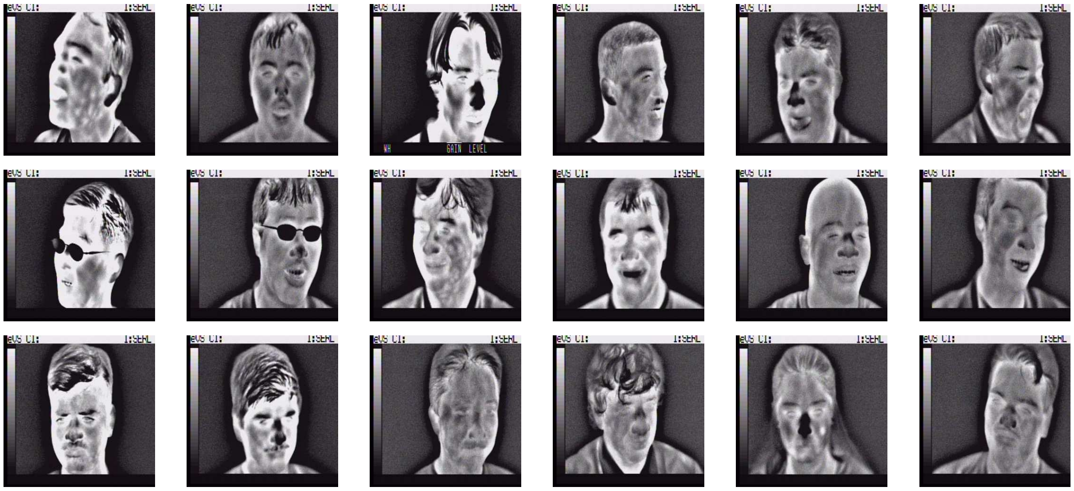


Figure 2.7: Sample images from each group. Row 1: Surprise, Row 2: Happy, Row 3: Angry.

of infinite mixture models is motivated by the fact that they combine flexibility in modeling, clarity of interpretation and intuitive analysis which is crucial in statistical inference from image data generally supposed to be generated from different sources. We have shown that fully Bayesian models provide a rigorous framework for challenging applications due to its ability to handle uncertainties associated with the involved data, by incorporating prior knowledge, and to its deep foundation on probability inference. According to the results, it is clear that performing feature selection in tandem with infinite mixture models leads to excellent clustering results and avoids overfitting. The experiments show clearly the broad applicability and generality of the proposed approach which is able to infer at the same time both meaningful clusters and meaningful features.

The adoption of generalized Gaussian is supported by various studies in image statistics which have shown that the statistics of natural images are generally non-Gaussian. The Bayesian clustering approach via infinite mixture models is clearly attractive in part due to recent advances in MCMC techniques which allows straightforward posteriors computation. However, it is also hindered by the very high computational cost.

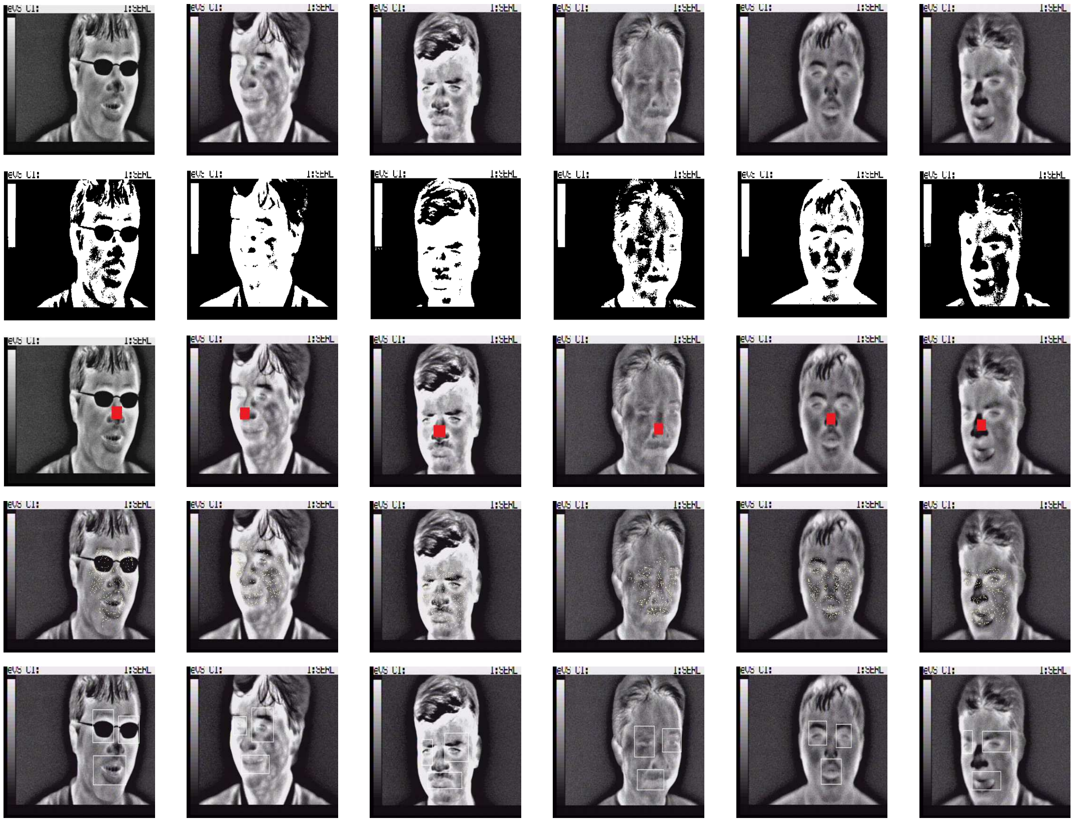


Figure 2.8: Processing steps shown for sample images from each group. Row 1: sample images, Row 2: Thresholding, Row 3: Center location, Row 4: Interest points detection, Row 5: Regions of interest extraction.

Table 2.4: Confusion matrices for the infrared facial expression recognition application using: (a) GM, (b) IGM, (c) GGM, (d) IGGM, (e) GM+FS, (f) IGM+FS, (g) GGM+FS, (h) IGGM+FS.

	Happy	Angry	Surprise
Happy	101	35	32
Angry	23	131	14
Surprise	35	15	118

(a)

	Happy	Angry	Surprise
Happy	109	31	28
Angry	22	137	9
Surprise	28	13	127

(b)

	Happy	Angry	Surprise
Happy	119	27	22
Angry	21	141	6
Surprise	24	11	133

(c)

	Happy	Angry	Surprise
Happy	124	23	21
Angry	21	143	4
Surprise	22	9	137

(d)

	Happy	Angry	Surprise
Happy	115	27	26
Angry	20	139	9
Surprise	26	11	131

(e)

	Happy	Angry	Surprise
Happy	121	26	21
Angry	18	144	6
Surprise	23	7	138

(f)

	Happy	Angry	Surprise
Happy	126	23	19
Angry	9	157	2
Surprise	21	5	142

(g)

	Happy	Angry	Surprise
Happy	133	21	14
Angry	7	161	0
Surprise	16	3	149

(h)

Table 2.5: Accuracies and NMI when applying the 8 methods for the infrared facial expression recognition.

	Accuracy(%)	NMI
IGGM+FS	87.90 (±2.15)	0.6249 (±0.0143)
GGM+FS	84.33 (±2.60)	0.5389 (±0.0173)
IGM+FS	79.96 (±3.35)	0.4407 (±0.0127)
GM+FS	76.39 (±4.65)	0.3713 (±0.0261)
IGGM	80.15 (±3.10)	0.4439 (±0.0206)
GGM	77.98 (±2.90)	0.4010 (±0.0193)
IGM	74.01 (±4.20)	0.3348 (±0.0201)
GM	69.44 (±4.70)	0.2668 (±0.0182)

Chapter 3

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

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

3.1 Introduction

Over the last decade, automatic segmentation of foreground from background in video sequences has attracted lots of attention in computer vision [55, 126, 127]. Foreground segmentation is often used as the primary step in video surveillance [128–130], optical motion capture [131, 132], and multimedia [133] in order to model the background and to detect the moving objects in the scene. Background subtraction involves the extraction of a background image which doesn't include any moving object, reference image, then subtracting each new frame from this image and thresholding the result in order to highlight regions of non-stationary objects. Normally, video surveillance systems can be employed in two kinds of environments: controlled and uncontrolled. Monitoring systems in controlled or indoor environments (i.e. airports, warehouses, and production plants) are easier to implement as they don't depend on weather changes. Uncontrolled environment is used to refer to outdoor scenes where illumination and temperature changes occur frequently, and where various atmospheric conditions can be observed. Normally, when developing background subtraction algorithms, there are two major problems that must be taken into consideration namely robustness and adaptation. These methods should be robust to illumination and weather changes, as well as able to detect addition, occlusion, and removal of objects in the scene. To take into account these problems of robustness and adaptation, many background modeling methods have been developed (a complete detailed survey can be found in [134]).

In the past, computational barriers have limited the complexity of real-time video processing applications. As a consequence, most systems were either too slow to be practical, or succeeded by restricting themselves to very controlled situations. Recently, faster computers have enabled researchers to consider more complex, robust models for real-time analysis of streaming data. These new methods allow researchers to begin modeling real world processes under varying conditions. Most recent methods assume that the images of the scene without the intruding objects exhibit some regular behavior that can be well described by a statistical model. If we have a statistical model of the scene, an intruding object can be detected by spotting the parts of the image that don't fit the model. In the majority of these methods, a common bottom-up approach has been applied to construct a probability density function for each pixel separately. Its idea is to segment the foreground moving objects by constructing over time a mixture model for each pixel and deciding, in a new input frame, whether the pixel belongs to the foreground or the background [2, 135]. Among the vast amount of approaches that have been proposed to accomplish this task, adaptive

Gaussian mixture models (GMMs) [2, 3] have proven their outstanding suitability in the surveillance domain because of their ability to achieve many of the requirements of a surveillance system, e.g. adaptability and multimodality, in real-time with low memory requirements. GMMs model the history of each pixel by a mixture of K Gaussian distributions. In [136], the authors implemented a pixel-wise EM framework for detection of vehicles by attempting to explicitly classify the pixel values into three separate predetermined distributions corresponding to the road color, the shadow color, and colors corresponding to vehicles. Stauffer et al. [2] generalized this idea by implementing on-line K-means approximation algorithm for modeling each pixel using a mixture of K Gaussians, where K was chosen in the range (3 to 5) depending on the computational power of the machine. In [3] the use of a negative prior evidence was introduced in order to discard the components that are not supported by the data, therefore being able to automatically select the number of components of the mixture used for each pixel. In [137] the use of an adaptive learning rate calculated for each Gaussian at every frame was proposed which led to an improved segmentation performance compared to the standard method. However, these methods have some drawbacks. Modeling the background using the GMM implies the assumption that the background and foreground distributions are Gaussians which isn't always the case as argued by [138]. Figure 3.1 shows the probability density function of a pixel throughout a video. From this figure we can notice that the distribution is not symmetrical. Applying the GMM, we can observe its inefficiency in modeling the data. In order to overcome these problems, some researchers have shown that the generalized Gaussian mixture (GGM) can be a good choice to model non-Gaussian data [20, 21, 139]. Compared to the Gaussian distribution (GD), the generalized Gaussian distribution (GGD) has one more parameter λ that controls the tail of the distribution: the larger the value of λ is, the flatter is the distribution; the smaller λ is, the more peaked is the distribution. Despite the higher flexibility that GGD offers, it is still a symmetric distribution inappropriate to model non-symmetrical data. From Fig. 3.1, we can recognize that the GGM is not suitable in modeling our data. In this chapter, we suggest the use of the asymmetric Gaussian distribution (AGD) capable of modeling asymmetrical data. The AGD uses two variance parameters for left and right parts of the distribution, which allow it to change its shape. As shown in Fig. 3.1 we can notice that the asymmetric Gaussian mixture (AGM) was able to accurately model the data.

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

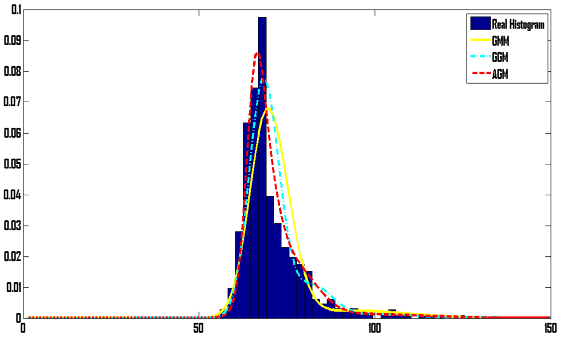


Figure 3.1: Probability density function of a pixel throughout a video sequence.

classified, from a computational point of view, into two classes: deterministic and stochastic methods. Deterministic methods estimate the model parameters for different range of M then choose the best value that maximize a model selection criteria such as the Akaike information criterion (AIC) [29], the minimum description length (MDL) [30] and the Laplace empirical criterion (LEC) [14]. 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 [36]. Despite their formal appeal, MCMC methods are too computationally demanding, therefore can't be applied for online applications such as foreground segmentation. For this reason, we are interested in deterministic approaches. In our proposed method, we use K-means algorithm to initialize the AGM parameters and successfully solve the initialization problem. The number of mixture components is automatically determined by implementing the minimum message length (MML) criterion [31] into the expectation-maximization (EM) algorithm. Therefore, the method can integrate simultaneously parameters estimation and model selection in a single algorithm, thus it is totally unsupervised.

Shadows, areas where direct light from a light source can not reach due to obstruction by different objects, are an ever-present aspect of color images. As a result of the difference between the light intensity reaching a shaded region and a directly lit region, shadows are often characterized by conspicuous strong brightness gradients. In outdoor scenes, the change between shadow and non-shadow regions is not entirely a brightness difference, but a color one as well. This property makes shadow detection task a highly problematic one in a number of different fields. Recently, there have been few studies concerning shadow removal, however, the best performing methods still require user interaction with image sequences to perform optimally. In our moving shadow detection algorithm, we implement a method compatible with the RGB color model and able to use our mixture model.

The rest of this chapter is organized as follows. First, we describe the AGM model and its learning algorithm. Then, we assess the performance of the new model with shadow removal scheme for foreground segmentation; while comparing it to other models. Finally, we discus the merits and demerits of our approach.

3.2 Finite AGM Model

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

$$p(\vec{X}|\Theta) = \sum_{j=1}^{M} p_j p(\vec{X}|\xi_j) \tag{3.1}$$

where

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

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

$$p(\vec{X}|\xi_j) = \prod_{k=1}^{d} \sqrt{\frac{2}{\pi}} \frac{1}{(\sigma_{l_j} + \sigma_{r_j})} \begin{cases} \exp\left[- \frac{(X_k - \mu_{jk})^2}{2\sigma_{l_{jk}}^2} \right] & \text{if } X_k < \mu_{jk} \\ \exp\left[- \frac{(X_k - \mu_{jk})^2}{2\sigma_{r_{jk}}^2} \right] & \text{if } X_k \geq \mu_{jk} \end{cases} \tag{3.2}$$

where $\xi_j = (\vec{\mu}_j, \vec{\sigma}_{l_j}, \vec{\sigma}_{r_j})$ is the set of parameters of component j where $\vec{\mu}_j$ = ($\mu_{j1}\ldots$,μ_{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 AGD, respectively. The AGD is chosen to be able to fit, in analytically simple and realistic way, symmetric or non-symmetric data by the combination of the left and right variances.

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

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{M} p(\vec{X}_i|\xi_j)p_j \tag{3.3}$$

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

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

$$p(\mathcal{X}, Z|\Theta) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left(p(\vec{X}_i|\xi_j)p_j\right)^{Z_{ij}} \tag{3.4}$$

3.2.1 Maximum Likelihood Estimation of the Mixture Parameters

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

$$L(\Theta, Z, \mathcal{X}) = \sum_{i=1}^{N} \sum_{j=1}^{M} Z_{ij} \log\left(p(\vec{X}_i|\xi_j)p_j\right) \tag{3.5}$$

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

$$\hat{Z}_{ij} = p(j|\vec{X}_i) = \frac{p(\vec{X}_i|\xi_j)p_j}{\sum_{j=1}^{M} p(\vec{X}_i|\xi_j)p_j} \tag{3.6}$$

Using equation 3.6 we can affect each observation to one of the M clusters. Now, using these expectations, we want to maximize the complete data log-likelihood with respect to our model parameters. This can be done by taking the gradient of the log-likelihood with respect to p_j, $\vec{\mu}_j$, $\vec{\sigma}_{l_j}$, and $\vec{\sigma}_{r_j}$. When estimating p_j we actually need to introduce Lagrange multiplier to ensure that the constraints $p_j > 0$ and $\sum_{j=1}^{M} p_j = 1$ are satisfied. Thus, the augmented log-likelihood function can be expressed by:

$$\Phi(\Theta, Z, \mathcal{X}, \Lambda) = \sum_{i=1}^{N} \sum_{j=1}^{M} Z_{ij} \log \left(p(\vec{X}_i|\xi_j)p_j \right) + \Lambda(1 - \sum_{j=1}^{M} p_j) \tag{3.7}$$

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

$$\hat{p}_j = \frac{1}{N} \sum_{i=1}^{N} p(j|\vec{X}_i) \tag{3.8}$$

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

$$\hat{\mu}_{jk} = \frac{\sum_{i=1}^{N} \hat{Z}_{ij} X_{ik}}{\sum_{i=1}^{N} \hat{Z}_{ij}} \tag{3.9}$$

$$\sum_{i=1, X_{ik} < \mu_{jk}}^{N} \frac{\hat{Z}_{ij}(X_{ik} - \mu_{jk})^2}{\sigma_{l_{jk}}^3} - \sum_{i=1}^{N} \frac{\hat{Z}_{ij}}{\sigma_{l_{jk}} + \sigma_{r_{jk}}} = 0 \tag{3.10}$$

$$\sum_{i=1, X_{ik} \geq \mu_{jk}}^{N} \frac{\hat{Z}_{ij}(X_{ik} - \mu_{jk})^2}{\sigma_{r_{jk}}^3} - \sum_{i=1}^{N} \frac{\hat{Z}_{ij}}{\sigma_{l_{jk}} + \sigma_{r_{jk}}} = 0 \tag{3.11}$$

We can notice that equations 3.10 and 3.11 are nonlinear, so we have decided to use the Newton-Raphson method for estimation:

$$\hat{\sigma}_{l_{jk}} \simeq \sigma_{l_{jk}} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk}}^2} \right)^{-1} \left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk}}} \right) \right] \tag{3.12}$$

$$\hat{\sigma}_{r_{jk}} \simeq \sigma_{r_{jk}} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk}}^2} \right)^{-1} \left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk}}} \right) \right] \tag{3.13}$$

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

3.2.2 Model Selection Using the Minimum Message Length Criterion

Different model selection methods have been introduced to estimate the number of components of a mixture. In this chapter, we are interested with deterministic approaches especially MML. The MML approach is based on evaluating statistical models according to their ability to compress a message containing the data (minimum coding length criteria). High compression is obtained by forming good models of the data to be coded. For each model in the model space, the message includes two parts. The first part encodes the model, using only prior information about the model and no information about the data. The second part encodes only the data in a way that makes use of the model encoded in the first part. When applying the MML, the optimal number of classes of the mixture is obtained by minimizing the following function [31, 140]:

$$\begin{aligned} MessLen &\approx -\log(p(\Theta)) - L(\Theta, Z, \mathcal{X}) + \frac{1}{2}\log|F(\Theta)| \\ &+ \frac{N_p}{2} - \frac{1}{2}\log(12) \end{aligned} \tag{3.14}$$

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

Derivation of the Prior $p(\Theta)$

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

$$p(\Theta) = p(P)\prod_{j=1}^{M} p(\vec{\mu}_j)p(\vec{\sigma}_{l_j})p(\vec{\sigma}_{r_j}) \tag{3.15}$$

where $P = (p_1, \ldots, p_M)$. In what follows, we will specify each of these priors separately. Starting with $p(P)$, we know that $P = (p_1,\ldots,p_M)$ is defined on the simplex $\{(p_1, \ldots, p_M) : \sum_{j=1}^{M} p_j = 1\}$.

Then, a natural choice as a prior for this vector is the Dirichlet distribution

$$p(P) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \tag{3.16}$$

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

$$p(P) = (M-1)! \tag{3.17}$$

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

$$p(\vec{\mu}_j) = \prod_{k=1}^{d} p(\mu_{jk}) = \prod_{k=1}^{d} \frac{1}{(\sigma_{l_k} + \sigma_{r_k})} \tag{3.18}$$

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

$$p(\vec{\sigma}_{l_j}) = \prod_{k=1}^{d} p(\sigma_{l_{jk}}) = \prod_{k=1}^{d} \frac{1}{\sigma_{l_k}} \tag{3.19}$$

$$p(\vec{\sigma}_{r_j}) = \prod_{k=1}^{d} p(\sigma_{r_{jk}}) = \prod_{k=1}^{d} \frac{1}{\sigma_{r_k}} \tag{3.20}$$

Finally, by replacing the priors of the parameter in equation 3.15 by each prior value in equations 3.17, 3.18, 3.19, and 3.20 we get

$$p(\Theta) = (M-1)! \prod_{k=1}^{d} \frac{1}{\sigma_{l_k}^{M} \sigma_{r_k}^{M} (\sigma_{l_k} + \sigma_{r_k})^{M}} \tag{3.21}$$

Derivation of the Determinant of the Fisher Information Matrix $|F(\Theta)|$

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

$$|F(\Theta)| = |F(P)| \prod_{j=1}^{M} |F(\vec{\mu}_j)||F(\vec{\sigma}_{l_j})||F(\vec{\sigma}_{r_j})| \tag{3.22}$$

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

$$|F(P)| = \frac{N^{M-1}}{\prod_{j=1}^{M} p_j} \tag{3.23}$$

For $|F(\vec{\mu}_j)|$, $|F(\vec{\sigma}_{l_j})|$, and $|F(\vec{\sigma}_{r_j})|$ let us consider the j^{th} class $\mathcal{X}_j$= $(\vec{X}_l, \ldots, \vec{X}_{l+n_j-1})$ of the mixture as the data in class j after classifying all the data $\mathcal{X}$ using the maximum a posteriori probability defined by Eq. 3.6. Note that n_j is the number of data vectors belonging to the j^{th} distribution. This given choice of the j^{th} class allows us to simplify the notation without loss of generality. Then, the Hessian matrices when we consider the vectors $\vec{\mu}_j$, $\vec{\sigma}_{l_j}$, and $\vec{\sigma}_{r_j}$ are given by:

$$F(\vec{\mu}_j)_{k_1,k_2} = \frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \mu_{jk_1} \partial \mu_{jk_2}} \tag{3.24}$$

$$F(\vec{\sigma}_{l_j})_{k_1,k_2} = \frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \sigma_{l_{jk_1}} \partial \sigma_{l_{jk_2}}} \tag{3.25}$$

$$F(\vec{\sigma}_{r_j})_{k_1,k_2} = \frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \sigma_{r_{jk_1}} \partial \sigma_{r_{jk_2}}} \tag{3.26}$$

where $(k_1, k_2) \in (1, \ldots, d)$. Using appendix B to compute the derivatives in equations 3.24, 3.25, 3.26, we obtain

$$|F(\vec{\mu}_j)| = \prod_{k=1}^{d} \Bigg[\sum_{i=l, X_{ik} < \mu_{jk}}^{l+n_j-1} \frac{1}{\sigma_{l_{jk}}^2} + \sum_{i=l, X_{ik} \geq \mu_{jk}}^{l+n_j-1} \frac{1}{\sigma_{r_{jk}}^2} \Bigg] \qquad for \qquad j = 1, \ldots, M \tag{3.27}$$

$$|F(\vec{\sigma}_{l_j})| = \prod_{k=1}^{d} \Bigg[\sum_{i=l}^{l+n_j-1} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})^2} - 3 \sum_{i=l, X_{ik} < \mu_{jk}}^{l+n_j-1} \frac{(X_{ik} - \mu_{jk})^2}{\sigma_{l_{jk}}^4} \Bigg] \qquad for \qquad j = 1, \ldots, M \tag{3.28}$$

$$|F(\vec{\sigma}_{r_j})| = \prod_{k=1}^{d} \Bigg[\sum_{i=l}^{l+n_j-1} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})^2} - 3 \sum_{i=l, X_{ik} \geq \mu_{jk}}^{l+n_j-1} \frac{(X_{ik} - \mu_{jk})^2}{\sigma_{r_{jk}}^4} \Bigg] \qquad for \qquad j = 1, \ldots, M \tag{3.29}$$

3.2.3 The AGM Model Learning Algorithm

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

Input: Data set $\mathcal{X}$ and M_{max}

Output: Θ_{M^*} (the values of Θ when M^* components are chosen) and M^*

```
Step 1: For $M = 1 : M_{max}$ do{
        1. Initialize the parameters.
        2. Repeat until convergence.
           (a) The E-step given by Eq. 3.6.
           (b) The M-step given by Eqs. 3.8, 3.9, 3.12, 3.13.
        3. Calculate the associated message length using Eq. (3.14).
 }END FOR
Step 2: Select the model $M^*$ with the smallest message length.
```

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. In order to detect the convergence of the EM, we stop the iterations when the difference of the log-likelihood from two successive iterations ℓ and $\ell+1$ is smaller than a predefined threshold ϵ.

3.3 Background Subtraction

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

3.3.1 Adaptive AGM algorithm

Adaptive Gaussian mixture algorithms are widely applied for background subtraction. In [2], the authors presented an online learning of a GMM for each pixel in the video frames. Their idea was to model each pixel in the scene by a mixture of K Gaussian distributions, where K is taken in the range (3-5). Then, they ordered the K distributions based on the fitness value p_j/σ_j and used the first B distributions to model the background of the scene, where B is estimated as

$$B = \arg\min_b \sum_{j=1}^{b} p_j > T, \tag{3.30}$$

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

$$\hat{p}_j^{\ell+1} = (1-\alpha)\hat{p}_j^{\ell} + \alpha\hat{p}(j|X^{\ell+1}) \tag{3.31}$$

$$\hat{\mu}_j^{\ell+1} = (1-\rho)\hat{\mu}_j^{\ell} + \rho X^{\ell+1} \tag{3.32}$$

$$\hat{\Sigma}_j^{\ell+1} = (1-\rho)\hat{\Sigma}_j^{\ell} + \rho(X^{\ell+1} - \hat{\mu}_j^{\ell+1})^T(X^{\ell+1} - \hat{\mu}_j^{\ell+1}) \tag{3.33}$$

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

$$\hat{p}(j|X^{\ell+1}) = \begin{cases} 1 & if\ X^{\ell}\ match\ the\ class\ j \\ 0 & otherwise \end{cases} \tag{3.34}$$

Finally, ρ is defined as:

$$\rho = \alpha \mathcal{N}(X^{\ell+1}; \hat{\mu}_j^\ell, \hat{\Sigma}_j^\ell) \tag{3.35}$$

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

In the case when none of the K distributions matches that pixel value, the least probable component is replaced by a distribution with the current value as its mean, an initially high variance, and a low weight parameter. According to their papers [2, 135, 141], only two parameters, α and T, have to be set in the method. However, there are four major problems when using this method. First, modeling both foreground and background pixels by a mixture of Gaussians implies the assumption that they are symmetrical which isn't always the case. Second, using a prefixed number of components to represent all pixels mixtures is not practical in real life. The method is not robust when dealing with busy environments because a clean background is rare. Last but not least, slow adaptations in the means and the covariance matrices, therefore the tracker can fail within a few seconds after initialization. In this chapter, we are trying to overcome these drawbacks by:

1. Using the AGM to model the non-symmetricity of the pixels distributions.

2. Using the MML to choose the right number of components for each pixel distribution.

3. Using another method to update the mixture parameters.

4. Removing the connection between the likelihood term and ρ.

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

$$p_j^{\ell+1} = p_j^\ell + B_\ell[p(j|\vec{X}^{\ell+1}) - p_j^\ell] \tag{3.36}$$

$$\xi_j^{\ell+1} = \xi_j^{\ell} + B_\ell \left[p(j|\vec{X}^{\ell+1}) \frac{\partial L(\Theta, Z, \vec{X}^{\ell+1})}{\partial \xi_j} \right] \tag{3.37}$$

where B_ℓ represents any sequence of positive numbers that decreases to zero. The derivatives in Eq. 3.37 are given in appendix A. If no match occurs, we create a new component for the mixture with the mean equal to the new value of the pixel. Then, we evaluate the new model $\mathcal{M}^{\ell+1}$ with MML (note that $\mathcal{M}^{\ell+1}$ denotes the mixture model associated with the pixel $\vec{X}$ at time $\ell+1$). In other words, we calculate the message length for the new mixture model with $M+1$ components: if $MessLen(\ell) > MessLen(\ell+1)$, then we use $\mathcal{M}^{\ell+1}$ else we use $\mathcal{M}^{\ell}$ and update the parameters using Eqs. (3.36) and (3.37). In the case when there is an empty component j, $p_j = 0$, we discard the component j of the mixture and set $M \leftarrow M-1$. Finally, using the same idea of [2], we order the mixture components by the value of $\frac{p_j}{||\vec{\sigma}_{l_j}||+||\vec{\sigma}_{r_j}||}$, where $||\vec{\sigma}_{l_j}||$ and $||\vec{\sigma}_{r_j}||$ are the norm of the left and right standard deviations of the component j, respectively. At that point, we use Eq. 3.30 to model the background by the first B components.

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

Step 1- Initialization for each pixel X:

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

Step 2- For each pixel $\vec{X}^{(\ell+1)}$ in a new frame **do** {

1. Verify if there is a match that exists for the new pixel value.
   (a) **If True**
      i. Update the new pixel model parameters using equations 3.36 and 3.37.
   (b) **If False**
      i. Add a new component to the mixture with mean equal the new pixel value.
      ii. Evaluate the new model $\mathcal{M}^{\ell+1}$ with MML.
         A. **If** $MessLen(\ell) > MessLen(\ell+1)\{M \leftarrow M+1\}$
         B. **If** $MessLen(\ell) < MessLen(\ell+1)$ { use $\mathcal{M}^{\ell}$ and update it using equations 3.36 and 3.37}**.**

2. Check if there is any p_j= 0 **then** $\{$
   $M \leftarrow M - 1 \}$.
3. Order the pixel mixture components by the values of $\frac{p_j}{||\vec{\sigma}_{l_j}||+||\vec{\sigma}_{r_j}||}$.
4. Use equation 3.30 to extract foreground objects.

3.3.2 Shadow Detection Algorithm

Related Works

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

Moving shadows in the scene are caused by the occlusion of light sources due to the moving objects. Therefore, shadow points have lower luminance values but similar chromaticity values. However, the texture characteristic around the shadow points remains unchanged since shadows do not alter the background surfaces. Shadow detection is known to be a challenging task because: (1) shadow points are mostly classified as foreground, since they differ significantly from the background; (2) shadow has the same motion as the moving object causing it, which make the task of differentiating between them very difficult; (3) shadow is always adjacent to moving object, which make it hard to remove using common segmentation techniques.

Generally, shadow detection algorithms can be classified into three categories: color-based, texture-based, and statistic-based. The color-based approaches attempt to describe the change in the color features of shadow pixels. In [142], the authors presented a robust shadow detection approach based on brightness, saturation, and hue properties in the HSV color space. Their idea is built on the hypothesis that shadows reduce surface brightness and saturation while maintaining hue properties in the HSV color space. The work in [143] addressed the shadow detection problem in YUV color space in order to avoid the time consuming HSV color space transformation. They distinguished the shadow regions from the foreground regions according to the observation that the YUV pixel value of shadows is lower than the linear pixels. According to the shadow model, Salvador et al. [144] identified an initial set of shadow pixels in RGB color space basing on the fact that shadow region darkens the surface. Then, combining color invariance with geometric properties of shadow they were able to detect the shadows in different scenes. In [145], Horprasert et al. built a model in the RGB color space to express normalized luminance variation and chromaticity distortions. Though color-based methods have shown their efficiency in shadow detection, they may not be reliable in the case when moving objects have similar color as moving shadows.

On the other hand, texture-based approaches are based on the fact that texture of shadow regions and the background are similar, while the texture of moving objects are different from the background. In [146], the authors explored ratio edges for shadow detection. They proved that ratio edge is illumination invariant and that the distribution of normalized ratio edge difference is a chi-square distribution. Then, a significance test was used to detect shadows. In addition to using scene brightness distortion and chromaticity distortion, Choi et al. [147] proposed three estimators which use the properties of chromaticity, brightness, and local intensity ratio. Hence, creating a chromaticity difference that obey a standard normalize distribution between the shadow region and the background. Finally, Finlayson et al. [148] used shadow edges along with illuminant invariant images to recover full color shadow-free images. Hence, texture-based methods may be the most promising technique for shadow detection because they are capable of capturing textual information of different scenes. However, they suffer from three major problems: (1) they require to set parameters for different scenes, (2) they can not handle complex and time-varying lighting conditions, and (3) they are too computationally demanding which limit their applications.

Recently, the statistical prevalence of cast shadows had been employed to learn shadows in the scenes. The principle of statistic-based methods is to build pixel-based statistical models in order to detect cast shadows. In [4], the authors used an adaptive Gaussian Mixture Model to detect moving cast shadows. The method consists of building a GMM to segment moving objects. Then, they identified the distribution of moving objects from shadows using an effective computational colour model similar to the one proposed in [145]. The work in [149] proposed the use of a Gaussian Mixture Shadow Model (GMSM). The algorithm models moving cast shadows of non-uniform and varying intensity, and builds statistical models to segment moving cast shadows by using the GMM learning ability. However, the shadow models need a long time to converge while the lighting conditions should remain stable which is a major drawback. Liu et al. [150] were able to remove shadow using multi-level information in HSV color space. They attempted to improve the convergence speed of pixel-based shadow model by using multi-level information. They used region-level information to increase the number of samples and global level information to update a pre-classifier. However, the method still suffers from the slow learning of the conventional GMM [2], and the low discriminativity of the pre-classifier in scenes having different types of shadows. Statistical-based methods are widely applied due to their robustness in different scenes, however, they are less effective in a real-world environment.

Proposed Approach

In this section, we present a novel pixel-based statistical approach to model moving cast shadows. Normally, when using adaptive mixture algorithms, values that are frequently seen by a pixel are captured into stable background distributions while values that are infrequently seen are classified into foreground objects. Shadow values lie between both situations: They are not as frequent as background values but their rate of appearance is higher than random foreground values. So, in most cases, they are classified as foreground objects. Hence, the purpose of our shadow detection algorithm is to remove cast shadows classified by the adaptive AGM as foreground objects. Our idea is very simple and makes use of the property that shadows darken the surface upon which they are cast [144]. Let us consider $\vec{X}^{\ell+1} = (X_R^{\ell+1}, X_G^{\ell+1}, X_B^{\ell+1})$ belonging to the foreground distribution. We classify the pixel $\vec{X}^{\ell+1}$ as shadow if its distribution mean is smaller than that of the pixel background models for all three channels. The steps of this approach can be summarized as:

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

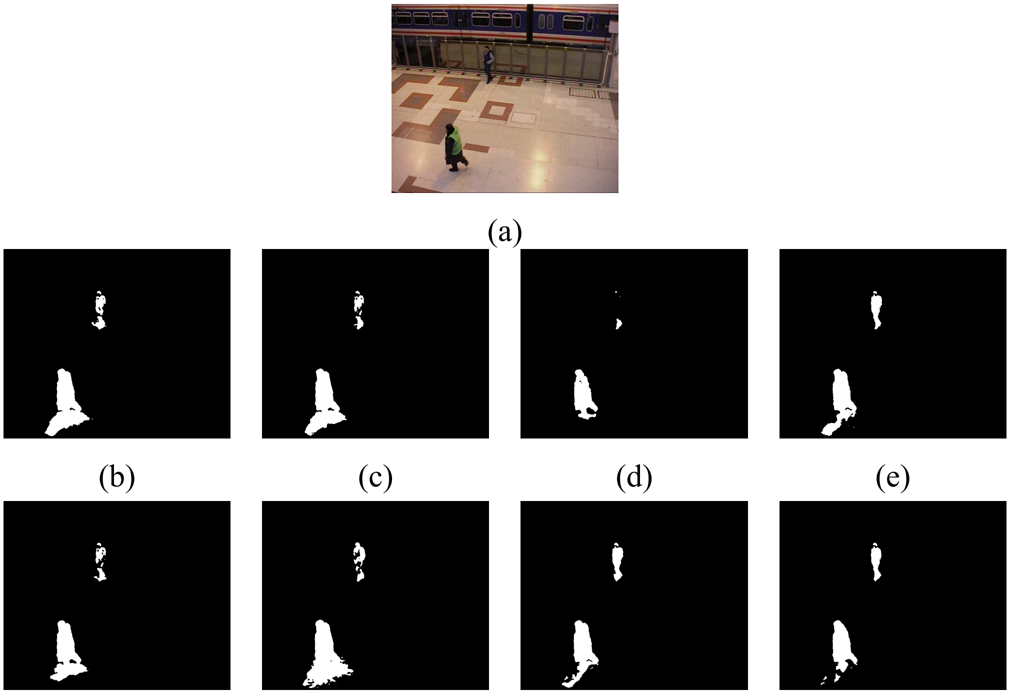
where μ_R^B, μ_G^B, μ_B^B are the B background distributions red, green, and blue means, respectively. Hence, after applying this algorithm we will end up with three different models for the background, foreground, and shadow.

3.3.3 Results

Our approach performance has been evaluated using the change detection dataset described in [151]. This dataset consists of 31 videos depicting indoor and outdoor scenes with boats, cars, trucks, and pedestrians that have been captured in different scenarios. The videos were taken with different cameras ranging from low-resolution IP cameras to thermal cameras. Therefore, spatial resolutions of the videos vary from 320 × 240 to 720 × 576 and the level of noise and compression artifacts varies from one video to another due to diverse lighting conditions present.

The videos are grouped into six categories according to the type of challenge each represents. The baseline category contains four videos, two indoor and two outdoor. There are six videos in the dynamic background category depicting outdoor scenes with strong background motion. The third category, Camera Jitter, contains one indoor and three outdoor videos captured by unstable cameras. Shadows: This category consists of two indoor and four outdoor videos exhibiting strong as well as faint shadows. Intermittent Object Motion is the fifth category which includes six videos with scenarios known for causing “ghosting” artifacts in the detected motion. The last category is composed of five (three outdoor and two indoor) sequences taken by far-infrared cameras. Figures (3.2- 3.7) show some sample frames taken from this dataset.

In order to validate our method, we have compared it with six state of the art methods. These methods can be divided into two main groups pixel based and Non-parametric Kernel Density Estimation (KDE) methods. For pixel based methods we have used: Stauffer et al. [2], Zivkovic [3], KaewTraKulPong et al. [4], and Evangelio et al. [5]; as for KDE methods we have chosen the methods introduced by ELgammal et al. [6] and Nonaka et al. [7]. Figures (3.2 to 3.7) show the segmentations of our method with and without Shadow detection as well as the six other methods. Note that in this application we set the maximum number of components for the AGM to 9 , the standard deviation factor $K = 2$, and the threshold $T = 0.6$. From figure 3.6, we can distinguish that the $AGM + SD$ wasn’t able to remove the shadow completely from the image because the difference in value between the shadow and foreground wasn’t large enough to construct a model that represents this shadow distribution. However, from qualitative evaluation, we can notice the



(a)

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

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

Figure 3.2: (a) Sample frame from Pets2006 video sequence in the baseline category, (b) Stauffer et al. [2], (c) Zivkovic [3], (d) KaewTraKulPong et al. [4], (e) Evangelio et al. [5], (f) ELgammal et al. [6], (g) Nonaka et al. [7], (h) AGM, (i) AGM+SD.

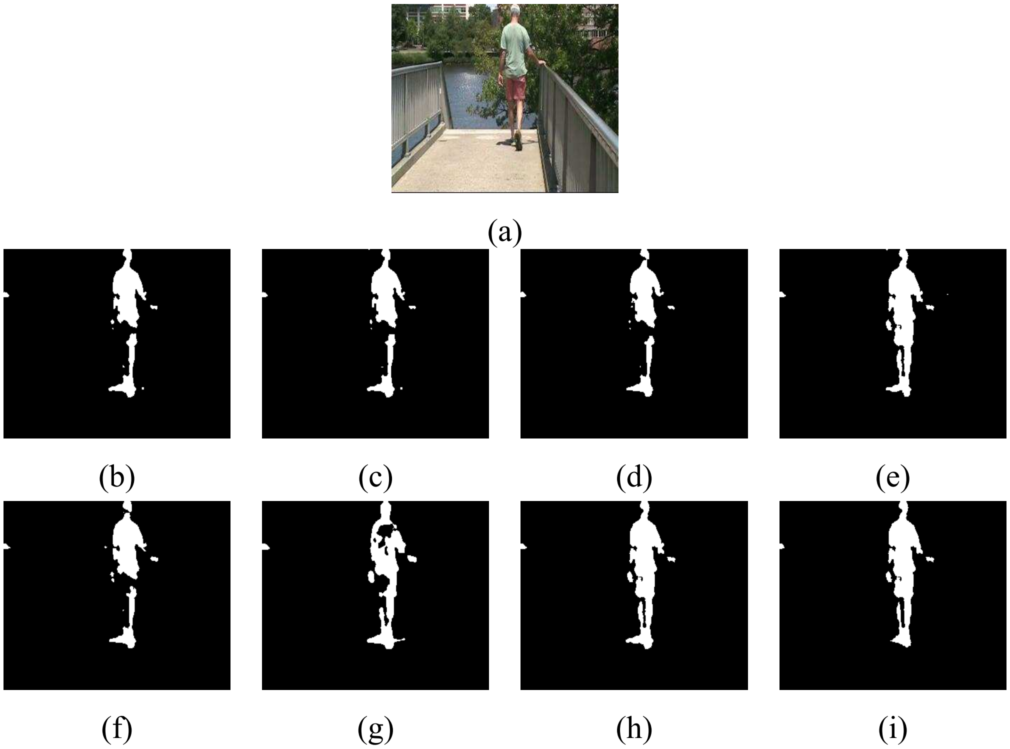
higher efficiency of our method.

In order to have a quantitative evaluation of the performance, we have used two well-known metrics, precision and recall, to quantify how well each algorithm works in classifying the data [152]. Precision (Eq. 3.38) represents the percentage of detected true positives to the total number of items detected by the algorithm. Recall (Eq. 3.39) is the percentage of number of detected true positives by the algorithm to the total number of true positives in the dataset:

$$Precision = \frac{TP}{TP + FP} \tag{3.38}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.39}$$

where TP is the total number of true positives correctly classified by the algorithm, FP is the total number of false positives, and FN is the number of true positives that were wrongly classified



(a)

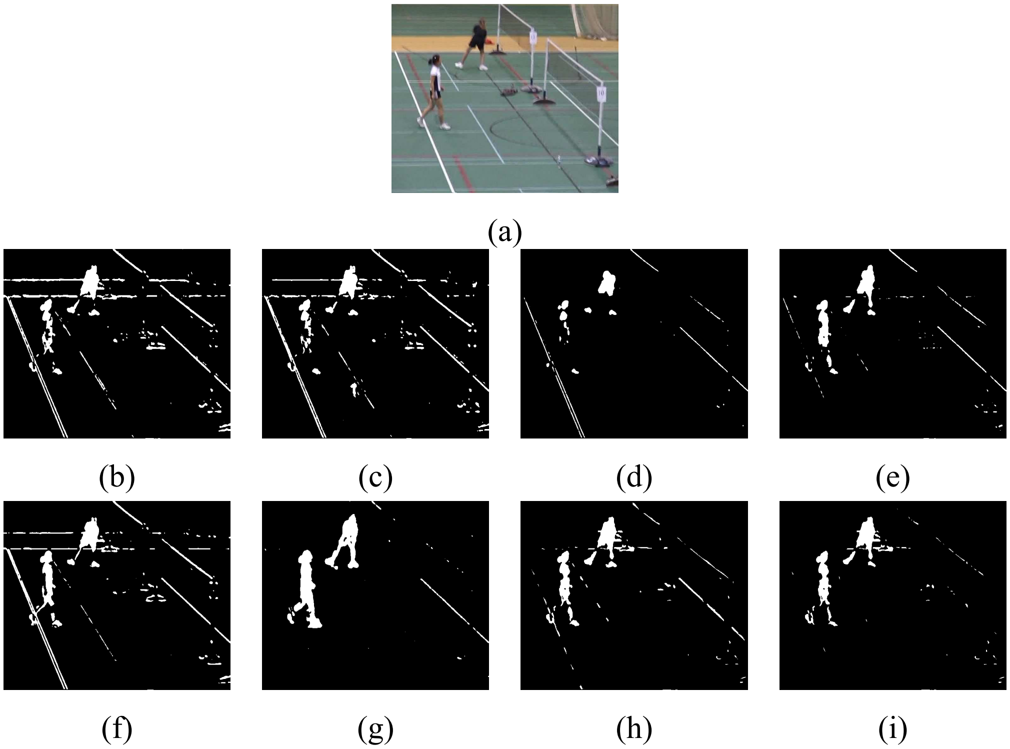
(b) (c) (d) (e)

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

Figure 3.3: (a) Sample frame from Overpass video sequence in the dynamic background category, (b) Stauffer et al. [2], (c) Zivkovic [3], (d) KaewTraKulPong et al. [4], (e) Evangelio et al. [5], (f) ELgammal et al. [6], (g) Nonaka et al. [7], (h) AGM, (i) AGM+SD.

as background (false negatives). Tables (3.1 to 3.7) show the average recall and precision for all methods. From table 3.7 we can deduce that our model is capable of detecting changes under different scenarios efficiently. According to quantitative and analytical analysis, we can conclude that the use of AGM in background detection with shadow detection increased the performance greatly.

In order to evaluate the effect of changing the parameters on the performance of our models, we have used the precision-recall curves. For simplicity, we have generated precision-recall curves by systematically changing the threshold parameter T and the standard deviation factor K. Figure 3.8 shows the effect of changing T and K on our method with and without Shadow detection. Based on the measurements shown in Figure 3.8, we can notice that both methods perform consistently very well. In addition, we can remark that varying T and K have little effect on the $AGM +$



(a)

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

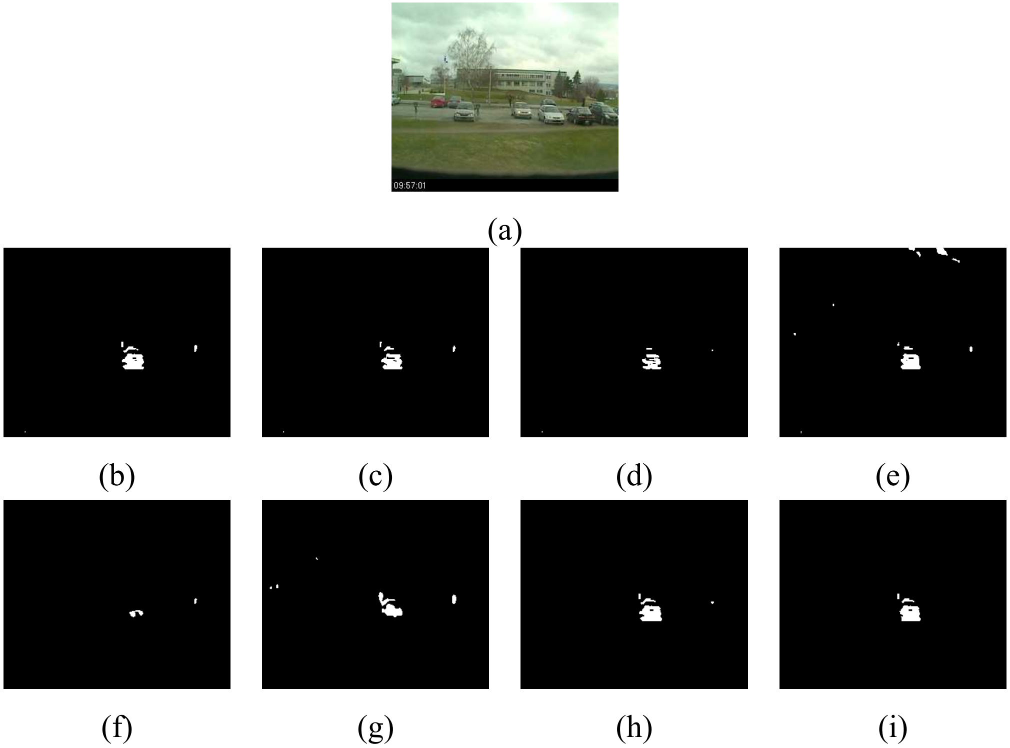
(f) (g) (h) (i)

Figure 3.4: (a) Sample frame from Badminton video sequence in the Camera Jitter category, (b) Stauffer et al. [2], (c) Zivkovic [3], (d) KaewTraKulPong et al. [4], (e) Evangelio et al. [5], (f) ELgammal et al. [6], (g) Nonaka et al. [7], (h) AGM, (i) AGM+SD.

SD performance, as for the AGM it is affected by T alteration. Furthermore, the high overall precision of both algorithms allows our methods to operate with a low false positive rate at their sensitive operating point. It is noteworthy that the computational time of our approach for adaptive background subtraction with and without shadow detection are around 12 and 13 frames per second for frames with 320 × 240 pixels resolution running on Core i7-2.4 GHz processor.

3.4 Discussion

Adaptive mixture models are popular methods for background modeling. The proposed method has provided three main improvements to the well-known adaptive Gaussian mixture model [2].



(a)

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

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

Figure 3.5: (a)Sample frame from Parking video sequence in the Intermittent Object Motion category, (b) Stauffer et al. [2], (c) Zivkovic [3], (d) KaewTraKulPong et al. [4], (e) Evangelio et al. [5], (f) ELgammal et al. [6], (g) Nonaka et al. [7], (h) AGM, (i) AGM+SD.

First, we have adopted the use of the asymmetric Gaussian distribution capable of modeling non-symmetrical data. Second, we have eliminated the problem of determining the number of clusters of the AGM by the use of the MML criterion. Finally, we have presented a novel pixel-based statistical approach for shadow detection and removal. Our shadow scheme identifies distributions of pixel values that could represent shadowed surfaces then uses them to build a second asymmetric Gaussian mixture model for shadows, hence, building a shadow models capable of evolving over time. Our background subtraction approach shows good performance in terms of adaptability, accuracy and robustness, in different indoor and outdoor scenes with complex illumination variations, background movements, shadows, and ghosting artifacts.

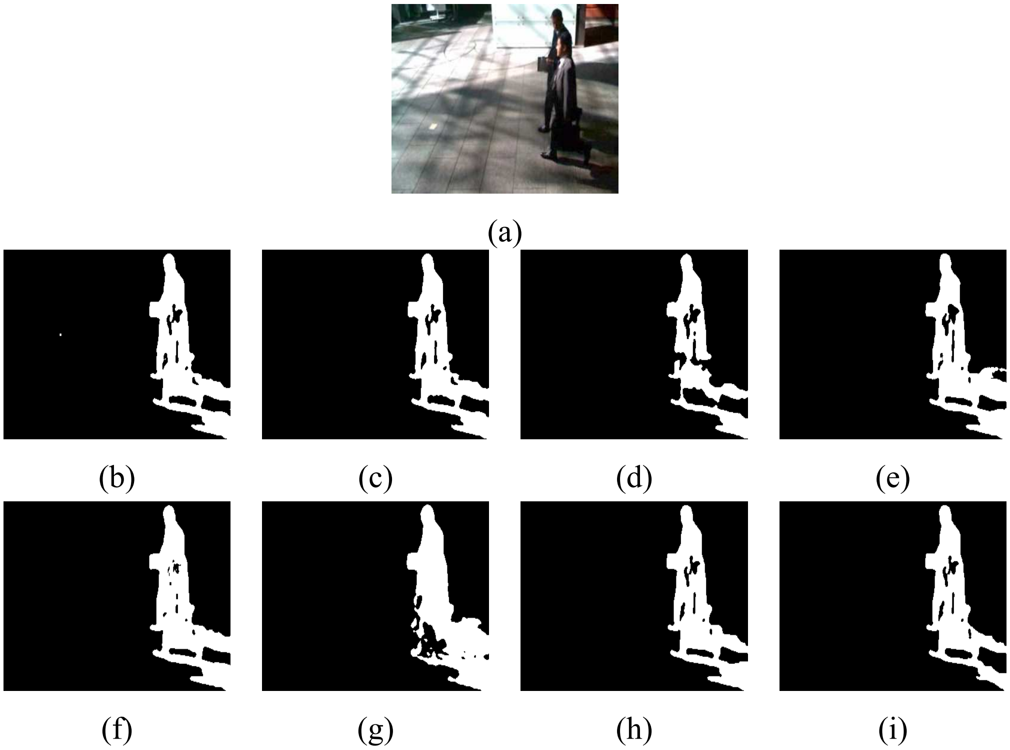
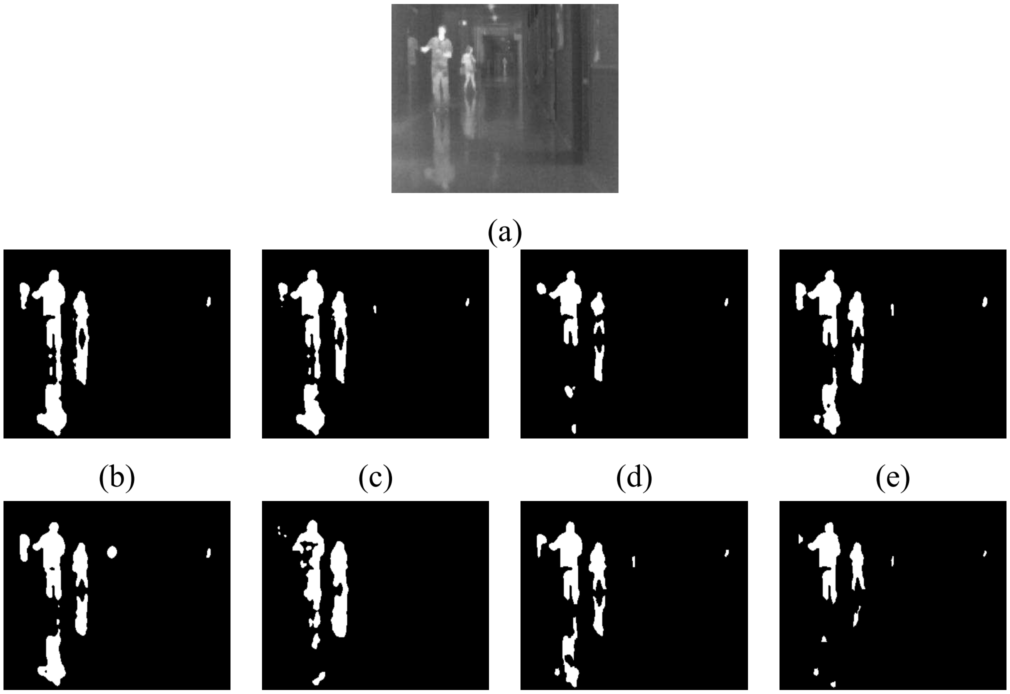


Figure 3.6: (a) Sample frame from PeopleInShade video sequence in the shadows category, (b) Stauffer et al. [2], (c) Zivkovic [3], (d) KaewTraKulPong et al. [4], (e) Evangelio et al. [5], (f) ELgammal et al. [6], (g) Nonaka et al. [7], (h) AGM, (i) AGM+SD.

Table 3.1: Baseline Precision and Recall

		Stauffer et al. [2]	Zivkovic [3]	KaewTraKulPong et al [4]	Evangelio et al. [5]	ELgammal et al. [6]	Nonaka et al. [7]	AGM	AGM+SD
Highway	Prec.	92.98%	91.63%	90.83%	91.29%	93.28%	90.33%	93.01%	93.89%
	Rec.	91.82%	89.16%	64.96%	87.81%	93.79%	94.59%	92.27%	89.07%
Office	Prec.	74.63%	92.90%	99.08%	81.58%	96.76%	81.54%	93.87%	96.69%
	Rec.	49.04%	50.75%	36.36%	67.48%	90.54%	30.36%	71.31%	65.10%
Pedestrians	Prec.	92.25%	93.86%	97.78%	79.64%	96.05%	73.72%	94.55%	97.91%
	Rec.	98.68%	98.20%	80.15%	95.93%	95.40%	99.72%	98.33%	97.18%
PETS2006	Prec.	78.56%	81.35%	93.60%	90.86%	82.84%	74.31%	84.51%	94.82%
	Rec.	87.65%	85.28%	53.04%	95.97%	79.04%	74.22%	89.23%	89.20%



(a)

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

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

Figure 3.7: (a) Sample frame from Corridor video sequence in the thermal category, (b) Stauffer et al. [2], (c) Zivkovic [3], (d) KaewTraKulPong et al. [4], (e) Evangelio et al. [5], (f) ELgammal et al. [6], (g) Nonaka et al. [7], (h) AGM, (i) AGM+SD.

Table 3.2: Dynamic Background Precision and Recall

		Stauffer et al. [2]	Zivkovic [3]	KaewTraKulPong et al [4]	Evangelio et al. [5]	ELgammal et al. [6]	Nonaka et al. [7]	AGM	AGM+SD
Boats	Prec.	70.14%	80.12%	89.72%	96.76%	60.89%	44.97%	92.61%	93.31%
	Rec.	23.87%	70.04%	68.34%	48.63%	65.75%	61.15%	71.40%	70.56%
Canoe	Prec.	89.82%	91.94%	99.51%	98.81%	93.96%	87.09%	94.18%	96.42%
	Rec.	86.59%	85.33%	67.76%	80.12%	83.15%	79.11%	86.15%	86.06%
Fountain01	Prec.	4.01%	4.31%	47.77%	3.99%	5.65%	6.51%	50.91%	51.07%
	Rec.	79.73%	75.06%	82.91%	73.76%	79.30%	96.35%	83.76%	82.13%
Fountain02	Prec.	74.51%	74.59%	97.99%	74.03%	79.55%	71.80%	95.30%	96.14%
	Rec.	87.17%	84.37%	59.06%	85.33%	85.28%	97.04%	86.10%	86.12%
Overpass	Prec.	91.91%	93.66%	85.97%	85.56%	85.12%	82.29%	91.90%	95.30%
	Rec.	82.94%	80.76%	73.74%	90.25%	80.03%	88.68%	84.22%	82.16%
Fall	Prec.	3.91%	28.17%	69.20%	40.33%	18.75%	32.12%	66.12%	71.27%
	Rec.	88.38%	85.60%	76.64%	84.79%	87.21%	81.75%	89.14%	84.66%

Table 3.3: Camera Jitter Precision and Recall

		Stauffer et al. [2]	Zivkovic [3]	KaewTraKulPong et al [4]	Evangelio et al. [5]	ELgammal et al. [6]	Nonaka et al. [7]	AGM	AGM+SD
Badminton	Prec.	63.70%	62.51%	92.11%	90.43%	66.68%	76.00%	89.41%	90.77%
	Rec.	75.53%	71.47%	54.80%	80.44%	79.04%	81.61%	78.07%	77.95%
Boulevard	Prec.	40.02%	43.79%	62.25%	65.21%	33.59%	70.57%	61.13%	61.90%
	Rec.	83.21%	79.77%	62.96%	75.82%	77.64%	58.73%	79.54%	77.89%
Sidewalk	Prec.	42.71%	35.99%	53.20%	89.86%	49.89%	64.25%	80.82%	88.16%
	Rec.	58.12%	51.06%	28.57%	50.49%	52.49%	64.65%	61.25%	57.92%
Traffic	Prec.	58.61%	52.58%	68.33%	64.57%	44.31%	68.88%	66.10%	70.44%
	Rec.	76.47%	73.68%	56.64%	76.76%	85.89%	87.63%	78.46%	72.72%

Table 3.4: Intermittent Object Motion Precision and Recall

		Stauffer et al. [2]	Zivkovic [3]	KaewTraKulPong et al [4]	Evangelio et al. [5]	ELgammal et al. [6]	Nonaka et al. [7]	AGM	AGM+SD
AbondonedBox	Prec.	65.52%	62.14%	67.75%	66.53%	53.73%	79.67%	67.41%	72.15%
	Rec.	45.74%	45.64%	39.51%	42.23%	87.45%	40.54%	45.18%	42.60%
Parking	Prec.	75.82%	73.82%	65.57%	78.93%	61.53%	92.49%	77.92%	81.15%
	Rec.	74.09%	69.87%	36.60%	65.80%	26.77%	46.45%	73.08%	70.22%
StreetLight	Prec.	89.16%	92.47%	99.69%	92.40%	48.01%	78.52%	97.56%	98.79%
	Rec.	32.25%	33.94%	23.57%	33.19%	31.46%	20.76%	30.33%	28.61%
Sofa	Prec.	85.92%	89.25%	96.93%	94.02%	85.72%	87.34%	92.52%	95.70%
	Rec.	51.62%	51.41%	32.60%	57.75%	51.91%	43.58%	59.90%	51.63%
Tramstop	Prec.	68.54%	56.36%	71.89%	68.20%	18.91%	97.01%	68.23%	75.44%
	Rec.	33.74%	59.34%	15.79%	39.88%	30.09%	38.64%	42.41%	32.62%
WinterDriveway	Prec.	16.32%	13.41%	15.37%	19.49%	8.64%	54.94%	25.68%	25.79%
	Rec.	71.10%	67.84%	60.50%	61.93%	74.40%	80.75%	60.15%	60.08%

Table 3.5: Shadows Precision and Recall

		Stauffer et al. [2]	Zivkovic [3]	KaewTraKulPong et al [4]	Evangelio et al. [5]	ELgammal et al. [6]	Nonaka et al. [7]	AGM	AGM+SD
Backdoor	Prec.	50.81%	50.94%	94.33%	60.84%	82.26%	89.94%	66.43%	94.89%
	Rec.	85.32%	82.34%	75.39%	88.56%	86.74%	99.07%	89.73%	76.82%
Bungalows	Prec.	71.97%	71.58%	63.36%	72.95%	70.73%	81.27%	72.62%	74.59%
	Rec.	89.41%	87.40%	58.29%	94.88%	83.31%	89.36%	94.89%	93.13%
BusStation	Prec.	88.28%	88.32%	93.74%	88.2%	84.02%	82.93%	88.50%	92.88%
	Rec.	73.40%	71.04%	45.19%	86.05%	74.37%	63.07%	90.11%	89.25%
Cubicle	Prec.	10.82%	55.11%	88.26%	67.22%	55.31%	85.86%	66.16%	87.19%
	Rec.	30.12%	78.61%	70.88%	92.64%	83.48%	77.28%	92.48%	90.24%
PeopleInShade	Prec.	84.05%	84.11%	84.66%	83.67%	84.21%	78.66%	84.61%	87.15%
	Rec.	94.39%	92.69%	79.47%	96.49%	96.01%	69.62%	96.55%	95.21%
CopyMachine	Prec.	79.42%	83.84%	90.29%	84.07%	83.06%	75.99%	88.54%	92.19%
	Rec.	53.92%	54.14%	50.15%	56.19%	88.29%	33.41%	57.43%	55.70%

Table 3.6: Thermal Precision and Recall

		Stauffer et al. [2]	Zivkovic [3]	KaewTraKulPong et al [4]	Evangelio et al. [5]	ELgammal et al. [6]	Nonaka et al. [7]	AGM	AGM+SD
Corridor	Prec.	80.75%	83.93%	96.20%	85.84%	88.06%	89.55%	90.72%	93.90%
	Rec.	82.52%	83.26%	65.71%	84.68%	83.20%	56.00%	89.24%	87.17%
Library	Prec.	84.76%	81.76%	96.68%	93.86%	97.14%	96.35%	94.66%	94.81%
	Rec.	28.00%	28.68%	24.03%	30.23%	92.20%	8.07%	31.33%	31.05%
Park	Prec.	80.66%	85.07%	99.95%	92.57%	85.85%	80.42%	88.14%	89.47%
	Rec.	63.96%	59.30%	16.24%	39.98%	60.81%	89.03%	64.00%	62.28%
DiningRoom	Prec.	93.37%	92.31%	98.54%	94.03%	88.42%	95.55%	93.74%	94.44%
	Rec.	70.21%	69.43%	43.16%	77.45%	75.74%	40.11%	79.57%	79.07%
Lakeside	Prec.	93.04%	92.23%	94.10%	96.86%	89.21%	96.36%	97.48%	98.53%
	Rec.	39.88%	36.41%	20.59%	35.80%	24.29%	14.12%	40.84%	39.78%

Table 3.7: Overall Precision and Recall

	Stauffer et al. [2]	Zivkovic [3]	KaewTraKulPong et al [4]	Evangelio et al. [5]	ELgammal et al. [6]	Nonaka et al. [7]	AGM	AGM+SD
Prec.	70.12%	70.79%	82.28%	78.12%	68.43%	76.63%	81.06%	83.11%
Rec.	71.08%	69.64%	50.72%	70.73%	74.42%	65.07%	79.18%	73.07%

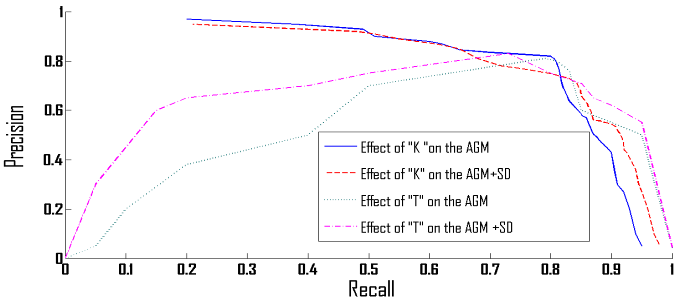


Figure 3.8: Precision and Recall curves for : The AGM and the $AGM + SD$ when varying T and K.

Chapter 4

Finite asymmetric generalized Gaussian mixture models learning for infrared object detection

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 chapter, 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.

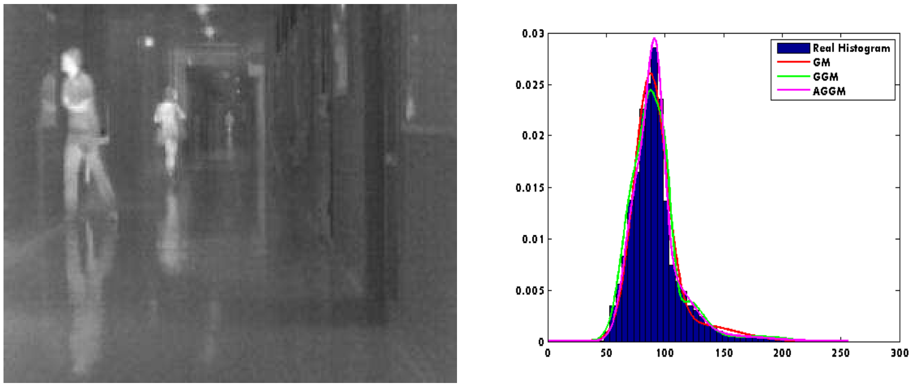
4.1 Introduction

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 minutes when engaged in monotonous and repetitive activities [153]. 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 [154]. In controlled or indoor environments (i.e. airports, warehouses, and production plants) monitoring is easier to implement as it doesn't depend on weather changes [129, 155]. Uncontrolled environment is used to refer to outdoor scenes where illumination and temperature changes occur frequently, and where various atmospheric conditions can be observed [129, 156].

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 [157]. However, under low illumination sensing in visible spectrum becomes infeasible [158]. 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 [159]. 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 [160, 161]. Some researchers decided to use both visible and infrared images together in order to increase the efficiency of surveillance systems [162, 163]. It is widely known in the field of image fusion that the combination 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. Figure 4.1 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 [20, 21]. 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 chapter, 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 Figure 4.1(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).



(a) (b)

Figure 4.1: (a) IR image, (b) Real and estimated (using GM, GGM and AGGM) histograms for the IR image.

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) [29], minimum description length (MDL) [30] and Laplace empirical criterion (LEC) [14]. 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 [36]. Despite their formal appeal, MCMC methods are too computationally demanding, therefore can't 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 [31] 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.

The rest of this chapter is organized as follows. Section 4.2 describes the AGGM model and gives a complete learning algorithm. In section 4.3, we assess the performance of the new model for pedestrian detection and multiple-target tracking; while comparing it to other models. Our last section is devoted to the conclusion and some perspectives.

4.2 Finite Asymmetric Generalized Gaussian Mixture Model

4.2.1 The Finite Mixture Model

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

$$p(\vec{X}|\Theta) = \sum_{j=1}^{M} p_j p(\vec{X}|\xi_j) \tag{4.1}$$

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, Θ= $\{p_1,\dots,p_M,\xi_1,\dots,\xi_M\}$ is the complete set of parameters fully characterizing the mixture, $M \geq 1$ is number of components in the mixture. For the AGGM, each component density $p(\vec{X}|\xi_j)$ is an AGGD:

$$p(\vec{X}|\xi_j) \ = \prod_{k=1}^{d} \frac{\beta_{jk}\left[\frac{\Gamma(3/\beta_{jk})}{\Gamma(1/\beta_{jk})}\right]^{1/2}}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})\Gamma(1/\beta_{jk})} \begin{cases} \exp\left[-A(\beta_{jk})\left(\frac{\mu_{jk}-X_k}{\sigma_{l_{jk}}}\right)^{\beta_{jk}}\right] & \text{if } X_k < \mu_{jk} \\ \exp\left[-A(\beta_{jk})\left(\frac{X_k-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}}\right] & \text{if } X_k \geq \mu_{jk} \end{cases} \tag{4.2}$$

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 $\vec{\mu}_j$ = (μ_{j1},...,μ_{jd}), $\vec{\sigma}_{lj}$ = ($\sigma_{l_{j1}}$,...,$\sigma_{l_{jd}}$), and $\vec{\sigma}_{rj}$ = ($\sigma_{r_{j1}}$,...,$\sigma_{r_{jd}}$) are the mean, the left standard deviation, and the right standard deviation of the d-dimensional AGGD, respectively. The parameter $\vec{\beta}_j$ = (β_{j1},...,β_{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 $\mathcal{X}$= ($\vec{X}_1$,...,$\vec{X}_N$) 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:

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N}\sum_{j=1}^{M} p(\vec{X}_i|\xi_j)p_j \tag{4.3}$$

where the set of parameters of the mixture with M classes is defined by Θ = ($\vec{\mu}_1$,..., $\vec{\mu}_M$, $\vec{\beta}_1$,..., $\vec{\beta}_M$, $\vec{\sigma}_{l_1}$,..., $\vec{\sigma}_{l_M}$, $\vec{\sigma}_{r_1}$,..., $\vec{\sigma}_{r_M}$, p_1,..., 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:

$$p(\mathcal{X}, Z|\Theta) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left(p(\vec{X}_i|\xi_j)p_j\right)^{Z_{ij}} \tag{4.4}$$

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:

$$L(\Theta, Z, \mathcal{X}) = \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{ij}\log\left(p(\vec{X}_i|\xi_j)p_j\right) \tag{4.5}$$

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

$$\hat{Z}_{ij} = p(j|\vec{X}_i) = \frac{p(\vec{X}_i|\xi_j)p_j}{\sum_{j=1}^{M} p(\vec{X}_i|\xi_j)p_j} \tag{4.6}$$

Using equation 4.6 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:

$$\Phi(\Theta, Z, \mathcal{X}, \Lambda) = \sum_{i=1}^{N}\sum_{j=1}^{M} Z_{ij} \log\Big(p(\vec{X}_i|\xi_j)p_j\Big) + \Lambda(1 - \sum_{j=1}^{M} p_j) \tag{4.7}$$

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

$$\hat{p}_j = \frac{1}{N}\sum_{i=1}^{N} p(j|\vec{X}_i) \tag{4.8}$$

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 $k = 1, \dots, d$:

$$\sum_{i=1, X_{ik}<\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})}{{\sigma_{l_{jk}}}^{\beta_{jk}}}(\mu_{jk} - X_{ik})^{\beta_{jk}-1} - \sum_{i=1, X_{ik}\geq\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})}{{\sigma_{r_{jk}}}^{\beta_{jk}}}(X_{ik} - \mu_{jk})^{\beta_{jk}-1} = 0 \tag{4.9}$$

$$\begin{aligned} & \sum_{i=1, X_{ik}<\mu_{jk}}^{N} Z_{ij}A(\beta_{jk})(\frac{\mu_{jk} - X_{ik}}{\sigma_{l_{jk}}})^{\beta jk}\left[\left(\frac{3\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{2\beta_{jk}}\right) - \log\left(\frac{\mu_{jk} - X_{ik}}{\sigma_{l_{jk}}}\right)\right] \\ + & \sum_{i=1, X_{ik}\geq\mu_{jk}}^{N} Z_{ij}A(\beta_{jk})(\frac{X_{ik} - \mu_{jk}}{\sigma_{r_{jk}}})^{\beta jk}\left[\left(\frac{3\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{2\beta_{jk}}\right) - \log\left(\frac{X_{ik} - \mu_{jk}}{\sigma_{r_{jk}}}\right)\right] \\ + & \sum_{i=1}^{N} Z_{ij}\left[\frac{1}{\beta_{jk}} - \frac{3}{2}\left(\frac{\Psi(3/\beta_{jk}) - \Psi(1/\beta_{jk})}{\beta_{jk}^2}\right)\right] = 0 \end{aligned} \tag{4.10}$$

$$\sum_{i=1, X_{ik}<\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})\beta_{jk}}{\sigma_{l_{jk}}}\left(\frac{\mu_{jk} - X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta jk} - \sum_{i=1}^{N}\frac{Z_{ij}}{\sigma_{l_{jk}} + \sigma_{r_{jk}}} = 0 \tag{4.11}$$

$$\sum_{i=1, X_{ik}\geq\mu_{jk}}^{N} Z_{ij}\frac{A(\beta_{jk})\beta_{jk}}{\sigma_{r_{jk}}}\left(\frac{X_{ik} - \mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta jk} - \sum_{i=1}^{N}\frac{Z_{ij}}{\sigma_{l_{jk}} + \sigma_{r_{jk}}} = 0 \tag{4.12}$$

where $\Psi(x) = \frac{\partial log[\Gamma(x)]}{\partial x}$. It is easy to notice that the equations from 4.9 to 4.12 related to all AGGD parameters are non linear. Thus, we decided to use the Newton-Raphson method to estimate these parameters:

$$\hat{\mu}_{jk} \simeq \mu_{jk} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial\mu_{jk}^2}\right)^{-1}\left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial\mu_{jk}}\right)\right] \tag{4.13}$$

$$\hat{\beta}_{jk} \simeq \beta_{jk} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \beta_{jk}^2} \right)^{-1} \left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \beta_{jk}} \right) \right] \tag{4.14}$$

$$\hat{\sigma}_{l_{jk}} \simeq \sigma_{l_{jk}} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk}}^2} \right)^{-1} \left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \sigma_{l_{jk}}} \right) \right] \tag{4.15}$$

$$\hat{\sigma}_{r_{jk}} \simeq \sigma_{r_{jk}} - \left[\left(\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk}}^2} \right)^{-1} \left(\frac{\partial L(\Theta, Z, \mathcal{X})}{\partial \sigma_{r_{jk}}} \right) \right] \tag{4.16}$$

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 C.

4.2.3 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) [31, 140]:

$$MessLen \approx -\log(p(\Theta)) - L(\Theta, Z, \mathcal{X}) + \frac{1}{2}\log|F(\Theta)| + \frac{N_p}{2} - \frac{1}{2}\log(12) \tag{4.17}$$

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 $|F(\Theta)|$.

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:

$$p(\Theta) = p(\mu)p(\beta)p(\sigma_l)p(\sigma_r)p(P) \tag{4.18}$$

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

$$p(P) = \frac{\Gamma(\sum_{j=1}^{M} \eta_j)}{\prod_{j=1}^{M} \Gamma(\eta_j)} \prod_{j=1}^{M} p_j^{\eta_j - 1} \tag{4.19}$$

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 $p_1 + \ldots + p_M = 1$. This prior is represented by

$$p(P) = (M - 1)! \tag{4.20}$$

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

$$p(\mu) = \prod_{j=1}^{M} \prod_{k=1}^{d} p(\mu_{jk}) = \prod_{k=1}^{d} \frac{1}{(\sigma_{l_k} + \sigma_{r_k})^M} \tag{4.21}$$

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

$$p(\beta) = \prod_{j=1}^{M} \prod_{k=1}^{d} p(\beta_{jk}) = \frac{1}{h^{Md}} \tag{4.22}$$

It is known that ($0 \leq \sigma_{l_{jk}} \leq \sigma_{l_k}$) and ($0 \leq \sigma_{r_{jk}} \leq \sigma_{r_k}$) 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 $\sigma_{r_{jk}}$

$$p(\sigma_l) = \prod_{j=1}^{M} \prod_{k=1}^{d} p(\sigma_{l_{jk}}) = \prod_{k=1}^{d} \frac{1}{\sigma_{l_k}^M} \tag{4.23}$$

$$p(\sigma_r) = \prod_{j=1}^{M} \prod_{k=1}^{d} p(\sigma_{r_{jk}}) = \prod_{k=1}^{d} \frac{1}{\sigma_{r_k}^M} \tag{4.24}$$

Finally, by replacing the priors in equation 4.18 by the expressions in equations 4.20, 4.21, 4.22, 4.23,4.24, we get

$$p(\Theta) = \frac{(M-1)!}{h^{Md}} \prod_{k=1}^{d} \frac{1}{\sigma_{l_k}^M \sigma_{r_k}^M (\sigma_{l_k} + \sigma_{r_k})^M} \tag{4.25}$$

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:

$$|F(\Theta)| = |F(P)| \prod_{j=1}^{M} |F(\vec{\mu}_j)||F(\vec{\beta}_j)||F(\vec{\sigma}_{l_j})||F(\vec{\sigma}_{r_j})| \tag{4.26}$$

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:

$$|F(P)| = \frac{N^{M-1}}{\prod_{j=1}^{M} p_j} \tag{4.27}$$

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

$$F(\vec{\mu}_j)_{k_1,k_2} = \frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \mu_{jk_1} \partial \mu_{jk_2}} \tag{4.28}$$

$$F(\vec{\beta}_j)_{k_1,k_2} = \frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \beta_{jk_1} \partial \beta_{jk_2}} \tag{4.29}$$

$$F(\vec{\sigma}_{l_j})_{k_1,k_2} = \frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \sigma_{l_{jk_1}} \partial \sigma_{l_{jk_2}}} \tag{4.30}$$

$$F(\vec{\sigma}_{r_j})_{k_1,k_2} = \frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \sigma_{r_{jk_1}} \partial \sigma_{r_{jk_2}}} \tag{4.31}$$

where (k_1 ,k_2) $\in$ (1, $\dots$,d). Note that $\mathcal{X}_j$= ($\vec{X}_l$,$\dots$,$\vec{X}_{l+n_j-1}$) represents the data in class j after classifying all the data $\mathcal{X}$ using the maximum a posteriori probability defined by Eq. 4.6. Using appendix C to compute the derivatives in equations 4.28, 4.29, 4.30, 4.31, we obtain

$$\begin{aligned} |F(\vec{\mu}_j)| &= \prod_{k=1}^{d} A(\beta_{jk})\beta_{jk}(\beta_{jk}-1)\Big[\sum_{i=l,X_{ik}<\mu_{jk}}^{l+n_j-1} \frac{(\mu_{jk}-X_{ik})^{\beta_{jk}-2}}{{\sigma_{l_{jk}}}^{\beta_{jk}}} \\ &+ \sum_{i=l,X_{ik}\geq\mu_{jk}}^{l+n_j-1} \frac{(X_{ik}-\mu_{jk})^{\beta_{jk}-2}}{{\sigma_{r_{jk}}}^{\beta_{jk}}}\Big] \qquad j=1,\ldots,M \end{aligned} \tag{4.32}$$

$$\begin{aligned} |F(\vec{\beta}_j)| = &\prod_{k=1}^{d}\sum_{i=l}^{l+n_j-1}\left[\frac{1}{\beta_{jk}^2}+\frac{3\Psi'(1/\beta_{jk})}{2\beta_{jk}^4}+3\frac{\Psi(1/\beta_{jk})-\Psi(3/\beta_{jk})}{\beta_{jk}^3}-\frac{9\Psi'(3/\beta_{jk})}{2\beta_{jk}^4}\right] \\ + &\ A(\beta_{jk})\sum_{i=l,X_{ik}<\mu_{jk}}^{l+n_j-1}\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta_{jk}}\Bigg[\left(\frac{9\Psi'(3/\beta_{jk})-\Psi'(1/\beta_{jk})}{2\beta_{jk}^3}+\frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}^2}\right) \\ + &\left(\frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}}-\log\left[\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right]\right)^2\Bigg] \\ + &\ A(\beta_{jk})\sum_{i=l,X_{ik}\geq\mu_{jk}}^{l+n_j-1}\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}}\Bigg[\left(\frac{9\Psi'(3/\beta_{jk})-\Psi'(1/\beta_{jk})}{2\beta_{jk}^3}+\frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}^2}\right) \\ + &\left(\frac{3\Psi(3/\beta_{jk})-\Psi(1/\beta_{jk})}{2\beta_{jk}}-\log\left[\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right]\right)^2\Bigg] \qquad j=1,\ldots,M \end{aligned} \tag{4.33}$$

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

$$\begin{aligned} |F(\vec{\sigma}_{l_j})| &= \prod_{k=1}^{d}\Big[\sum_{i=l}^{l+n_j-1}\frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} \\ &- A(\beta_{jk})\beta_{jk}(\beta_{jk}+1)\sum_{i=l,X_{ik}<\mu_{jk}}^{l+n_j-1}\frac{1}{\sigma_{l_{jk}}^2}\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta jk}\Big] \qquad j=1,\ldots,M \end{aligned} \tag{4.34}$$

$$\begin{aligned} |F(\vec{\sigma}_{r_j})| &= \prod_{k=1}^{d}\Big[\sum_{i=l}^{l+n_j-1}\frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} \\ &- A(\beta_{jk})\beta_{jk}(\beta_{jk}+1)\sum_{i=l,X_{ik}\geq\mu_{jk}}^{l+n_j-1}\frac{1}{\sigma_{r_{jk}}^2}\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta jk}\Big] \qquad j=1,\ldots,M \end{aligned} \tag{4.35}$$

4.2.4 Complete AGGM Learning Algorithm

In the following steps, we summarize the algorithm used for the learning of our AGGM:
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.4.6.
            (b) The Maximization step using Eqs.(4.13-4.16).
        3. Calculate the associated message length using Eq.(4.17).
    }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 [14].

4.3 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.

4.3.1 Pedestrian detection

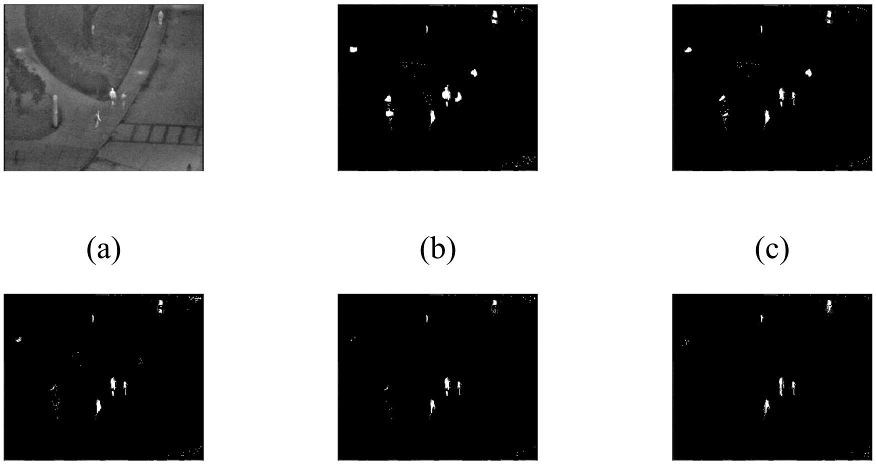
Pedestrian detection is an essential task in intelligent automatic video surveillance systems. In recent years, numerous approaches have been proposed in order to detect, track, and recognize human activity [164]. However, pedestrian detection still remains an active research area in computer vision. 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 chapter, 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 [158]. 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 4.2.4 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) [65], the GGM with MML [21], and the infinite generalized Gaussian mixture model (IGGM) [139]. We have used the OSU

Thermal Pedestrian Database [165] 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. Figure 4.2 shows an image taken in the presence of Haze for five pedestrians.

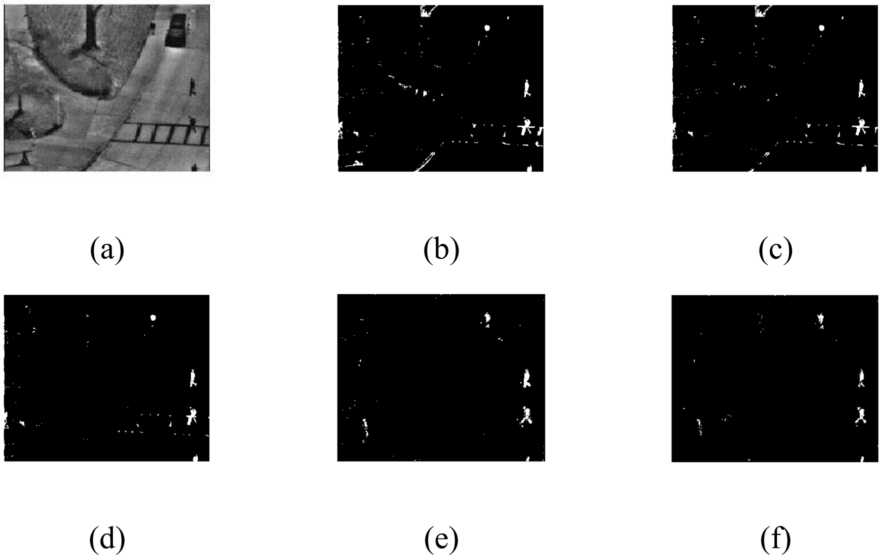


(a) (b) (c)

(d) (e) (f)

Figure 4.2: (a) IR image for five pedestrians in the presence of Haze, (b) GM, (c) IGM, (d) GGM, (e) IGGM, and (f) AGGM.

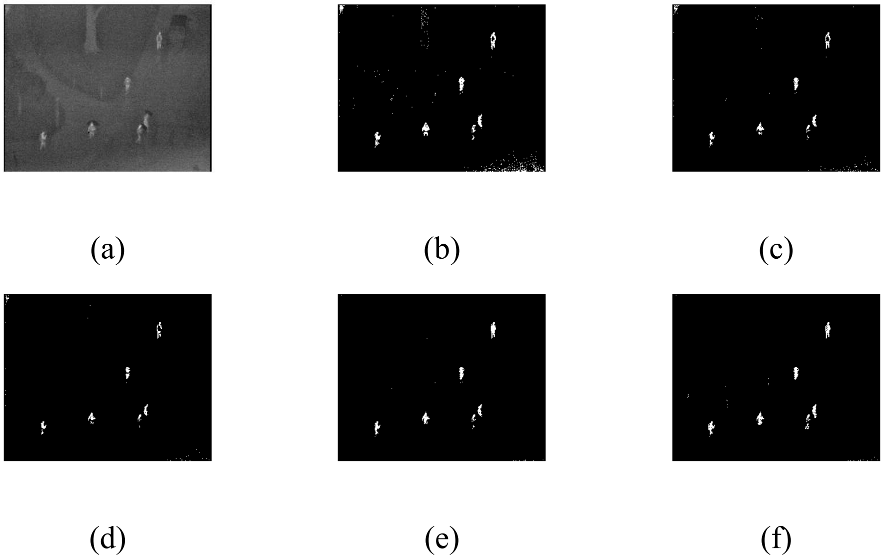
Comparing the four outputs together we can notice that the GM and the IGM both have wrongly modeled the two white street parts and the street lamp, also the pedestrian behind the tree wasn't correctly represented. As for the GGM, it has taken the street lamp into consideration, and failed to represent the pedestrian behind the tree. Both IGGM and our method were able to recognize the five pedestrians without any problem. Figure 4.3 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. Figure 4.4 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 have a quantitative evaluation of the performance, we have used two well-known metrics, precision and recall, to quantify how well



(a) (b) (c)

(d) (e) (f)

Figure 4.3: (a) IR Image for six pedestrians on a very cloudy day, (b) GM, (c) IGM, (d) GGM, (e) IGGM, and (f) AGGM.



(a) (b) (c)

(d) (e) (f)

Figure 4.4: (a) IR Image for six pedestrians on a rainy day, (b) MoG, (c) IMoG, (d) MoGG, (e) IMoGG, and (f) MoAGG.

each algorithm works in classifying the data [152]. Precision (Eq. 4.36) represents the percentage of detected true positives (pedestrians) to the total number of items detected by the algorithm. Recall (Eq. 4.37) is the percentage of number of detected true positives by the algorithm to the total number of true positives in the dataset:

$$Precision = \frac{TP}{TP + FP} \tag{4.36}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.37}$$

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 4.1 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 [165] and [158]. From this table we can deduce that our model is capable of detecting pedestrians efficiently. According to quantitative and analytical analysis, it's clear that our method was able to identify all pedestrians in each image even under polarity switch phenomenon. From table 4.1 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 can't be applied efficiently for online applications like pedestrian detection. Furthermore, we can notice that the method introduced in [158] 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 method to the two stage approach introduced in [165] we found the our method has a higher recall which means that it is more effective in identifying pedestrians.

Table 4.1: Precision and Recall

	Davis et al. [165]	Dai et al. [158]	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%

4.3.2 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. [166] introduced another algorithm that performs a gradient-descent search on the region of interest in images but it wasn't 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 [167], 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 colour 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 opens the way to video systems combining thermal and colour 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 [168]. 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. 4.5.

Our method can be divided into three main components: Detection, Low level tracking, and

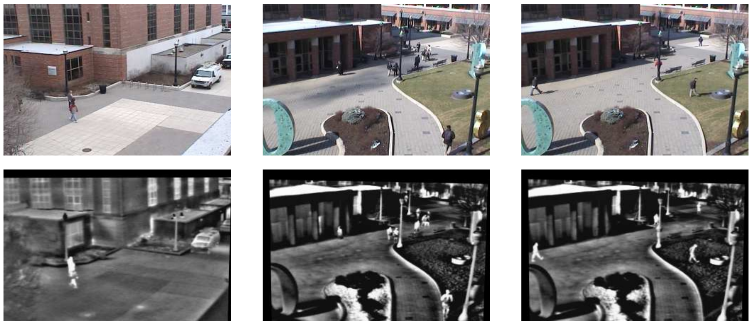


Figure 4.5: Some sample frames from the OSU Color-Thermal database.

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 [167] to deal with fusion of both visible and infrared inputs.

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 [3, 169, 170] (a complete detailed survey can be found in [134]). 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 [2, 127]. However, these methods have some drawbacks. Modeling the background using the GM implies the assumption that the background and foreground distributions are Gaussians which isn't always the case for uncontrolled environments as argued by [138].

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 [127] 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 colour 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

$$A = argmin_a \sum_{j=1}^{a} p_j > T, \tag{4.38}$$

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 doesn't 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

$$p_j^{(t+1)} = p_j^{(t)} + B_{(t)}[p(j|\vec{X}^{(t+1)}) - p_j^{(t)}] \tag{4.39}$$

$$\xi_j^{(t+1)} = \xi_j^{(t)} + B_{(t)}\left[p(j|\vec{X}^{(t+1)})\frac{\partial L(\Theta, Z, \mathcal{X}^{(t+1)})}{\partial \xi_j}\right] \tag{4.40}$$

where $B_{(t)}$ represents any sequence of positive numbers that decreases to zero. The derivatives in Eq. 4.40 are given in appendix C. 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, . . ., d: set $\sigma_{l_{1k}} = \sigma_{r_{1k}} = 0.2$, $\mu_{1k} = X_k^{(0)}$, and $\beta_{jk} = 2$.

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 equations 4.39 and 4.40.

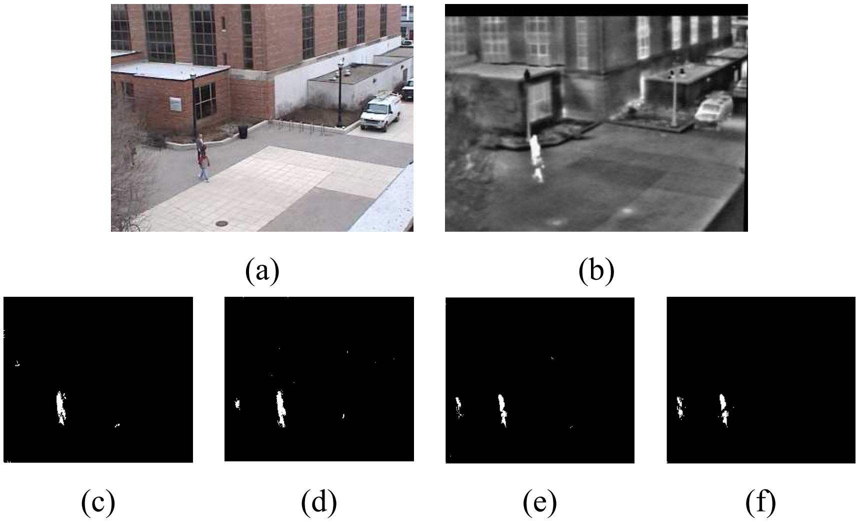
   (c) Order the pixel mixture components by the values of $\frac{p_j}{||\vec{\sigma}_{l_j}||+||\vec{\sigma}_{r_j}||}$.

   (d) Extract the foreground objects by identifying the first A components that represent the background using equation 4.38 .

In order to evaluate our algorithm for foreground segmentation we compared it with the well-known GM method introduced in [2], the GGM [21], and the IGGM introduced in [139]. Figures 4.6, 4.7, and 4.8 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 [1]:

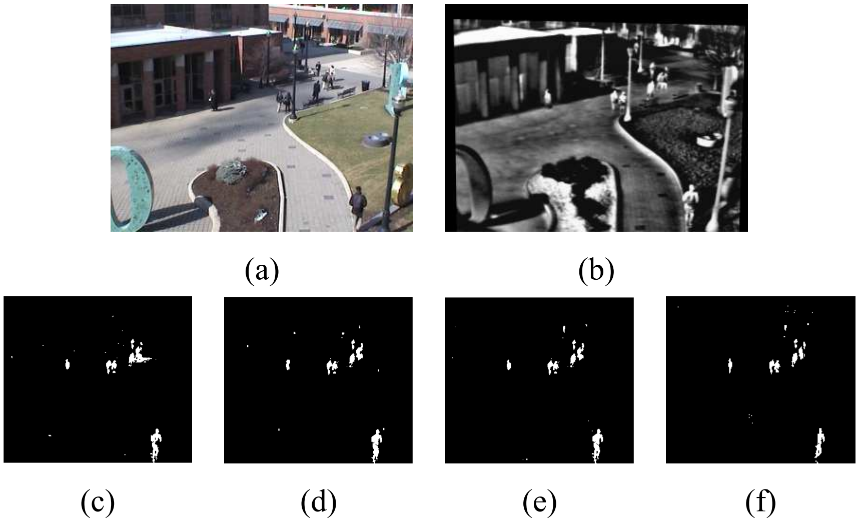
$$D = cD_{color} + hD_{hist} + mD_{motion} \tag{4.41}$$

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 4.2). 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. It is noteworthy that the computational time of our approach for adaptive background subtraction is around 11 frames per second for frames with 320 × 240 pixels resolution running on Core i7-2.4 GHz processor. We can notice from figures 4.6, 4.7, and 4.8 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.


(a) (b)

(c) (d) (e) (f)

Figure 4.6: (a) Color image, (b) IR image, (c) GM, (d) GGM, (e) IGGM, and (f) AGGM.

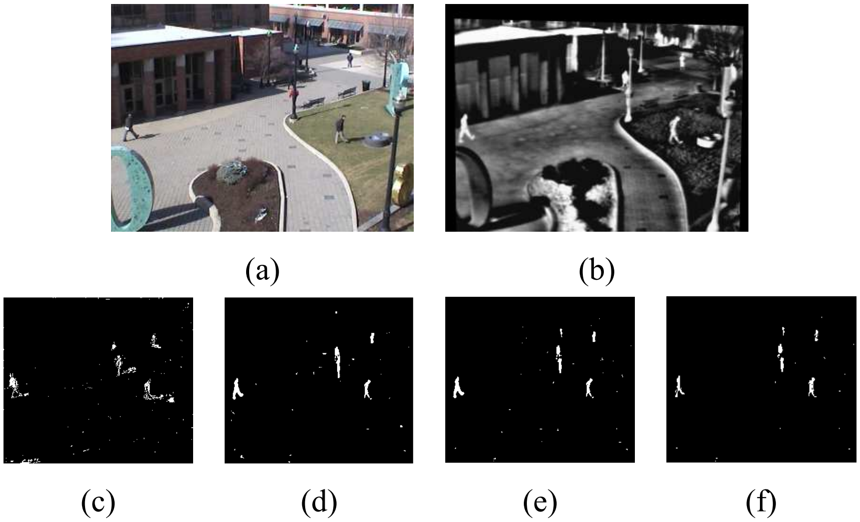

(a) (b)

(c) (d) (e) (f)

Figure 4.7: (a) Color image, (b) IR image, (c) GM, (d) GGM, (e) IGGM, and (f) AGGM.

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 [167]. We have chosen this method as it is good in identifying each



(a) (b)

(c) (d) (e) (f)

Figure 4.8: (a) Color image, (b) IR image, (c) GM, (d) GGM, (e) IGGM, and (f) AGGM.

Table 4.2: Combined performance measure [1].

GM	GGM	IGGM	AGGM
0.43	0.31	0.26	0.28

object and will be useful when dealing with object collision. Thus, the l-observed blob at time t is represented by $z_l^t = (x_l^t, y_l^t, h_l^t, w_l^t, \theta_l^t)$, 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 [167] in order to predict the target state. Then, the target state for an ellipse representation is represented by $z_l^t = (x_l^t, \dot{x}_l^t, y_l^t, \dot{y}_l^t, h_l^t, \dot{h}_l^t, w_l^t, \dot{w}_l^t, \theta_l^t)$ 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's 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). Figure 4.9 shows an example of tracking using the ellipse centroid metric.

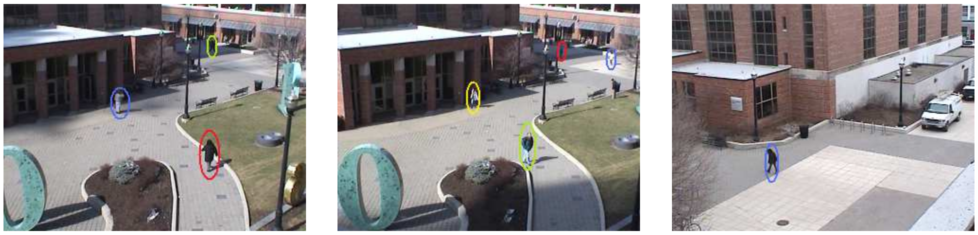


Figure 4.9: Low level tracking example using the ellipse centroid metric.

High Level Tracker

The Kalman filter introduced above is able to predict the state for multiple targets. However, it can not 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 deals with object appearance. Collins et al. [171] 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:

$$h = w_1 \times I + w_2 \times R + w_3 \times G + w_4 \times B \tag{4.42}$$

where $w_a \in$ (-1,0,1); a = (1, . . . , 4). Equation 4.42 represents linear combinations composed of integer coeff cients between -1 and 1. The total number of such candidates would be 3^4, however, by excluding redundant coeff cients 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 i^{th} feature can be computed as:

$$L^i(k) = \log \frac{\max(p_k^i, \in)}{\max(q_k^i, \in)} \tag{4.43}$$

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 k^{th} bin of the i^{th} feature of the target and background histogram, respectively. Then, features are evaluated according to the variance-ratio of the log-likelihood:

$$VR^i(k) = var \frac{\max(p_k^i, \in)}{\max(q_k^i, \in)} \tag{4.44}$$

Thus, features are ranked according to their variance ratio (the higher is the better). In this application, opposed to the method of [171], 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 i^{th} feature at time t can be computed by:

$$m_t^i = m_{t-1}^i + \frac{1}{n_i}(p_t^i - m_t^i) \tag{4.45}$$

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.

$$\mu_t^i = \mu_{t-1}^i + \frac{1}{n-1}(d_{B_t}^i - \mu_t^i) \tag{4.46}$$

$$\sigma_t^2 = \frac{n-3}{n-2}\sigma_{t-1}^2 + (n-1)(\mu_t^i - \mu_{t-1}^i) \tag{4.47}$$

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. Figure 4.10 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. Figure 4.11 shows some collision detection examples.

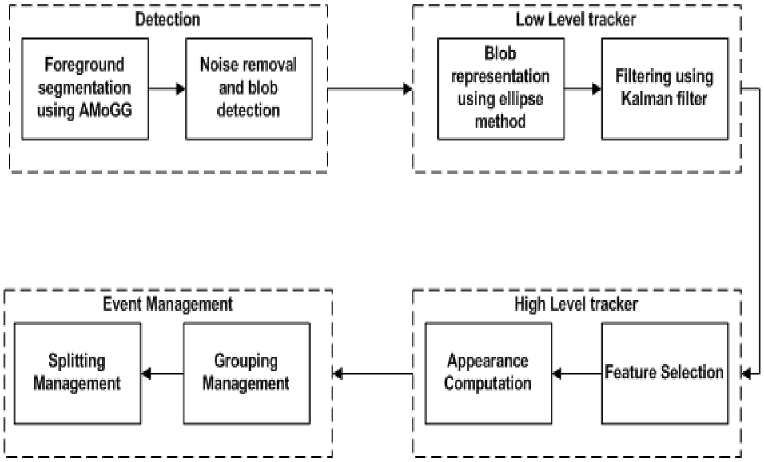
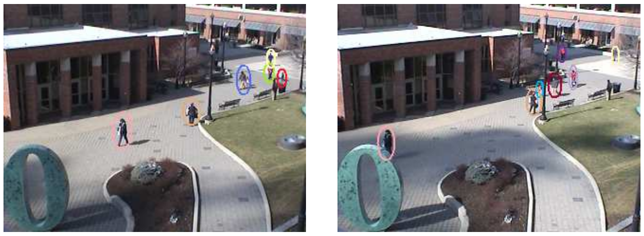


Figure 4.10: System architecture.



(a) (b)

Figure 4.11: Collision detection example. (a) Object before collision (b) Grouping and splitting events identified.

Table 4.3: Multiple target tracking results.

		Total	Our Method		Rowe et al. [167]	
			Visible	Fusion	Visible	Fusion
1^{st} 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
2^{nd} 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
3^{rd} 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
4^{th} 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
5^{th} 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
6^{th} 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

Results

Our approach performance has been evaluated using the OSU Color-Thermal Database [168]. 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. In order to validate our method, we have used the method of Rowe et al. [167]. Table 4.3 represents the performance of the two methods with only visible camera as input and with both thermal and visible used as input. 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.

4.4 Discussion

In this chapter, 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 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.

Chapter 5

Simultaneous high-dimensional clustering and feature selection using Asymmetric mixture models

Finite mixture models are broadly applied in a wide range of applications concerning density estimation and clustering due to their sound mathematical basis and to the interpretability of their results. Indeed, they permit the incorporation of domain knowledge which allows to provide better insight into the nature of the clusters and then uncovers application-specific desirable patterns that the practitioner is looking for. However, most of the works done on mixture models, when applied to computer vision tasks, assume that per-components data follow a mixture of Gaussians which may not hold as data are generally non Gaussian. The effect of the Gaussian mixture is analogous to the deployment of Euclidean or Mahalanobis type distances for discrimination purposes. Thus, this mixture cannot be applied efficiently in several applications involving asymmetric shapes. In this chapter, we overcome this problem by using two asymmetric mixture models namely AGM and AGGM. Both distributions can change their shapes to model non-symmetrical and heavy tailed real world data which make them a good choice for modeling data with outliers. Modern computer vision applications generally generate complex high-dimensional data and usually, some features are noisy, redundant, or uninformative which may affect the speed and also compromise the accuracy of the used learning algorithm. Therefore, this chapter addresses also the problem of unsupervised feature selection. In addition, we propose two approaches for learning the resulting statistical framework. The first approach is based on the minimization of a message length objective and the second one considers rival penalized competitive learning. Our extensive simulations and experiments involving two challenging tasks namely visual scene categorization and facial expression recognition indicate that the method developed in this chapter is efficient and has merits.

5.1 Introduction

Mathematical models in general and statistical approaches in particular have been widely used for the development of useful computer vision, signal and image processing algorithms [172–174]. Many of these approaches are based on finite mixture models (i.e. a weighted sum of distributions) which have been the topic of extensive research in the past [14] and have been applied in several applications such as content-based images categorization and retrieval [175]. In the field of finite mixtures, Gaussian mixture model (GMM) has been widely considered, studied and used [176–178]. However, the Gaussian assumption is rarely justified and met in practice [179] and this is especially true in the case of natural images as shown by several studies and research works [180]. Gaussian density has several drawbacks such as its symmetry around the mean and the rigidity of its shape, which prevent it from having a good approximation to data with outliers. Therefore, we suggest the consideration of the AGD and AGGD capable of modeling heavy and short tailed data [46, 181]. 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.

Concerning parameters estimation, the most popular approach is perhaps the one based on the maximization of the likelihood function through the expectation maximization (EM) framework [182]. It is well-known that the EM algorithm needs an appropriate predefined number of clusters. Therefore, in the past decades, a lot of research has been devoted to the automatic selection of the number of clusters which best describe a given data set and a lot of selection criteria have been proposed such as Akaike's information criterion (AIC), minimum description length (MDL), Laplace empirical criterion (LEC), and minimum message length (MML) [14, 26]. In particular, the MML criterion has been shown to outperform the majority of existing selection criteria. Thus, we shall consider it in this work by comparing it to another approach based on the rival penalized EM (RPEM) algorithm which has received a lot of attention [33, 183]. The RPEM is able to automatically select an appropriate number of densities by fading out the redundant densities from a density mixture which can save the computing time. Thus, we propose to use the RPEM algorithm to perform model selection and parameters learning together in a single step.

Modern computer vision application generate high-dimensional vectors. Handling data defined in high-dimensional feature spaces is a difficult problem [184]. Theoretically, the more information we have about each pattern, the better a learning algorithm is expected to perform. However, in many cases, some features can be noisy or uninformative which can degrade clustering efficiency [185]. Thus, in order to achieve a good performance of data modeling, irrelevant features have to be discarded. An accurate feature selection (FS), the task of choosing the best feature subset, allows to improve understandability, scalability, and accuracy of the resulting learned models that generalize better to unseen data. Indeed, several recent studies have shown that selecting relevant features allows more meaningful modeling results [186, 187]. However, the problem is challenging especially in unsupervised settings because of the absence of class labels that could guide the selection process [188]. Therefore, there have been only few feature selection techniques that have been applied in mixture-based clustering [41, 43, 78] since the aim is to identify simultaneously two inter-related unknowns that are optimal feature subset and optimal number of clusters. In this article, and following recent approaches (see, for instance [41, 43, 78]), we perform unsupervised feature selection approach by casting it as an estimation problem, thus avoiding any combinatorial search. For each feature, we associate a relevance weight which measures the degree of its dependence on class labels.

The remainder of this chapter is organized as follows: After the introduction we describe our feature selection approach for both models. In Section 5.3, we address the issue of identifying the models orders using the minimum message length approach. In Section 5.4, we integrate the concept of feature saliency into the RPEM algorithm. The subsequent section 5.5 demonstrates some computer simulation and experimental results on challenging applications. Finally, the chapter closes with a summary of the work and concluding remarks.

5.2 Feature Selection for Asymmetric Mixture Models

Recently, finite mixture models have attracted a great deal of interest as a powerful framework for probabilistic inference and allow for reasoning with incomplete data. Let $\mathcal{X} = \{\vec{X}_1, \ldots, \vec{X}_N\}$ be a collection of N data points to be clustered, where each $\vec{X}_i \in \mathbb{R}^d$; $i \in \{1, \ldots, N\}$; is a vector of d

dimensions. We aim to fit the data points in $\mathcal{X}$ by a mixture model with M clusters:

$$p(\mathcal{X}|\Theta) = \prod_{i=1}^{N}\sum_{j=1}^{M} p_j p(\vec{X}_i|\xi_j) \tag{5.1}$$

where ξ_j is the set of the parameters of the jth component, $\{p_j\}$ are the mixing proportions which must be positive and sum to one, Θ= $\{p_1,\ldots,p_M,\xi_1,\ldots,\xi_M\}$ is the complete set of parameters fully characterizing the mixture, $M \geq 1$ is the number of components in the mixture. For the AGM, each component density $p(\vec{X}_i|\xi_j)$ is an asymmetric Gaussian distribution (AGD):

$$p(\vec{X}_i|\xi_j) = \prod_{k=1}^{d}\sqrt{\frac{2}{\pi}}\frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})} \times \begin{cases} \exp\left[-\frac{(X_{ik}-\mu_{jk})^2}{2\sigma_{l_{jk}}^2}\right] & \text{if } X_{ik} < \mu_{jk} \\ \exp\left[-\frac{(X_{ik}-\mu_{jk})^2}{2\sigma_{r_{jk}}^2}\right] & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{5.2}$$

where ξ_j = $(\vec{\mu}_j, \vec{\sigma}_{l_j}, \vec{\sigma}_{r_j})$ is the set of the parameters of component j where $\vec{\mu}_j$ = $(\mu_{j1},\ldots,\mu_{jd})$, $\vec{\sigma}_{l_j}$ = $(\vec{\sigma}_{l_{j1}},\ldots,\vec{\sigma}_{l_{jd}})$, and $\vec{\sigma}_{r_j}$ = $(\vec{\sigma}_{r_{j1}},\ldots,\vec{\sigma}_{r_{jd}})$ are the mean, the left standard deviation, and the right standard deviation of the d-dimensional AGD, respectively. Regarding the AGGM, each component $p(\vec{X}_i|\xi_j)$ is modeled by an AGGD given by:

$$p(\vec{X}_i|\xi_j) = \prod_{k=1}^{d}\frac{\beta_{jk}\left[\frac{\Gamma(3/\beta_{jk})}{\Gamma(1/\beta_{jk})}\right]^{1/2}}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})\Gamma(1/\beta_{jk})} \times \begin{cases} \exp\left[-A(\beta_{jk})\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta_{jk}}\right] & \text{if } X_{ik} < \mu_{jk} \\ \exp\left[-A(\beta_{jk})\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}}\right] & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{5.3}$$

where $A(\beta_{jk}) = \left[\frac{\Gamma(3/\beta_{jk})}{\Gamma(1/\beta_{jk})}\right]^{\beta_{jk}/2}$; ξ_j= $\{\vec{\mu}_j, \vec{\sigma}_{lj}, \vec{\sigma}_{rj}, \vec{\beta}_j\}$; $\vec{\mu}_j$ = $(\mu_{j1},\ldots,\mu_{jd})$, $\vec{\sigma}_{lj}$ = $(\sigma_{l_{j1}},\ldots,\sigma_{l_{jd}})$, 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. We introduce stochastic indicator variables, $Z_i = (Z_{i1},\ldots,Z_{iM})$, one for each observation, whose role is to encode to which component the observation belongs. In other words, Z_{ij}, the unobserved or missing vector, equals 1 if $\vec{X}_i$ belongs to class j and 0, otherwise. The complete-data likelihood for this case is then:

$$p(\mathcal{X}, Z|\Theta) = \prod_{i=1}^{N}\prod_{j=1}^{M}\left(p_j p(\vec{X}_i|\xi_j)\right)^{Z_{ij}} \tag{5.4}$$

where $Z = \{Z_1, \ldots, Z_N\}$. Taking the logarithm of equation 5.4 we can get the complete data log-likelihood by:

$$\log p(\mathcal{X}, Z|\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} Z_{ij} \log[p_j p(\vec{X}_i|\xi_j)] \tag{5.5}$$

Note that Eq.5.1 assumes that the d features have equal importance and carry pertinent information which is not usually the case, since many of which can be irrelevant for the intended application [33, 34, 35, 36]. We approach this problem by assuming that irrelevant features follow a background Gaussian distribution with parameter $\vec{\lambda} = \{\vec{\eta}, \vec{\delta}\}$ for all classes, where $\vec{\eta} = (\eta_1, \ldots, \eta_d)$ and $\vec{\delta} = (\delta_1, \ldots, \delta_d)$ represent the mean and standard deviation of the Gaussian distribution, respectively. We adopt the feature relevancy approach suggested by [43] in the case of the finite Gaussian mixture, because it is suitable for unsupervised learning. The main idea is to consider the k^{th} feature as irrelevant if its distribution is independent of the class labels and can follow our common Gaussian density $p(X_k|\lambda_k)$. Then, the mixture density in Eq.5.1 can be written as:

$$p(\vec{X}|\Theta, \vec{\lambda}, \vec{\varphi}) = \sum_{j=1}^{M} p_j \prod_{k=1}^{d} p(X_k|\xi_{jk})^{\varphi_k} p(X_k|\lambda_k)^{1-\varphi_k} \tag{5.6}$$

where $\lambda_k = (\eta_k, \delta_k)$ and $\vec{\varphi} = [\varphi_1, \ldots, \varphi_d]^T$ is a set of binary parameters, such that φ_k= 1 if the k^{th} feature is relevant and φ_k= 0, otherwise. Note that, $\{\varphi_k\}$ can be considered as missing variables. Thus, the resulting model can be given by [43]:

$$p(\vec{X}|\Theta_M) = \sum_{j=1}^{M} p_j \prod_{k=1}^{d} \Big[\omega_k p(X_k|\xi_{jk}) + (1-\omega_k) p(X_k|\lambda_k)\Big] \tag{5.7}$$

where $\Theta_M = \{\Theta, \vec{\omega}, \vec{\lambda}\ \}$ is the complete set of parameters fully characterizing the mixture. We suppose that not all the features of an observation are important, through the weight relevancy of these features. That is, the weight is denoted as $\omega = [\omega_1, \ldots, \omega_d]^T$ with $0 \leq \omega_k \leq 1$, where ω_k represents the probability that the k^{th} feature is relevant to all the clusters ($\omega_k = p(\varphi_k = 1)$). Therefore, the irrelevant features have little contribution to a given cluster in the subspace, thus their distributions are common to all the clusters in this case. We finally note that the previous model is reduced to the one in Eq.5.1 when all the feature are considered as relevant.

5.3 Learning via EM and MML

In the following, we present our first unsupervised learning approach for simultaneous clustering and feature selection. In particular, we propose an approach to find the optimal number of model components using MML and to estimate the different parameters using EM.

5.3.1 Parameter estimation using EM

In this section, we develop the equations that learn the parameters of the model while simultaneously consider the relevancy of features. To achieve this goal, we adopt common EM approach which generates a sequence of models with non-decreasing log-likelihood on the data. First, we suppose the number of mixture M is known. The maximum likelihood method consists of getting the mixture parameters that maximize the log-likelihood function given by:

$$\begin{aligned} L(\mathcal{X},\Theta_M,Z,\varphi) &= \sum_{i,j,\varphi} p(Z_i=j,\varphi|\vec{X}_i)\Bigg\{\log p_j + \sum_k \Bigg(\varphi_k(\log p(X_{ik}|\xi_{jk}) \\ &+ \log w_k) + (1-\varphi_k)(\log p(X_{ik}|\lambda_k) + \log(1-w_k))\Bigg)\Bigg\} \\ &= \sum_{i,j} p(Z_i=j|\vec{X}_i)\log p_j + \sum_{i,j}\sum_k \sum_{\varphi_k=0}^{1} p(Z_i=j,\varphi_k|\vec{X}_i) \\ &\times \Bigg(\varphi_k(\log p(X_{ik}|\xi_{jk}) + \log w_k) + (1-\varphi_k)(\log p(X_{ik}|\lambda_k) + \log(1-w_k))\Bigg) \end{aligned} \tag{5.8}$$

Thus, following [43], the EM algorithm for parameters estimation can be given by:

- **Expectation Step:**

$$h(j|\vec{X}_i,\Theta_M) = \frac{p_j \prod_{k=1}^{d} \zeta_{ijk}}{\sum_{j=1}^{M} p_j \prod_{k=1}^{d} \zeta_{ijk}} \tag{5.9}$$

- **Maximization Step:**

$$p_j^{new} = \frac{\sum_{i=1}^{N} h(j|\vec{X}_i,\Theta_M)}{N} \tag{5.10}$$

$$\xi_j^{new} = \text{arg max}_{\xi} L(\mathcal{X},\Theta_M,Z,\varphi) \tag{5.11}$$

$$\eta_k^{new} = \frac{\sum_{i=1}^{N} \Big[\sum_{j=1}^{M} q_{ijk} \Big] X_{ik}}{\sum_{i=1}^{N} \sum_{j=1}^{M} q_{ijk}} \tag{5.12}$$

$$\delta_k^{2^{new}} = \frac{\sum_{i=1}^{N} \Big[\sum_{j=1}^{M} q_{ijk} \Big] (X_{ik} - \eta_k)^2}{\sum_{i=1}^{N} \sum_{j=1}^{M} q_{ijk}} \tag{5.13}$$

$$\omega_k^{new} = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} r_{ijk}}{N} \tag{5.14}$$

where

$$\zeta_{ijk} = \omega_k p(X_{ik}|\xi_{jk}) + (1 - \omega_k) p(X_{ik}|\lambda_k) \tag{5.15}$$

$$r_{ijk} = \frac{\omega_k p(X_{ik}|\xi_{jk}) h(j|\vec{X}_i, \Theta_M)}{\zeta_{ijk}} \qquad q_{ijk} = \frac{(1 - \omega_k) p(X_{ik}|\lambda_k) h(j|\vec{X}_i, \Theta_M)}{\zeta_{ijk}} \tag{5.16}$$

In the case of the AGM model, equation 5.11 gives:

$$\mu_{jk}^{new} = \frac{\sum_{i=1}^{N} r_{ijk} X_{ik}}{\sum_{i=1}^{N} r_{ijk}} \tag{5.17}$$

$$\sigma_{l_{jk}}^{new} = \sigma_{l_{jk}}^{old} - \left[\left(\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{l_{jk}}^2} \right)^{-1} \left(\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{l_{jk}}} \right) \right] \tag{5.18}$$

$$\sigma_{r_{jk}}^{new} = \sigma_{r_{jk}}^{old} - \left[\left(\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{r_{jk}}^2} \right)^{-1} \left(\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{r_{jk}}} \right) \right] \tag{5.19}$$

Concerning the AGGM model, equation 5.11 gives:

$$\mu_{jk}^{new} = \mu_{jk}^{old} - \left[\left(\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \mu_{jk}^2} \right)^{-1} \left(\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \mu_{jk}} \right) \right] \tag{5.20}$$

$$\beta_{jk}^{new} = \beta_{jk}^{old} - \left[\left(\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \beta_{jk}^2} \right)^{-1} \left(\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \beta_{jk}} \right) \right] \tag{5.21}$$

$$\sigma_{l_{jk}}^{new} = \sigma_{l_{jk}}^{old} - \left[\left(\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{l_{jk}}^2} \right)^{-1} \left(\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{l_{jk}}} \right) \right] \qquad (5.22)$$

$$\sigma_{r_{jk}}^{new} = \sigma_{r_{jk}}^{old} - \left[\left(\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{r_{jk}}^2} \right)^{-1} \left(\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{r_{jk}}} \right) \right] \qquad (5.23)$$

Note that the gradients with respect to ($\sigma_{l_{jk}}$, $\sigma_{r_{jk}}$) and (μ_{jk}, β_{jk}, $\sigma_{l_{jk}}$, $\sigma_{r_{jk}}$) are nonlinear for the AGM and AGGM model, respectively. Therefore, we have decided to use the Newton-Raphson method for estimation. Thus, we have calculated the first and second gradient of the complete data loglikelihood with respect to these parameters as shown in appendix D.

5.3.2 Model selection using MML

Generally, the maximum likelihood estimate favors higher values for M which leads to overfitting. Therefore, a model selection criterion is needed to estimate the number of components of a mixture model. 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 building a short code for your data. Therefore, in the case of MML, the optimal number of classes in the mixture is obtained by minimizing the following cost function [32, 59]:

$$MessLens \approx -\log p(\theta_M) + \frac{c}{2}(1 + \log \frac{1}{12}) + \frac{1}{2} \log |I(\Theta_M)| - \log p(\mathcal{X}|\theta_M) \qquad (5.24)$$

where $p(\theta_M)$, $I(\Theta_M)$, and $p(\mathcal{X}|\theta_M)$ denote the prior distribution, the Fisher information matrix, and the likelihood, respectively. $|.|$ denotes the determinant, and c represents the total number of parameters. Thus, c $=M + d + 3dM + 2d$ and c $=M + d + 4dM + 2d$ for the AGM and AGGM, respectively. Note that the information matrix of the model is very difficult to obtain analytically, therefore, we assume the independence of the different groups of parameters, which allows the factorization of both $p(\theta_M)$ and $|I(\Theta_M)|$. Furthermore, we approximate the Fisher information $|I(\Theta_M)|$ using the complete likelihood which assumes labeled observations. Additionally, since we have no knowledge about the parameters, we adopt the uninformative

Jeffrey's prior for each group of parameters as prior distribution. From this, we obtain the following objective:

$$\begin{aligned} MessLens &\approx \frac{c}{2}(1+\log\frac{1}{12})+\frac{c}{2}(\log N)+\frac{DM}{2}\sum_{k=1}^{d}\log\omega_k+\frac{Dd}{2}\sum_{j=1}^{M}\log p_j \\ &+ \sum_{k=1}^{d}\log\left(1-\omega_k\right)-\log p(\mathcal{X}|\theta_M) \end{aligned} \tag{5.25}$$

where D is the number of parameters for each component of the mixture (D = 3 and D = 4 for the AGM and AGGM, respectively). We minimize eq.5.25 under the constraints $0 < p_j \leq 1$, $0 < \omega_k \leq 1$, and $\sum_{j=1}^{M} p_j = 1$ in a manner similar to the one followed in [43]. In order to use the MML approach the EM algorithm undergoes a minor modification in the calculation of the mixing proportions p_j and the feature relevancy ω_k:

$$p_j^{new} = \frac{max\Bigg(\sum_{i=1}^{N} h(j|\vec{X}_i,\Theta_M)-\frac{Dd}{2},0\Bigg)}{\sum_{j=1}^{M} max\Bigg(\sum_{i=1}^{N} h(j|\vec{X}_i,\Theta_M)-\frac{Dd}{2},0\Bigg)} \tag{5.26}$$

$$\omega_k^{new} = \frac{max\Bigg(\sum_{i=1}^{N}\sum_{j=1}^{M} r_{ijk}-\frac{DM}{2},0\Bigg)}{max\Bigg(\sum_{i=1}^{N}\sum_{j=1}^{M} r_{ijk}-\frac{DM}{2},0\Bigg)+max\Bigg(\sum_{i=1}^{N}\sum_{j=1}^{M} q_{ijk}-1,0\Bigg)} \tag{5.27}$$

5.3.3 The Complete learning algorithm

The following script summarizes the main steps of the algorithm used for parameters estimation and model selection:

1. Initialize Θ_M:
   - The feature relevancy is set to $\omega_k = 0.5$.
   - The number of components $M = M_{max} = 10$.
   - For both models, Θ is initialized using the Fuzzy C-means. Note that, we initialized both the left and right standard deviations with the standard deviation values obtained from the Fuzzy C-means. Furthermore, we initialized the shape parameter β with 2 in the case of the AGGM.

```
    • Perform the common Gaussian density $\vec{\lambda}$ parameters estimation to cover
      the whole data.
2. Implement the EM+MML approach
   **While** $M < M_{max}$ **do**{
   (a) **While** not converged **do** {
         i. Perform E-step according to Eq. 5.9.
        ii. Perform M-step according to Eqs. 5.11 to 5.13, 5.26, and 5.27.
       iii. **If** $p_j$= 0, **Then** the $j^{th}$ component is eliminated.
        iv. **If** $\omega_k$ = 0, **Then** $p(X_{ik}|\xi_{jk})$ is eliminated.
         v. **If** $\omega_k$ = 1, **Then** $p(X_{ik}|\lambda_k)$ is eliminated.
       }**End While**
   (b) Calculate the associated message length using Eq. 5.25.
   (c) Remove the component $j$ with the smallest $p_j$.
   }**End While**
3. Return the model parameters with the smallest message length.
```

5.4 Learning via RPEM

Recently, the RPEM algorithm [35] has been suggested to determine the model order automatically together with the estimation of the model parameters. This algorithm introduces unequal weights into the likelihood; thus the weighted likelihood in our case is written below:

$$Q(\Theta_M, \mathcal{X}) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{M}(\Theta_M, \vec{X}_i) \tag{5.28}$$

with

$$\begin{aligned}\mathcal{M}(\Theta_M, \vec{X}_i) &= \sum_{j=1}^{M} g(j|\vec{X}_i, \Theta_M)\ln\Bigg\{p_j\prod_{k=1}^{d}\Bigg[\omega_k p(X_{ik}|\xi_{jk}) \\ &+ (1-\omega_k)p(X_{ik}|\lambda_k)\Bigg]\Bigg\} - \sum_{j=1}^{M} g(j|\vec{X}_i, \Theta_M)\ln h(j|\vec{X}_i, \Theta_M)\end{aligned} \tag{5.29}$$

where $g(j|\vec{X}_i, \Theta_M)$, j = [1, ..., M], are designable weight functions, satisfying the two constraints below:

$$\sum_{j=1}^{M} g(j|\vec{X}_i, \Theta_M) = 1 \tag{5.30}$$

$$\forall j \qquad g(j|\vec{X}_i, \Theta_M) = 0 \quad if \quad h(j|\vec{X}_i, \Theta_M) = 0 \tag{5.31}$$

Thus, following [35], the weight $g(j|\vec{X}_i, \Theta_M)$ can be expressed by:

$$g(j|\vec{X}_i, \Theta_M) = (1+\varepsilon)I(j|\vec{X}_i, \Theta_M) - \varepsilon h(j|\vec{X}_i, \Theta_M) \tag{5.32}$$

where ε is a small positive quantity which we took as 1. Also

$$I(j|\vec{X}_i, \Theta_M) \quad = \quad \begin{cases} 1 & \text{if } j = c \\ 0 & \text{if } j \neq c \end{cases} \tag{5.33}$$

and $c = \arg\max_{\{1 \leq j \leq M\}} h(j|\vec{X}_i, \Theta_M)$. More details about RPEM can be found in [35]. In the following, we summarize the steps of the feature weighted RPEM (FW-RPEM) algorithm [189] for our two models.

1. Initialize Θ_M:
   - The feature relevancy is set to ω_k = 0.5.
   - The number of components M= M_{max} = 10.
   - For both models, Θ is initialized using the Fuzzy C-means. Note that, we initialized both the left and right standard deviations with the standard deviation values obtained from the Fuzzy C-means. Furthermore, we initialized the shape parameter β with 2 in the case of the AGGM.
2. Perform the common Gaussian density $\vec{\lambda}$ parameters estimation:

$$\eta_k = \frac{1}{N}\sum_{i=1}^{N} X_{ik} \tag{5.34}$$

$$\delta_k^2 = \frac{1}{N}\sum_{i=1}^{N}(X_{ik} - \eta_k)^2 \tag{5.35}$$

3. Repeat until convergence for each $\vec{X}_i$, $i = 1, \ldots, N$

- **Expectation Step**

$$h(j|\vec{X}_i, \Theta_M) = \frac{p_j \prod_{k=1}^{d} \zeta_{ijk}}{\sum_{j=1}^{M} p_j \prod_{k=1}^{d} \zeta_{ijk}}$$

$$g(j|\vec{X}_i, \Theta_M) \quad = \quad \begin{cases} 2 - h(j|\vec{X}_i, \Theta_M) & \text{if } j = c \\ -h(j|\vec{X}_i, \Theta_M) & \text{if } j \neq c \end{cases} \tag{5.36}$$

- **Maximization Step**

$$\begin{aligned} \phi_j^{new} &= \phi_j^{old} + \gamma_\phi \frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial \phi_j}\bigg|_{\Theta_M^{old}} \\ &= \phi_j^{old} + \gamma_\phi (g(j|\vec{X}_i, \Theta_M) - p_j^{old}) \end{aligned} \tag{5.37}$$

$$\xi_{jk}^{new} = \xi_{jk}^{old} + \gamma \frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial \xi_{jk}}\bigg|_{\Theta_M^{old}} \tag{5.38}$$

$$\begin{aligned} \omega_k^{new} &= \omega_k^{old} + \gamma_\omega \frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial \omega_k}\bigg|_{\Theta_M^{old}} \\ &= \omega_k^{old} + \gamma_\omega \sum_{j=1}^{M} g(j|\vec{X}_i, \Theta_M) \frac{\upsilon_{ijk}}{\zeta_{ijk}} \\ & if \quad \omega_k > 1 \quad then \quad \omega_k = 1 \\ & if \quad \omega_k < 0 \quad then \quad \omega_k = 0 \end{aligned} \tag{5.39}$$

where

$$p_j = \frac{\exp(\phi_j)}{\sum_{j=1}^{M} \exp(\phi_j)} \quad for \quad 1 \leq j \leq M \tag{5.40}$$

$$\upsilon_{ijk} = p(X_{ik}|\xi_{jk}) - p(X_{ik}|\lambda_k) \tag{5.41}$$

In the case of the AGM model, equation 5.38 can be given by:

$$\begin{aligned} \mu_{jk}^{new} &= \mu_{jk}^{old} + \gamma \frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial \mu_{jk}}\bigg|_{\Theta_M^{old}} \\ &= \mu_{jk}^{old} + \gamma g(j|\vec{X}_i, \Theta_M) \frac{\omega_k^{old}}{\zeta_{ijk}} \frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \mu_{jk}} \end{aligned} \tag{5.42}$$

$$
\begin{align}
S_{l_{jk}}^{new} &= S_{l_{jk}}^{old} + \gamma \left.\frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial S_{l_{jk}}}\right|_{\Theta_M^{old}} \tag{5.43}\\
&= S_{l_{jk}}^{old} + \gamma g(j|\vec{X}_i, \Theta_M) \frac{\omega_k^{old}}{\zeta_{ijk}} \frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial S_{l_{jk}}} \nonumber
\end{align}
$$

$$
\begin{align}
S_{r_{jk}}^{new} &= S_{r_{jk}}^{old} + \gamma \left.\frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial S_{r_{jk}}}\right|_{\Theta_M^{old}} \tag{5.44}\\
&= S_{r_{jk}}^{old} + \gamma g(j|\vec{X}_i, \Theta_M) \frac{\omega_k^{old}}{\zeta_{ijk}} \frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial S_{r_{jk}}} \tag{5.45}
\end{align}
$$

where

$$
S_{l_{jk}} = \frac{1}{\sigma_{l_{jk}}^2} \qquad\qquad S_{r_{jk}} = \frac{1}{\sigma_{r_{jk}}^2} \tag{5.46}
$$

Concerning the AGGM model, equation 5.38 can be given by:

$$
\begin{align}
\mu_{jk}^{new} &= \mu_{jk}^{old} + \gamma \left.\frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial \mu_{jk}}\right|_{\Theta_M^{old}} \tag{5.47}\\
&= \mu_{jk}^{old} + \gamma g(j|\vec{X}_i, \Theta_M) \frac{\omega_k^{old}}{\zeta_{ijk}} \frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \mu_{jk}} \nonumber
\end{align}
$$

$$
\begin{align}
\beta_{jk}^{new} &= \beta_{jk}^{old} + \gamma \left.\frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial \beta_{jk}}\right|_{\Theta_M^{old}} \tag{5.48}\\
&= \beta_{jk}^{old} + \gamma g(j|\vec{X}_i, \Theta_M) \frac{\omega_k^{old}}{\zeta_{ijk}} \frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \beta_{jk}} \nonumber
\end{align}
$$

$$
\begin{align}
\sigma_{l_{jk}}^{new} &= \sigma_{l_{jk}}^{old} + \gamma \left.\frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial \sigma_{l_{jk}}}\right|_{\Theta_M^{old}} \tag{5.49}\\
&= \sigma_{l_{jk}}^{old} + \gamma g(j|\vec{X}_i, \Theta_M) \frac{\omega_k^{old}}{\zeta_{ijk}} \frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \sigma_{l_{jk}}} \nonumber
\end{align}
$$

$$
\begin{align}
\sigma_{r_{jk}}^{new} &= \sigma_{r_{jk}}^{old} + \gamma \left.\frac{\partial \mathcal{M}(\Theta_M, \vec{X}_i)}{\partial \sigma_{r_{jk}}}\right|_{\Theta_M^{old}} \tag{5.50}\\
&= \sigma_{r_{jk}}^{old} + \gamma g(j|\vec{X}_i, \Theta_M) \frac{\omega_k^{old}}{\zeta_{ijk}} \frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \sigma_{r_{jk}}} \nonumber
\end{align}
$$

Note that the learning rates γ_ϕ, γ, and γ_ω are taken as 0.0001, 0.001, and 0.0001, respectively. Also, ($\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \mu_{jk}}$, $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial S_{l_{jk}}}$, $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial S_{r_{jk}}}$) for the AGM model and ($\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \mu_{jk}}$, $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \beta_{jk}}$, $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \sigma_{l_{jk}}}$, $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \sigma_{r_{jk}}}$) for the AGGM model are given in appendix E.

5.5 Experimental Results

In this section, the effectiveness of the two proposed frameworks for learning the AGM and AGGM models are tested on two real-world applications namely scene categorization and facial expression recognition.

5.5.1 Scene Categorization

One of the most impressive feats of the human visual system is how rapidly, accurately and comprehensively it can recognize and understand a complex scene [190]. This remarkable ability, known as ”visual recognition”, has recently drawn considerable interest and has been successfully applied in various applications such as the automatic understanding of images, object recognition, image databases browsing and content-based images annotation, suggestion and retrieval [191–199]. In this section, we build a method for recognizing scene categories by imitating the human perception. Thus, our approach can be divided into three main components: feature extraction, image representation, and scene classification. In feature extraction, we normalize the images then we represent each image by a collection of local image patches. In addition, we scan local image patches and extract their low level features vectors. Then, we use the bag of visual words (BOW) approach to have an overall representation for each image [200]. The last step in our image representation step is to apply a probabilistic Latent Semantic Analysis (pLSA) to the obtained histograms to represent each image by a d-dimensional vector where d is the number of latent aspects [201]. Our final goal is to classify the overall image to its right group using our two models.
The remainder of this section is organized as follows, first we represent some related works for scene classification. Then, we introduce the databases used in this application. Later, we describe the three components of our algorithm. Finally, we evaluate the performance of our algorithm for scene classification.

Related Works

Classifying images into semantic types of scenes [202–204] is a classical image understanding problem. Automatic techniques for recognizing scenes have an enormous impact for improving the performance of other computer vision applications such as browsing, retrieval and object recognition. However, scenes classification is not an easy task owing to their variability, ambiguity, and the wide range of illumination and scale conditions that may apply. In [205], the authors presented a stratified approach to both binary (outdoor-indoor) and multiple categories scene classification. Their idea was to learn mixture models for 20 basic classes of local image content based on color and texture information. Then, they applied these models to the test image in order to produce 20 probability density response maps (PDRM) indicating the likelihood that each image region was produced by each class. Later, they extracted some very simple features from those PDRMs, and used them to train a bagged LDA classifier for 10 scene categories. In [202], the authors used a simplified low-level feature set to predict multiple semantic scene attributes that are integrated probabilistically to obtain a final indoor/outdoor scene classification. An initial indoor/outdoor prediction is obtained by using support vector machines for classifying computationally efficient, low-dimensional color and wavelet texture features. Furthermore, they used the same low-level features to explicitly predict the presence of semantic features including grass and sky. The semantic scene attributes are then integrated using a Bayesian network designed for improved indoor/outdoor scene classification. In [206], the authors proposed a hierarchical generative model that classifies the overall scene, recognizes and segments each object component, as well as annotates the image with a list of tags. Visually relevant objects are represented by regions and patches, while visually irrelevant textual annotations are influenced directly by the overall scene class. However, these methods use manually annotated patches which may be time consuming and unpractical. The authors in [207] proposed the use of pLSA to discover object categories in images using the bag-of-words document representation. Then, they used the nearest neighbor classifier for scenes classification. Our research efforts are focused on extending this last approach in order to construct a robust system capable of classifying images into different scenes.

Scene Databases

In this section, we test our approach on two well-known data sets. Our first dataset contains 1579 diverse scene images from 8 categories [206]: rowing (250 images), badminton (200 images), polo (182 images), bocce (137 images), snowboarding (190 images), croquet (236 images), sailing (190 images), and rock climbing (194 images). This data set is very challenging because most images have highly cluttered and diverse background, and object classes are highly diverse. In addition, object sizes and poses are very different in each image. The second data set contains 15 categories of natural scenes [208, 209]: highway (260 images), inside of cities (308 images), tall buildings (356 images), streets (292 images), suburb residence (241 images), forest (328 images), coast (360 images), mountain (374 images), open country (410 images), bedroom (174 images), kitchen (151 images), livingroom (289 images), office (216 images), store (315 images), and industrial (311 images). The major sources of the pictures in the data set include the COREL collection, personal photographs, and Google image search. The average size of each image is approximately 250 × 300 pixels. Figures 5.1 and 5.2 show example images from the two data sets under consideration.

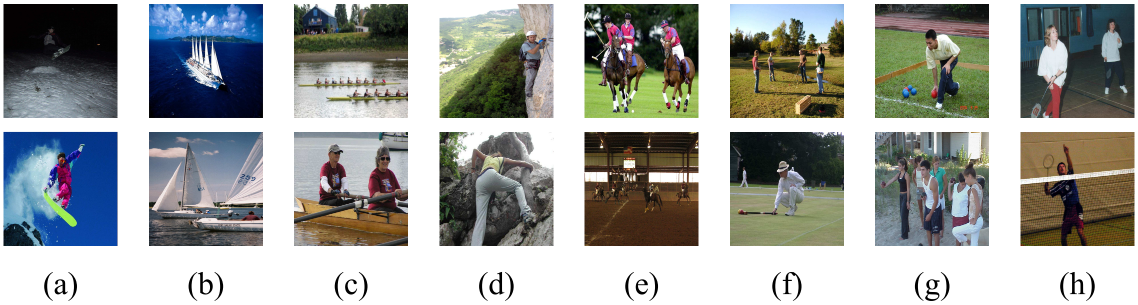


Figure 5.1: Sample images from the UIUC sports event data set; (a) Snow-boarding, (b) Sailing, (c) Rowing, (d) Rock-climbing, (e) Polo, (f) Croquet, (g) Bocce, (h) Badminton.

Feature Extraction and Image Representation

Representing an image by a collection of local image patches of certain size has become very popular and achieved certain success in visual recognition, image retrieval,

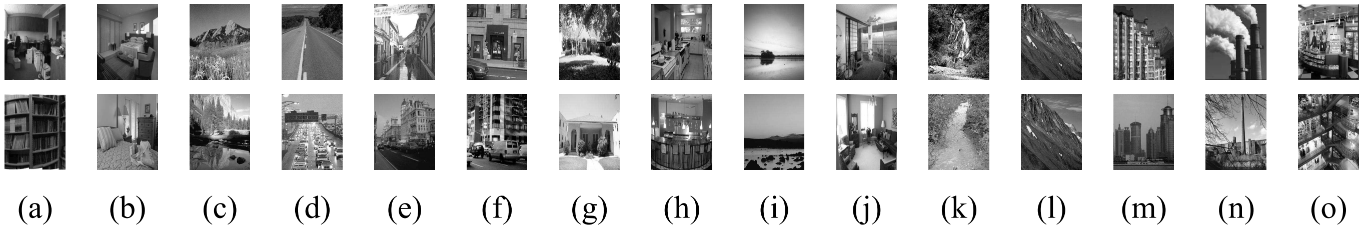


Figure 5.2: Sample images from the 15 categories data set; (a) Office, (b) Bedroom, (c) Open country, (d) Highway, (e) Street, (f) inside-city, (g) Suburb-residence, (h) kitchen, (i) Coast, (j) Living-room, (k) Forest, (l) Mountain, (m) Tall-buildings, (n) Industrial, (o) Store.

scene modeling/categorization, etc., due to its robustness to occlusions, geometric deformations and illumination variations. In addition, the strategy of dense sampling has been shown to provide better performance than interest points for scene classification [208]. Furthermore, SIFT descriptor is robust to illumination, clutter and scale changes. Therefore in this section we use dense SIFT descriptors of 16×16 pixel patches computed over a grid with spacing of 8 pixels.
Inspired by the huge success of the Bag of words (BOW) method in scenes classification, we decided to employ it in order to represent each image by a feature vector [200]. In order to build the Bag of Words dictionary, also known as codebook, we use a K-means algorithm to cluster our training-set descriptors in a vocabulary of v visual words. Then, for each SIFT point in an input image, the nearest neighbor in the vocabulary is calculated; based on this statistics a v-dimensional feature vector is built collecting the number of points in the image that can be approximated by each of the v visual words. Thus, each image can be represented as a frequency histogram over the v visual words. Then, we apply the pLSA model to the bag of visual words representation which allows the description of each image as a d-dimensional vector, where d is the number of aspects (or learned topics). Note that, for both data sets, we fixed v and d to 900 and 50, respectively.

Scene Classification and Results

The goal of classification is to estimate the most likely scene class. While classification might be easy for human beings it is very hard for machines. Recently, several classification methods were introduced and most of them fall into two broad classes: deterministic and probabilistic classification. Deterministic approaches classify each image to one of a number of classes. This is done by considering some metric that defines the distance between classes and by defining the class boundaries. On the other hand, the probabilistic method classifies each image by calculating its probabilities of belonging to each class of interest. We believe that a probabilistic classification approach is more suitable because of its robustness to measurement error and its effectiveness in identifying similar characteristics from supervised training images. Therefore, we use the AGM and AGGM to model the training images of each class. Then, for each input image, we calculate its likelihood of being generated from each class. Finally, we classify each image to the class that maximizes more its likelihood.

For the UIUC sports event data set, we have used a color SIFT descriptor in order to incorporate color information. The only difference between the color SIFT and the regular Gray SIFT is the number of input channels (one versus three). Therefore, we represented each image using the HSV (Hue, Saturation, and Value) color space. For each event class, 70 randomly selected images are used for training and 60 are used for testing. Note that, we evaluated the performance of the proposed algorithm by running it 20 times, as well as using the $RPEM$ and the $EM + MML$ for the unsupervised learning of our models parameters. The confusion matrices calculated by the $AGM + EM + MML$, the $AGM + RPEM$, the $AGGM + EM + MML$, and the $AGGM + RPEM$ are shown in tables 5.1, 5.2, 5.3, and 5.4, respectively. It is noteworthy that the computational time excluding the pre-processing time of feature detection and visual vocabulary formation of the AGM-EM-MML, AGM-RPEM, AGGM-EM-ML, and AGGM-RPEM approaches running on Core i7-2.4 GHz processor are 7, 5, 10, 6 minutes for the training set.

Table 5.1: The confusion matrix of the AGM-EM-MML for the UIUC sports event data set.

	Snow-boarding	Sailing	Rowing	Rock-climbing	Polo	Croquet	Bocce	Badminton.
Snow-boarding	**0.81**	0.02	0.00	0.02	0.00	0.01	0.01	0.13
Sailing	0.04	**0.83**	0.05	0.00	0.00	0.00	0.00	0.08
Rowing	0.00	0.00	**0.79**	0.00	0.03	0.13	0.04	0.01
Rock-climbing	0.02	0.01	0.00	**0.91**	0.00	0.00	0.00	0.06
Polo	0.00	0.00	0.05	0.00	**0.67**	0.09	0.19	0.00
Croquet	0.00	0.00	0.11	0.00	0.10	**0.53**	0.26	0.00
Bocce	0.00	0.00	0.04	0.00	0.16	0.24	**0.54**	0.02
Badminton	0.03	0.00	0.00	0.06	0.00	0.00	0.00	**0.91**

Table 5.2: The confusion matrix of the AGM-RPEM for the UIUC sports event data set.

	Snow-boarding	Sailing	Rowing	Rock-climbing	Polo	Croquet	Bocce	Badminton.
Snow-boarding	**0.81**	0.02	0.00	0.02	0.00	0.01	0.01	0.13
Sailing	0.04	**0.83**	0.05	0.00	0.00	0.00	0.00	0.08
Rowing	0.00	0.00	**0.79**	0.00	0.03	0.13	0.04	0.01
Rock-climbing	0.02	0.01	0.00	**0.91**	0.00	0.00	0.00	0.06
Polo	0.00	0.00	0.05	0.00	**0.67**	0.09	0.19	0.00
Croquet	0.00	0.00	0.11	0.00	0.09	**0.54**	0.26	0.00
Bocce	0.00	0.00	0.04	0.00	0.16	0.22	**0.56**	0.02
Badminton	0.03	0.00	0.00	0.06	0.00	0.00	0.00	**0.91**

Table 5.3: The confusion matrix of the AGGM-EM-MML for the UIUC sports event data set.

	Snow-boarding	Sailing	Rowing	Rock-climbing	Polo	Croquet	Bocce	Badminton.
Snow-boarding	**0.83**	0.04	0.00	0.01	0.00	0.02	0.02	0.08
Sailing	0.04	**0.83**	0.05	0.00	0.00	0.00	0.00	0.08
Rowing	0.00	0.04	**0.70**	0.00	0.02	0.17	0.05	0.02
Rock-climbing	0.06	0.01	0.00	**0.89**	0.00	0.00	0.00	0.04
Polo	0.01	0.00	0.05	0.00	**0.67**	0.11	0.16	0.00
Croquet	0.00	0.00	0.08	0.00	0.13	**0.50**	0.29	0.00
Bocce	0.00	0.00	0.04	0.00	0.15	0.23	**0.57**	0.01
Badminton	0.02	0.00	0.00	0.07	0.00	0.00	0.00	**0.91**

Table 5.4: The confusion matrix of the AGGM-RPEM for the UIUC sports event data set.

	Snow-boarding	Sailing	Rowing	Rock-climbing	Polo	Croquet	Bocce	Badminton.
Snow-boarding	**0.84**	0.03	0.00	0.01	0.00	0.02	0.02	0.08
Sailing	0.01	**0.85**	0.04	0.00	0.00	0.00	0.00	0.0
Rowing	0.00	0.04	**0.70**	0.00	0.02	0.17	0.05	0.02
Rock-climbing	0.06	0.01	0.00	**0.89**	0.00	0.00	0.00	0.04
Polo	0.01	0.00	0.05	0.00	**0.67**	0.11	0.16	0.00
Croquet	0.00	0.00	0.08	0.00	0.12	**0.53**	0.27	0.00
Bocce	0.00	0.00	0.04	0.00	0.15	0.24	**0.56**	0.01
Badminton	0.02	0.00	0.00	0.07	0.00	0.00	0.00	**0.91**

For the 15 scenes categories data set, we are using 100 images per class for training and the rest for testing as suggested in [208, 209]. Note that, we evaluated the performance of the proposed algorithm by running it 20 times, as well as using the $RPEM$ and the $EM + MML$ for the unsupervised learning of our models parameters. The confusion matrices calculated by the $AGM + EM + MML$, the $AGM + RPEM$, the $AGGM + EM + MML$, and the $AGGM + RPEM$ are shown in tables 5.5, 5.6, 5.7, and 5.8, respectively. It is noteworthy that the computational time excluding the pre-processing time of feature detection and visual vocabulary formation of the AGM-EM-MML, AGM-RPEM, AGGM-EM-ML, and AGGM-RPEM approaches running on Core i7-2.4 GHz processor are 13, 11, 17, 14 minutes for the training set.

In order to evaluate the performance of both algorithms, we compare them with a

Table 5.5: The confusion matrix of the AGM-EM-MML for the 15 scenes categories data set.

	highway	inside-city	tall-buildings	streets	suburb	forest	coast	mountain	open-country	bedroom	kitchen	living-room	office	store	industrial
highway	**0.86**	0.00	0.00	0.03	0.00	0.00	0.07	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.01
inside-city	0.00	**0.81**	0.01	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.04	0.05
tall-buildings	0.00	0.02	**0.91**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07
streets	0.08	0.02	0.00	**0.90**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
suburb	0.00	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
forest	0.00	0.00	0.00	0.00	0.00	**0.95**	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.01	0.00
coast	0.01	0.00	0.00	0.00	0.00	0.00	**0.82**	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.01
mountain	0.00	0.00	0.00	0.00	0.00	0.03	0.00	**0.89**	0.08	0.00	0.00	0.00	0.00	0.00	0.00
open-country	0.06	0.00	0.00	0.00	0.00	0.00	0.16	0.07	**0.71**	0.00	0.00	0.00	0.00	0.00	0.00
bedroom	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.68**	0.14	0.18	0.00	0.00	0.00
kitchen	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	**0.69**	0.11	0.08	0.00	0.00
living-room	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.23	0.16	**0.61**	0.00	0.00	0.00
office	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	**0.93**	0.00	0.00
store	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.00	**0.75**	0.11
industrial	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.21	**0.65**

Table 5.6: The confusion matrix of the AGM-RPEM for the 15 scenes categories data set.

	highway	inside-city	tall-buildings	streets	suburb	forest	coast	mountain	open-country	bedroom	kitchen	living-room	office	store	industrial
highway	**0.86**	0.00	0.00	0.03	0.00	0.00	0.07	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.01
inside-city	0.00	**0.82**	0.01	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.04	0.05
tall-buildings	0.00	0.02	**0.93**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05
streets	0.08	0.01	0.00	**0.91**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
suburb	0.00	0.00	0.00	0.00	**0.99**	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
forest	0.00	0.00	0.00	0.00	0.00	**0.95**	0.00	0.01	0.03	0.00	0.00	0.00	0.00	0.01	0.00
coast	0.01	0.00	0.00	0.00	0.00	0.00	**0.82**	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.01
mountain	0.00	0.00	0.00	0.00	0.00	0.02	0.00	**0.91**	0.07	0.00	0.00	0.00	0.00	0.00	0.00
open-country	0.06	0.00	0.00	0.00	0.00	0.00	0.16	0.07	**0.71**	0.00	0.00	0.00	0.00	0.00	0.00
bedroom	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.68**	0.14	0.18	0.00	0.00	0.00
kitchen	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	**0.69**	0.09	0.00	0.00	0.00
living-room	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.23	0.16	**0.61**	0.00	0.00	0.00
office	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	**0.93**	0.00	0.00
store	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.00	**0.75**	0.11
industrial	0.00	0.00	0.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.21	**0.65**

Table 5.7: The confusion matrix of the AGGM-EM-MML for the 15 scenes categories data set.

	highway	inside-city	tall-buildings	streets	suburb	forest	coast	mountain	open-country	bedroom	kitchen	living-room	office	store	industrial
highway	**0.82**	0.00	0.00	0.07	0.00	0.00	0.06	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.02
inside-city	0.00	**0.81**	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05
tall-buildings	0.02	0.03	**0.89**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06
streets	0.08	0.01	0.00	**0.91**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
suburb	0.00	0.00	0.00	0.01	**0.98**	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
forest	0.00	0.00	0.00	0.00	0.00	**0.96**	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00
coast	0.00	0.00	0.00	0.00	0.00	0.00	**0.85**	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.00
mountain	0.00	0.00	0.00	0.00	0.00	0.03	0.00	**0.89**	0.08	0.00	0.00	0.00	0.00	0.00	0.00
open-country	0.04	0.00	0.00	0.00	0.00	0.03	0.17	0.06	**0.70**	0.00	0.00	0.00	0.00	0.00	0.00
bedroom	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.69**	0.11	0.20	0.00	0.00	0.00
kitchen	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.18	**0.70**	0.12	0.00	0.00	0.00
living-room	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.15	**0.63**	0.02	0.00	0.00
office	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.03	0.01	**0.92**	0.00	0.00
store	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	**0.78**	0.09
industrial	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.20	**0.65**

Table 5.8: The confusion matrix of the AGGM-RPEM for the 15 scenes categories data set.

	highway	inside-city	tall-buildings	streets	suburb	forest	coast	mountain	open-country	bedroom	kitchen	living-room	office	store	industrial
highway	**0.83**	0.00	0.00	0.07	0.00	0.00	0.05	0.00	0.03	0.00	0.00	0.00	0.00	0.00	0.02
inside-city	0.00	**0.81**	0.00	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.05
tall-buildings	0.02	0.05	**0.90**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03
streets	0.10	0.00	0.00	**0.90**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
suburb	0.00	0.00	0.00	0.01	**0.99**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
forest	0.00	0.00	0.00	0.00	0.00	**0.96**	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00
coast	0.00	0.00	0.00	0.00	0.00	0.00	**0.85**	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.00
mountain	0.00	0.00	0.00	0.00	0.00	0.03	0.00	**0.89**	0.08	0.00	0.00	0.00	0.00	0.00	0.00
open-country	0.00	0.00	0.00	0.00	0.00	0.04	0.15	0.06	**0.75**	0.00	0.00	0.00	0.00	0.00	0.00
bedroom	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0.69**	0.11	0.20	0.00	0.00	0.00
kitchen	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.18	**0.70**	0.12	0.00	0.00	0.00
living-room	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.15	**0.63**	0.02	0.00	0.00
office	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.03	0.01	**0.92**	0.00	0.00
store	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.13	0.00	0.00	**0.78**	0.09
industrial	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.0	0.00	0.00	0.19	**0.65**

number of state-of-the-art approaches:

1. **GMM-EM-MML [43]:** We use the same proposed method with the Gaussian mixture model for classification. Note that, we perform parameters estimations as well as model complexity determination simultaneously by incorporating a MML penalty in the model learning step.
2. **GMM-RPEM [189]:** We use the same proposed method with the Gaussian mixture model for classification. Note that, we perform parameters estimations as well as model complexity determination simultaneously by fading out the redundant densities in the mixture using the RPEM.
3. **GIST [210]:** A spatial envelope that represents the dominant spatial structure of a scene is proposed then for classification the K nearest neighbors (KNN) classifier is used.
4. **Hierarchical [208]:** The image is represented by a collection of local regions denoted as codewords then latent dirichlet allocation (LDA) algorithm is used to identify the different themes of the image.
5. **Probabilistic [211]:** A probabilistic model for jointly modeling the image, its class label, and its annotations is used.
6. **SPM [209]:** The Deformable Part-Based Models evaluated with LSVM training process are coupled together.

7. **BOW [200, 212]:** SIFT is used as input for the bag of words method and for classification SVM is used.
8. **Scene-Objects [206]:** An Integrative Model is used to classify events by integrating scene and object categorization.
9. **MLE-Scene [213]:** Both low level features and global features are extracted. Then, the maximum likelihood estimation is used to learn an upstream scene model.
10. **MM-Scene [213]:** Both low level features and global features are extracted. Then, the max-margin learning is used to learn an upstream scene model.
11. **OB-SVM [214]:** The object Bank method represents an image as a scale-invariant response map of a large number of pre-trained generic object detectors and uses the SVM algorithm to classify it.

Tables 5.9 and 5.10 show the average classification rates for the methods under consideration.

Table 5.9: The average accuracies (%) for the UIUC event data set.

GMM-EM-MML	69.51%	GMM-RPEM	69.76%	GIST	63.88%	Probabilistic	66.00%	SPM	71.57%
BOW	69.25%	Scene-Objects	73.40%	MLE-Scene	69.87%	MM-Scene	71.70%	OB-SVM	76.30%
AGM-EM-MML	74.87%	AGM-RPEM	75.08%	AGGM-EM-MML	73.75%	AGGM-RPEM	74.37%		

Table 5.10: The average accuracies (%) for the 15 categories data set.

GMM-EM-MML	74.78%	GMM-RPEM	74.21%	GIST	74.00%	Hierarchical	65.20%	SPM	81:20%	BOW	73.50%
OB-SVM	80.90%	AGM-EM-MML	80.69%	AGM-RPEM	81.38%	AGGM-EM-MML	81.20%	AGGM-RPEM	81.67 %		

From both evaluations, we can conclude that our methods with both learning algorithms can achieve good results for the task of scenes categorization. Also, we found that the RPEM approach achieve higher results than the EM-MML.

5.5.2 Static Facial Expression Recognition

Facial expressions are the facial changes with regards to a person's internal emotional states, intentions, or social communications. The detection of faces and the interpretation of facial expression under varying conditions regardless of context, culture, and gender is an easy task for humans, however, it represents a difficult problem for computer based systems. Facial expression represents a very important task for humans for example the impression induced from a displayed expression will affect our interpretation of the spoken word and even our attitude towards the speaker himself. Furthermore, facial expression analysis can be applied in many areas such as emotion and paralinguistic communication, clinical psychology, psychiatry, neurology, pain assessment, lie detection, intelligent environments, and multimodal human computer interface (HCI). Therefore, facial expression analysis has been an active research topic for behavioral scientists since the work of Darwin in 1872 [215]. The authors in [216] introduced a method to automatically analyze facial expressions by tracking the motion of 20 identified spots on an image sequence. Thereafter, much progress has been made to build computer systems to help us understand and use this natural form of human communication [217–221]. The main goal of theses researches is to create a computer system capable of automatically detecting the emotional state of any person. Thus, machine vision and pattern recognition algorithms are widely applied to face expression recognition (FER) on both still images and/or video sequences in order to extract emotional content from visual patterns of a persons face. Despite the progress made in recognizing facial expressions, reliable and accurate automatic FER is still an evolving research subject due to the subtlety, complexity and variability of facial expressions. Facial expression analysis methods can be classified into image based and video based. Facial expression are dynamic in nature, as a result video based methods are more robust since they encode the facial dynamics which are not available in static. However, there are some scenarios when temporal data is not available and image based facial expression analysis are highly needed such as in classifying expressions in consumer level photographs, expression based album creation, and smile detection. Therefore, in this section, we present a novel real-time system for static facial expressions recognition capable of recognizing seven facial expressions (The six basic facial expressions: Anger, Disgust, Fear, Joy, Sadness, Surprise with the Neutral expression). Fig. 5.3 shows a sample image from each of the seven facial expressions. Generally, the task of automatic facial expression analysis can be divided into three main steps: face detection, facial feature extraction, and classification into expressions.

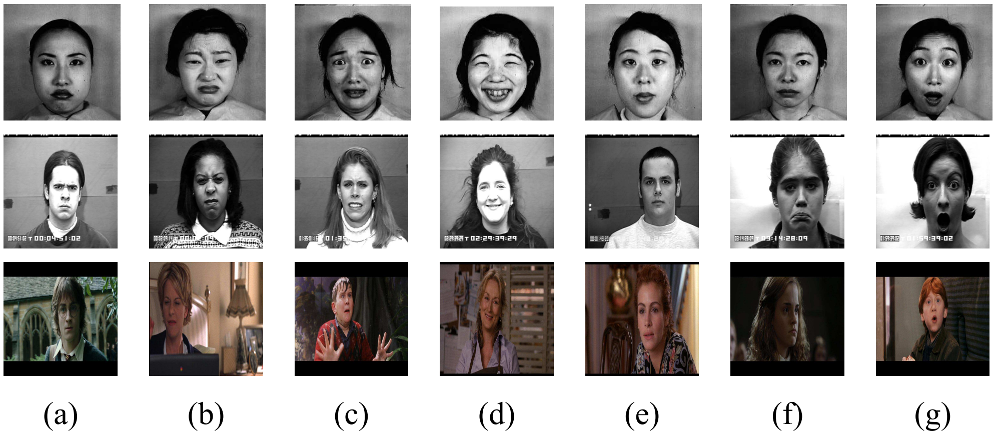


Figure 5.3: Sample images for the seven emotions from the three datasets under consideration. First Row: JAFFE dataset, Second Row: Cohn-Kanade dataset, and Thrid Row: SFEW dataset. Facial expressions are: (a) Angry, (b) Disgust, (c) Fear, (d) Joy, (e) Neutral, (f) Sadness, and (g) Surprised.

Face detection is used to recognize and locate face-like objects regardless of their positions, scales, orientations, poses, and illumination in the given image. Face detection is not straightforward due to various variations of image appearance, such as pose variation (front, non-front), occlusion, image orientation, and illuminating condition. Many novel methods have been proposed to automatically recognizing faces in images, such as motion detection (e.g. eye blinks), skin color segmentation and neural network based methods. However, motion based approaches are not applicable for our static case and skin tone detection does not perform equally well on different skin colors and is sensitive to changes in illumination. In this section, we are interested with real time system, therefore, we decided to use the method introduced in [222] for its small computational time and high detection accuracy. Their approach is capable of eliminating the need to compute a multi-scale image pyramid and thus reducing the time required for face detection significantly. The method can be divided into three main tasks; first, calculate the integral image where the integral image at a given location contains the sum of the pixels above and to the left of this location. Then, compute some rectangle features (Haar features) from this integral image and use AdaBoost for feature selection. Finally, a cascade of weak classifiers is employed for face detection. After faces are localized, facial feature extraction is performed on the cropped faces. Extracting facial feature is an important step and essential requirement in automated facial expression recognition and has been widely studied in the literature [217, 223, 224]. In this section, we test different feature extraction methods:

LBP [225]: The Local Binary Pattern (LBP) descriptor assigns binary labels to pixels by thresholding the neighborhood pixels with the central value. Later the operator was extended to use uniform patterns which are LBP that contain at most two bitwise transitions from 0 to 1 or vice versa. The advantage of using uniform patterns is that uniform patterns account for a bit less than 90 % of all patterns while can be represented by only 59 bins [226]. Note that, we have used the code available online [1]. Thus, in order to use the LBP descriptor, each facial image is segmented into a grid of 5 × 5 regions, then, we down-sampled the image into 3 resolutions with scale of 0.6. Therefore, we end up with a 4425-dimensional feature vector for each facial image (3 × 59 × 25).

LPQ [227, 228]: The Local Phase Quantization (LPQ) is a novel descriptor for texture classification. The descriptor uses phase information computed locally in a window for every image position. The phases of the four low-frequency coefficients are decorrelated and uniformly quantized into eignt-dimensional space. Then, a histogram of these integer values from all image positions is created and used as a feature for classication. Because only phase information is used, the method is also invariant to uniform illumination changes. Note that, we have used the code available online[2]. For this descriptor the cropped faces were divided into 5×5 blocks and the neighbourhood size was set to 8.

HOG [229]: Histogram of Oriented Gradients (HOG) descriptor counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

PHOG [230]: Pyramid of Histogram of Oriented Gradients (PHOG) descriptor represents local image shape and its spatial layout by a spacial pyramid kernel. Note that,

[1] http://www.cse.oulu.fi/CMV/Downloads/LBPMatlab

[2] http://www.cse.oulu.fi/CMV/Downloads/LPQMatlab

we have used the code available online[3]. Thus, we used a three level pyramid with 8 bin length and [0-360] angle range in our experiments.

WLD [231]: The Weber Local Descriptor (WLD) characterizes texture information of an image by considering the ratio of changes in pixel intensity. Different to LBP, it uses the gradient orientations to describe the direction of edges. For implementation, we use the code available online [4]. Note that, we used the same setup used for the LBP descriptor.

Then, the well established Principle Component Analysis (PCA) technique is used to reduce the dimensionality scope. The last step is facial expression classification which is used to identify the emotional state of the input person face by searching a database of known different emotional expressions. Over the years different classifiers were tested in facial expression recognition, however, we are interested with Support vector machine (SVM) and mixture models. Therefore, we decided to test three SVM classifiers: Linear (LSVM), polynomial (PolySVM), and SVM with Radial Basis Function (RBF) kernels (RBFSVM). In addition, for mixture based classifiers, we are using the Gaussian mixture model (GMM), the AGM, and the AGGM for comparison purposes. Furthermore, we employ the RPEM and the EM with MML approaches for model learning.

We have experimented the various algorithms on three well known databases:

JAFFE [118]: The Japanese Female Facial Expression (JAFFE) is one of the earliest static facial expressions databases. It contains 219 images of 10 Japanese females performing the 7 facial expressions under consideration. It has been extensively used in expression research. However, it has been created in a lab controlled environment with a limited number of samples. Figure 5.3 shows some sample images from this dataset. For our experiments, we have used a five-fold cross validation script [14], which creates five subsets of the dataset.

Cohn-Kanade [232]: Cohn-Kanade database is one of the most widely used test-beds for facial expression algorithm development and evaluation. This database consists of 97 university students between the ages of 18 to 50 years, of which 69% are female, 13% are African-American, 81% are Euro-American, and 6% are from other groups.

[3] http://www.robots.ox.ac.uk/ vgg/research/caltech

[4] http://vipl.ict.ac.cn/resources/codes

Subjects were instructed to perform a series of 23 facial displays, six of which were based on description of prototypic emotions, where each display began and ended with a neutral face. In addition, Image sequences were digitized into 640 × 490 pixel arrays with 8-bit precision for grayscale values. Figure 5.3 shows some sample images from this dataset. For our experiments, we selected 320 image sequences where each come from one of the six basic emotions. The sequences come from 96 subjects, with 1-6 emotions per subject. For each sequence, the neutral face and three peak frames were used for expression recognition, resulting in 1280 images (108 Anger, 120 Disgust, 99 Fear, 282 Joy, 126 Sadness, 225 Surprise and 320 Neutral). To evaluate the performance of the algorithms under consideration, we adopted a 10 cross validation testing scheme repeated for 10 times in our experiments.

SFEW [221, 233]: Static Facial Expression in the Wild (SFEW) has been developed by selecting frames from Acted Facial Expressions in the Wild (AFEW) database. The database covers unconstrained facial expressions, varied head poses, large age range, occlusions, varied focus, different resolution of face and close to real world illumination. Frames were extracted from AFEW sequences and labeled based on the label of the sequence. In total, SFEW contains 700 images that have been labeled for the seven facial expressions by two independent labelers. Figure 5.3 shows some sample images from this dataset. For the purpose of consistent evaluation of different algorithms, the images are divided into two sets of 346 images and 354 images, respectively. The sets are created in a strict person independent manner. Then, we use the first and second set to train and test our algorithms and vice versa.

In order to evaluate the performance of these algorithms, we have used the two metrics used for human action recognition: precision and Recall. Tables 5.11, 5.12, and 5.13 show the various precision and recall of all descriptor/Classifier combinations for the JAFFE, Cohn-Kanade, and SFEW databases, respectively.

From experimental results, we found that both WLD and PHOG descriptors achieve the highest results for all classifiers, and that both the AGM and AGGM models outperform the GMM and SVM for classification. It is noteworthy that the computational time excluding the pre-processing time of feature extraction of the AGM-EM-MML, AGM-RPEM, AGGM-EM-ML, and AGGM-RPEM approaches running on Core i7-2.4 GHz processor are (3, 2, 4, 2), (9, 7, 12, 10), and (5, 4, 7, 5) minutes for the training

	LBP		LPQ		HOG		PHOG		WLD	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
LSVM	75.1%	79.3%	**78.7%**	68.6%	78.5%	83.4%	74.9%	86.1%	75.7%	85.5%
PolySVM	75.9%	80.2%	65.3%	68.7%	81.1%	83.6%	82.7%	86.2%	76.3%	85.7%
RBFSVM	**78.3%**	80.8%	75.1%	69.0%	81.4%	83.6%	82.8%	86.4%	76.3%	85.7%
GMM-EM-ML	58.4%	76.9%	58.9%	67.3%	82.5%	80.6%	80.3%	83.6%	67.9%	82.0%
GMM-RPEM	58.4%	76.8%	61.3%	67.1%	82.2%	81.0%	80.3%	83.6%	68.1%	82.1%
AGM-EM-ML	69.4%	**81.2%**	72.0%	70.1%	**86.3%**	84.2%	83.2%	86.8%	**81.8%**	86.2%
AGM-RPEM	69.4%	**81.2%**	71.2%	69.9%	85.0%	84.5%	**84.3%**	86.9%	**81.8%**	86.3%
AGGM-EM-ML	70.7%	80.2%	71.5%	**70.3%**	86.2%	84.0%	83.4%	86.8%	81.5%	86.1%
AGGM-RPEM	71.4%	80.6%	70.7%	69.0%	85.3%	**85.7%**	84.0%	**88.3%**	81.6%	**86.5%**

Table 5.11: Average precision and recall (%) of all the descriptor/Classifier combinations for the JAFFE database.

	LBP		LPQ		HOG		PHOG		WLD	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
LSVM	87.1%	86.3%	68.3%	77.1%	61.8%	66.4%	**91.3%**	95.3%	94.6%	95.7%
PolySVM	86.2%	86.3%	69.0%	77.3%	61.8%	66.7%	85.3%	95.4%	94.6%	95.7%
RBFSVM	88.3%	**86.9%**	81.4%	77.4%	58.1%	67.2%	87.4%	95.9%	94.1%	**95.9%**
GMM-EM-ML	85.8%	81.2%	84.4%	76.6%	**63.4%**	62.3%	81.7%	92.0%	93.3%	92.6%
GMM-RPEM	85.8%	81.2%	84.4%	76.5%	**63.4%**	62.3%	81.7%	91.9%	92.1%	92.1%
AGM-EM-ML	**88.5%**	86.4%	**87.1%**	78.0%	59.3%	**67.8%**	88.1%	96.4%	97.2%	**95.9%**
AGM-RPEM	**88.5%**	86.5%	**87.1%**	78.0%	58.8%	67.4%	89.5%	**96.7%**	**97.6%**	95.7%
AGGM-EM-ML	88.3%	86.7%	87.0%	**78.2%**	59.4%	67.6%	89.4%	96.0%	97.3%	95.8%
AGGM-RPEM	88.1%	86.7%	87.0%	**78.2%**	59.5%	67.6%	89.9%	96.3%	97.2%	95.8%

Table 5.12: Average precision and recall (%) of all the descriptor/Classifier combinations for the Cohn- Kanade database.

	LBP		LPQ		HOG		PHOG		WLD	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
LSVM	43.6%	46.2%	29.9%	44.1%	37.4%	29.0%	44.5%	45.9%	41.5%	48.4%
PolySVM	44.0%	46.3%	29.9%	44.1%	37.4%	29.0%	44.5%	45.8%	48.3%	**48.6%**
RBFSVM	46.8%	46.7%	31.7%	44.2%	36.2%	**29.7%**	44.9%	46.1%	49.1%	**48.6%**
GMM-EM-ML	53.1%	42.5%	34.0%	41.3%	31.3%	27.8%	40.8%	42.9%	47.6%	45.7%
GMM-RPEM	54.2%	42.9%	34.0%	41.3%	31.6%	27.4%	40.7%	42.6%	47.5%	45.5%
AGM-EM-ML	55.3%	46.8%	**41.2%**	44.4%	37.7%	29.2%	**51.1%**	**46.3%**	**49.2%**	48.4%
AGM-RPEM	**55.5%**	**47.0%**	41.1%	**44.9%**	37.3%	28.9%	**51.1%**	46.2%	**49.2%**	48.4%
AGGM-EM-ML	55.4%	46.9%	41.0%	44.5%	37.6%	29.5%	51.0%	46.2%	**49.2%**	**48.6%**
AGGM-RPEM	55.4%	46.9%	39.9%	44.6%	**37.9%**	29.4%	**51.1%**	46.2%	**49.2%**	**48.6%**

Table 5.13: Average precision and recall (%) of all the descriptor/Classifier combinations for the SFEW database.

set of the JAFFE, Cohn-Kanade, and SFEW database, respectively.

5.6 Discussion

In this chapter, we have presented two approaches for clustering high dimensional data using mixture models. Our approaches consider that data arises from a mixture of asymmetric distributions which represents a good choice for high dimensional data due to its asymmetrical property and its ability to model different shapes. Furthermore, we tackled the problem of noisy and uninformative features by determining a set of relevant features for each data cluster. For model learning, the RPEM is used to allow simultaneous parameters estimation and model selection for the first approach and the expectation-maximization was penalized with the minimum message length criterion for the second approach. The merits of the proposed work are shown through complicated computer vision examples and applications, involving high-dimensional data and large number of classes, namely scene categorization and facial expression recognition. From experimental results, we can conclude that both algorithms are effective in clustering, however, we favor the RPEM due to its lower computational time.

Chapter 6

Conclusions

Clustering is the task of classifying patterns or observations into clusters or groups. Generally, clustering in high-dimensional feature spaces has a lot of complications such as: the unidentified or unknown data shape which is typically non-Gaussian and follows different distributions; the unknown number of clusters in the case of unsupervised learning; and the existence of noisy, redundant, or uninformative features which normally compromises modeling capabilities and speed. Therefore, high-dimensional data clustering has been a subject of extensive research in data mining, pattern recognition, image processing, and other areas for several decades. However, most of these researches tackle one or two problems at a time which is unrealistic because all problems are connected and should be tackled simultaneously.

In this thesis, first we have proposed a hierarchical infinite mixture model of generalized Gaussian distributions for visual learning based on non-parametric Bayesian estimation. The specific choice of infinite mixture models is motivated by the fact that they combine flexibility in modeling, clarity of interpretation and intuitive analysis which is crucial in statistical inference from image data generally supposed to be generated from different sources. We have shown that fully Bayesian models provide a rigorous framework for challenging applications due to its ability to handle uncertainties associated with the involved data, by incorporating prior knowledge, and to its deep foundation on probability inference. According to the results, it is clear that performing feature selection in tandem with infinite mixture models leads to excellent clustering results and avoids overfitting. The experiments show clearly the broad applicability and generality of the proposed approach which is able to infer at the same time both meaningful clusters and meaningful features.

Second, we have extended the application of mixture models to foreground subtraction. The proposed algorithm handles the problem of modeling the background and detecting the moving objects in the scene by adopting the use of the Asymmetric Gaussian mixtures. Our approach demonstrates that the assumption that the background and foreground distributions are Gaussian isn't always the case for most environments. Also, we introduced a novel approach for shadow detection by constructing an model capable to adapt in order to represent shadows in various video sequences.

Third, we have developed a framework for automatic surveillance and monitoring systems that takes into consideration fusion of both colour and thermal camera outputs in order to produce information otherwise not obtainable by viewing each sensor output separately. Also it demonstrates that modeling non-Gaussian data can be tackled by the use of the asymmetric generalized Gaussian distribution and the problem of determining the number of clusters can be overcame by the use of the Minimum Message Length criterion.

Last but not least, we have proposed two unified statistical frameworks based on finite asymmetric mixture models. Our approaches tackle simultaneously four of the critical issues that arise when clustering and modeling objects using finite mixture models: (1) determination of what features best discriminate among the different clusters, (2) choice of the probability density functions, (3) estimation of the mixture parameters and (4) automatic determination of the number of mixture. The first algorithm aims at finding the best overall model in the entire set of available models rather than selecting one among a set of candidate models by incorporating a Minimum Message Length (MML) penalty in the model learning step. The second algorithm learns the asymmetric models via an RPEM technique which allows simultaneous parameters estimation and model selection. Also, for both algorithms, we tackled the problem of noisy and uninformative features by determining a set of relevant features for each data cluster. The merits of the proposed work are shown through complicated computer vision examples and applications, involving high-dimensional data and large number of classes,namely scenes categorization and facial expression recognition.

In conclusion, compared to existing techniques, generally based on the Gaussian assumption, our approaches not only can model non-Gaussian data, but also can reach better approximation for the model parameters, and even a better selection for the number of clusters. Future works could be devoted to the development of a variational approach to learn the proposed models since it can offer a deterministic alternative for Bayesian approximate inference by maximizing a lower bound on the marginal likelihood. Variational frameworks have many advantages such as computational efficiency and guaranteed convergence that can be easily assessed as compared to MCMC-based approaches as they do not need calculations of high-dimensional integrals.

List of References

[1] C. Erdem, B. Sankur, and A.M. Tekalp. Performance measures for video object segmentation and tracking. *IEEE Transactions on Image Processing*, 13(7):937–951, 2004.

[2] C. Stauffer and W.E.L Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 252–258, 1999.

[3] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28–31, 2004.

[4] P. Kaewtrakulpong and R. Bowden. An improved adaptive background mixture model for realtime tracking with shadow detection. In *Workshop on Advanced Video Based Surveillance Systems*, pages 1–5, 2001.

[5] R.H. Evangelio, M. Patzold, and T. Sikora. Splitting gaussians in mixture models. In *IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*, pages 300–305, 2012.

[6] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *European Conference on Computer Vision (ECCV)*, pages 751–767, 2000.

[7] Y. Nonaka, A. Shimada, H. Nagahara, and R. Taniguchi. Evaluation report of integrated background modeling based on spatio-temporal features. In *IEEE*

Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 9–14, 2012.

[8] S.Y. Kung and M.W. Mak and S.H. Lin. *Biometrie Authentication: A Machine Learning Approach*. Prentice Hall, information and system sciences series edition edition, 2004.

[9] E. Osuna, R. Freund, and F. Girasi. Training support vector machines: an application to face detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 130–136, 1997.

[10] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[11] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Pattern Analysis andMachine Intelligence*, 28(1):84–95, 1980.

[12] G. Polatkan, S. Jafarpour, A. Brasoveanu, S. Hughes, and I. Daubechies. Detection of forgery in paintings using supervised learning. In *IEEE International Conference on Image Processing (ICIP)*, pages 2921–2924, 2009.

[13] M. Dorigo and U. Schnepf. Genetics-based machine learning and behaviour based robotics: A new synthesis. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(1):141–154, 1993.

[14] G. McLachlan and D. Peel. *Finite Mixture Models*. New York: Wiley, 2000.

[15] C.E. Rasmussen. The infinite gaussian mixture model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 554–560, 1999.

[16] C.P. Robert. *The Bayesian Choice From Decision-Theoretic Foundations to Computational Implementation*. Springer, 2007.

[17] M.B. Palacios and M.F.J. Steel. Non-gaussian bayesian geostatistical modeling. *Journal of the American Statistical Association*, 101(474):604–618, 2006.

[18] J. H. Miller and J. B. Thomas. Detectors for discrete-time signals in non-gaussian noise. *IEEE Transactions on Information Theory*, 18(2):241–250, 1972.

[19] N. Farvardin and J. W. Modestino. Optimum quantizer performance for a class of non-gaussian memoryless sources. *IEEE Transactions on Information Theory*, 30(3):485–497, 1984.

[20] T. Elguebaly and N. Bouguila. Bayesian learning of finite generalized gaussian mixture models on images. *Signal Processing*, 91(4):801–820, 2011.

[21] M.S. Allili, N. Bouguila, and D. Ziou. Finite general gaussian mixture modeling and application to image and video foreground segmentation. *Journal of Electronic Imaging*, 17(1):1–13, 2008.

[22] W. Mauersberger. Experimental results on the performance of mismatched quantizers. *IEEE Transactions on Information Theory*, 25(4):381–386, 1979.

[23] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[24] M.N. Do and M. Vetterli. Wavelet-based texture retrieval using generalized gaussian density and kullback-leibler distance. *IEEE Transactions on Image Processing*, 11(2):146–158, 2002.

[25] J.K. Ghosh and M. Delampady and T. Samanta. *An Introduction to Bayesian Analysis Theory and Methods*. Springer, 2006.

[26] N. Bouguila and D. Ziou. High-dimensional unsupervised selection and estimation of a finite generalized dirichlet mixture model based on minimum message length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1716–1731, 2007.

[27] M.S. Allili, D. Ziou, N. Bouguila, and S. Boutemedjet. Image and video segmentation by combining unsupervised generalized gaussian mixture modeling and feature selection. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(10):1373–1377, 2010.

[28] M.A.T. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):4–37, 2002.

[29] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[30] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1987.

[31] C.S. Wallace and D.M. Boulton. An information measure for classification. *The Computer Journal*, 11(2):195–209, 1968.

[32] M.A.T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

[33] L. Xu, A. Krzyzak, and E. Oja. Rival penalized competitive learning for clustering analysis. *IEEE Transactions on Neural Networks*, 4(4):636–646, 1993.

[34] Y.M. Cheung. Rival penalization controlled competitive learning for data clustering with unknown cluster number. In *International Conference on Neural Information Processing, Volume 2*, pages 18–22, 2002.

[35] Y.-M. Cheung. Maximum weighted likelihood via rival penalized em for density mixture clustering with automatic model selection. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):750–761, 2005.

[36] N. Bouguila and D. Ziou. A dirichlet process mixture of generalized dirichlet distributions for proportional data modeling. *IEEE Transactions on Neural Networks*, 21(1):107–122, 2010.

[37] R. Khardon and D. Roth. Model-based subspace clustering. *Artificial Intelligence*, 97(1-2):169–193, 1997.

[38] C. Cardie. A cognitive bias approach to feature selection and weighting for case-based learners. *Machine Learning*, 41(1):85–116, 2000.

[39] B. Heisele, T. Serre, S. Mukherjee, and T. Poggio. Feature reduction and hierarchy of classifiers for fast object detection in video images. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18–24, 2001.

[40] K. Kümmel, T. Scheidat, C. Vielhauer, and J. Dittmann. Feature selection on handwriting biometrics: security aspects of artificial forgeries. In *International*

Conference on Communications and Multimedia Security (CMS), pages 16–25, 2012.

[41] S. Boutemedjet, N. Bouguila, and D. Ziou. A hybrid feature extraction selection approach for high-dimensional non-gaussian data clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(8):1429–1443, 2009.

[42] H-L. Wei and S.A. Billings. Feature subset selection and ranking for data dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):162–166, 2007.

[43] M.H.C. Law, M.A.T. Figueiredo, and A.K. Jain. Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004.

[44] W. Pan and X. Shen. Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research*, 8:1145–1164, 2007.

[45] T. Elguebaly and N. Bouguila. Generalized gaussian mixture models as a nonparametric bayesian approach for clustering using class-specific visual features. *Journal of Visual Communication and Image Representation*, 23(8):1199–1212, 2012.

[46] T. Elguebaly and N. Bouguila. Background subtraction using finite mixtures of asymmetric gaussian distributions and shadow detection. *Machine Vision and Applications*, 25(5):1145–1162, 2014.

[47] T. Elguebaly and N. Bouguila. Finite asymmetric generalized gaussian mixture models learning for infrared object detection. *Computer Vision and Image Understanding*, 117(12):1659–1671, 2013.

[48] T. Elguebaly and N. Bouguila. Indoor scene recognition with a visual attention-driven spatial pooling strategy. In *IEEE Canadian Conference on Computer and Robot Vision (CRV)*, pages 268–275, 2014.

[49] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[50] S. Meignen and H. Meignen. On the modeling of small sample distributions

with generalized gaussian density in a maximum likelihood framework. *IEEE Transactions on Image Processing*, 15(6):1647–1652, 2006.

[51] J. Huang and D. Mumford. Statistics of natural images and models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 541–547, 1999.

[52] D.J. Field. What is the goal of sensory coding? *Neural Computation*, 6(4):559–601, 1994.

[53] A. Srivastava, A. Lee, E. Simoncelli, and S. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1):17–33, 2003.

[54] I. Laptev. Improving object detection with boosted histograms. *Image and Vision Computing*, 27(5):535–544, 2009.

[55] D. Farcas, C. Marghes, and T. Bouwmans. Background subtraction via incremental maximum margin criterion: a discriminative subspace approach. *Machine Vision and Applications*, 23(6):1083–1101, 2012.

[56] Z. Gao, B. Belzer, and J. Villasenor. A comparison of the z, e8, and leech lattices for quantization of low-shape-parameter generalized gaussian sources. *IEEE Signal Processing Letters*, 2(10):197–199, 1995.

[57] T. J. Hebert and K. Lu. Expectation-maximization algorithm, null spaces, and map image restoration. *IEEE Transactions on Image Processing*, 4(8):1084–1095, 1995.

[58] D. Madigan and A.E. Raftery. Model selection accounting for model unceratinity in graphical models using occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.

[59] D. Draper. Assessment propagation of model uncertainity (with discussion). *Journal of the Royal Statistical Society, Series B*, 57(1):45–97, 1995.

[60] H. Rue. New loss functions in bayesian imaging. *Journal of the American Statistical Association*, 90(431):900–908, 1995.

[61] T.J. Hebert and R. Leahy. Statistic-based map image reconstruction from

poisson data using gibbs priors. *IEEE Transactions on Signal Processing*, 40(9):2290–2303, 1992.

[62] S.J. Godsill. Bayesian enhancement of speech and audio signals which can be modelled as arma processes. *International Statistical Review*, 65(1):1–21, 1997.

[63] T. Elguebaly and N. Bouguila. Bayesian learning of generalized gaussian mixture models on biomedical images. In *Artificial Neural Networks in Pattern Recognition(ANNPR), Lecture Notes in Computer Science Volume 5998*, pages 207–218, 2010.

[64] P. Stoica and Y. Selen. Model-order selection: A review of information criterion rules. *IEEE Signal Processing Magazine*, 21(4):36–47, 2004.

[65] C.E. Rasmussen. The infinite gaussian mixture model. In *Advances in Neural Information Processing Systems (NIPS)*, pages 554–560, 2000.

[66] T.S. Ferguson. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

[67] M.D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[68] R.M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265, 2000.

[69] P.D. Hoff. Model-based subspace clustering. *Bayesian Analysis*, 1(2):321–344, 2006.

[70] E. Regazzinin, A. Guglielmi, and G. Di Nunno. Theory and numerical analysis for exact distributions of functionals of a dirichlet process. *The Annals of Statistics*, 30(5):1376–1411, 2002.

[71] A. Guglielmi, C.C. Holmes, and S.G. Walker. Perfect simulation involving functionals of a dirichlet processes. *Journal of Computational and Graphical Statistics*, 11(2):306–310, 2002.

[72] A. Rodriguez, D.B. Dunson, and A.E. Gelfand. The nested dirichlet process (with discussion). *Journal of the American Statistical Association*, 103(483):1131–1154, 2008.

[73] N. Bouguila and D. Ziou. A dirichlet process mixture of dirichlet distributions for classification and prediction. In *IEEE Workshop on Machine Learning for Signal Processing (MLSP)*, pages 297–302, 2008.

[74] T. Elguebaly and N. Bouguila. Infinite generalized gaussian mixture modeling and applications. In *International Conference on Image Analysis and Recognition (ICIAR), Lecture Notes in Computer Science Volume 6753*, pages 201–210, 2011.

[75] A.E. Gelfand and A. Kottas. A computational approach for full nonparametric bayesian inference under dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 11(2):289–305, 2002.

[76] J.K. Ghosh and R.V. Ramamoorthi. *Bayesian Nonparametrics*. Springer, 2003.

[77] S. Watanabe. *Pattern Recognition: Human and Mechanical*. New York: Wiley, 1985.

[78] Y. Li, M. Dong, and J. Hua. Simultaneous localized feature selection and model detection for gaussian mixtures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):953–960, 2009.

[79] J. Zhang and J.W. Modestino. A model-fitting approach to cluster validation with application to stochastic model-based image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):1009–1017, 1990.

[80] D.L. Ruderman and W. Bialek. Statistics of natural images: Scaling in the woods. *Physical Review Letters*, 73(6):814–818, 1994.

[81] J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *Artificial Intelligence*, 78(1-2):507–545, 1995.

[82] H. Zhang, W. Gao, X. Chen, and D. Zhao. Object detection using spatial histogram features. *Image and Vision Computing*, 24(4):327–341, 2006.

[83] A. Bosch, X. Muñoz, and J. Freixenet. Segmentation and description of natural outdoor scenes. *Image and Vision Computing*, 25(5):727–740, 2007.

[84] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *IEEE International Conference on Computer Vision (ICCV)*, pages 281–288, 2003.

[85] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 555–562, 1998.

[86] K. Levi and Y. Weiss. Learning object detection from a small number of examples: The importance of good features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 53–60, 2004.

[87] P. M. Baggenstoss. Class-specific feature sets in classification. *IEEE Transactions on Signal Processing*, 47(12):3428–3432, 1999.

[88] P.M. Baggenstoss and H. Niemann. A theoretically optimal probabilistic classifier using class-specific features. In *International Conference on Pattern Recognition (ICPR)*, pages 763–768, 2000.

[89] S. Baluja and D. Pomerleau. Dynamic relevance: Vision-based focus of attention using artificial neural networks. *Artificial Intelligence*, 97(1-2):381–395, 1997.

[90] D.B. Dunson. Bayesian nonparametric hierarchical modeling. *Biometrical Journal*, 51(2):273–284, 2009.

[91] P. Clifford. Discussion on the meeting on the gibbs sampler and other markov chain monte carlo methods. *Journal of the Royal Statistical Society*, 55(1):53–102, 1993.

[92] W.R. Gilks and P. Wild. Algorithm as 287: Adaptive rejection sampling from log-concave density functions. *Applied Statistics*, 42(4):701–709, 1993.

[93] J.S. Rosenthal. Minorization conditions and convergence rates for markov chain monte carlo. *Journal of the American Statistical Association*, 90(430):558–566, 1995.

[94] K.L. Mengersen and R.L. Tweedie. Rates of convergence of the hastings and metropolis algorithms. *Annals of Statistics*, 24(1):101–121, 1996.

[95] G.O. Roberts and J.S. Rosenthal. Convergence of slice sampler markov chains. *Journal of the Royal Statistical Society Series B*, 61(3):643–660, 1999.

[96] A.E. Raftery and S.M. Lewis. One long run with diagnostics: Implementation strategies for markov chain monte carlo. *Statistical Science*, 7(4):493–497, 1992.

[97] A.E. Raftery and S.M. Lewis. *Implementing MCMC, Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

[98] F. Cutzu, R. Hammoud, and A. Leykin. Estimating the photorealism of images: Distinguishing paintings from photographs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 305–312, 2003.

[99] F. Cutzu, R. Hammoud, and A. Leykin. Distinguishing paintings from photographs. *Computer Vision and Image Understanding*, 100(3):249–273, 2005.

[100] C. Wallraven, R.W. Fleming, D. W. Cunningham, J. Rigau, M. Feixas, and M. Sbert. Categorizing art: Comparing humans and computers. *Computers & Graphics*, 33(4):484–495, 2009.

[101] B. Gunsel, S. Sariel, and O. Icoglu. Content-based access to art paintings. In *IEEE International Conference on Image Processing (ICIP)*, pages 558–561, 2005.

[102] R. Hammoud. Color texture signatures for art-paintings vs. scene-photographs based on human visual system. In *International Conference on Pattern Recognition (ICPR)*, pages 525–528, 2004.

[103] V. Athitsos, M.J. Swain, and C. Frankel. Distinguishing photographs and graphics on the world wide web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 10–17, 1997.

[104] W. Chen, Y.Q. Shi, and G. Xuan. Identifying computer graphics using hsv color model and statistical moments of characteristic functions. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1123–1126, 2007.

[105] S. Lyu and H. Farid. How realistic is photorealistic? *IEEE Transactions on Signal Processing*, 53:845–850, 2005.

[106] J-P. Wang. Stochastic relaxation on partitions with connected components and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):619–636, 1998.

[107] F. Moscheni, S. Bhattacharjee, and M. Kunt. Spatiotemporal segmentation based on region merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):897–915, 1998.

[108] W-L. Hung, M-S. Yang, and D-H. Chen. Bootstrapping approach to feature-weight selection in fuzzy c-means algorithms with an application in color image segmentation. *Pattern Recognition Letters*, 29(9):1317–1325, 2008.

[109] S. Konishi and A. L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 125–132, 2000.

[110] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background substraction. In *European Conference on Computer Vision (ECCV)*, pages 751–767, 2000.

[111] M.S. Allili and D. Ziou. Globally adaptive region information for color-texture image segmentation. *Pattern Recognition Letters*, 28(15):1946–1956, 2007.

[112] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE International Conference on Computer Vision (ICCV)*, pages 416–423, 2001.

[113] Y. Raja, S.J. McKenna, and S. Gong. Colour model selection and adaption in dynamic scenes. In *European Conference on Computer Vision (ECCV)*, pages 460–474, 1998.

[114] M. Pantic and L.J.M. Rothkrantz. Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1424–1445, 2000.

[115] B. Fasel and J. Luettin. Automatic facial expression analysis: A survey. *Pattern Recognition*, 36(1):259–275, 2003.

[116] M. Pantic and L.J.M. Rothkrantz. An expert system for recognition of facial actions and their intensity. *Image and Vision Computing*, 18:881–905, 2000.

[117] Y.-L. Tian, T. Kanade, and J. Cohn. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):97–115, 2001.

[118] M.J. Lyons, J. Budynek, and S. Akamatsu. Automatic classification of single facial images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12):1357–1362, 1999.

[119] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999.

[120] D.A. Socolinsky, L.B. Wolff, J.D. Neuheisel, and C.K. Eveland. Illumination invariant face recognition using thermal infrared imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 527–534, 2001.

[121] F. Prokoski. History, current status, and future of infrared identification. In *IEEE Workshop on Computer Vision Beyond the Visible Spectrum: Methods and Applications (CVBVS)*, page 5, 2000.

[122] L. Trujillo, G. Olague, R.I. Hammoud, and B. Hernandez. Automatic feature localization in thermal images for facial expression recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 14, 2005.

[123] G. Olague, R.I. Hammoud, L. Trujillo, B. Hernandez, and E. Romero. Facial expression recognition in nonvisual imagery. In *Augmented Vision Perception in Infrared*, pages 213–238, 2009.

[124] B. Hern*á*ndez, G. Olague, R.I. Hammoud, L. Trujillo, and E. Romero. Visual learning of texture descriptors for facial expression recognition in thermal imagery. *Computer Vision and Image Understanding*, 106(2-3):258–269, 2007.

[125] A. Strehl and J. Ghosh. Cluster ensembles a knowledge reuse framework for

combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

[126] A. Tavakkoli, M. Nicolescu, G. Bebis, and M. Nicolescu. Non-parametric statistical background modeling for efficient foreground region detection. *Machine Vision and Applications*, 7(2):1–15, 2009.

[127] 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.

[128] R.G. Abbott and L.R. Williams. Multiple target tracking with lazy background subtraction and connected components analysis. *Machine Vision and Applications*, 20(2):93–101, 2009.

[129] L.M. Fuentes and S.A. Velastin. People tracking in surveillance applications. *Image and Vision Computing*, 24(11):1165–1171, 2006.

[130] Y.L. Tian, A. Senior, and M. Lu. Robust and efficient foreground analysis in complex surveillance videos. *Machine Vision and Applications*, 23(5):967–983, 2012.

[131] A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 302–309, 2004.

[132] Y. Ren, C.-S. Chua, and Y.-K. Ho. Motion detection with nonstationary background. *Machine Vision and Applications*, 13(5-6):332–343, 2003.

[133] F. El Baf, T. Bouwmans, and B. Vachon. Comparison of background subtraction methods for a multimedia learning space. In *International Conference on Signal Processing and Multimedia Applications (SIGMAP)*, pages 153–158, 2007.

[134] T. Bouwmans, F. El Baf, and B. Vachon. Background modeling using mixture of gaussians for foreground detection - a survey. *Recent Patents on Computer Science*, 1(3):219–237, 2008.

[135] C. Stauffer and W.E.L Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[136] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. In *Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 1–3, 1997.

[137] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):827–832, 2005.

[138] D. Wang, W. Xie, J. Pei, and Z. Lu. Moving area detection based on estimation of static background. *Journal of Information and Computing Science*, 2(1):129–134, 2005.

[139] T. Elguebaly and N. Bouguila. A nonparametric bayesian approach for enhanced pedestrian detection and foreground segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPRW)*, pages 21–26, 2011.

[140] R.A. Baxter and J.J. Olivier. Finding overlapping components with mml. *Statistical Computing*, 10(1):5–16, 2000.

[141] W.E.L. Grimson, C. Stauffer, R. Romano, and L. Lee. Using adaptive tracking to classify and monitor activities in a site. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22–29, 1998.

[142] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti. Improving shadow suppression in moving object detection with hsv color information. In *IEEE Intelligent Transportation Systems*, pages 334–339, 2001.

[143] O. Schreer, I. Feldmann, U. Goelz, and P. Kauff. Fast and robust shadow detection in videoconference applications. In *IEEE International Symposium on Video/Image Processing and Multimedia Communications*, pages 371–375, 2002.

[144] E. Salvador, A. Cavallaro, and T. Ebrahimi. Cast shadow segmentation using invariant color features. *Computer Vision and Image Understanding*, 95(2):238–259, 2004.

[145] T. Horprasert, D. Hardwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–19, 1999.

[146] SW. Zhang, X.Z. Fang, and X.K. Yang. Moving cast shadows detection using ratio edge. *IEEE Transactions on Multimedia*, 9(6):1202–1214, 2007.

[147] J.M. Choi, Y.J. Yoo, and J.Y. Choi. Adaptive shadow estimator for removing shadow of moving object. *Computer Vision and Image Understanding*, 114(9):1017–1029, 2010.

[148] G.D. Finlayson, S.D. Hordley, C. Lu, and M.S. Drew. On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006.

[149] N. Martel-Brisson and A. Zaccarin. Learning and removing cast shadows through a multidistribution approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1133–1146, 2007.

[150] Z. Liu, K. Huang, T. Tan, and L. Wang. Cast shadow removal combining local and global features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[151] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar. changedetection.net: A new change detection benchmark dataset. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1–8, 2012.

[152] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance application. *IEEE Transactions on Image Processing*, 17(7):1168–1177, 2008.

[153] E.W. Kerce. *Boredom at Work: Implications for the Design of Jobs with Variable Requirements*. Navy Research, 1985.

[154] L. St-Laurent, X. Maldague, and D. Prevost. Combination of colour and thermal sensors for enhanced object detection. In *International Conference on Information Fusion*, pages 1–8, 2007.

[155] F. De la Torre, E. Martinez, M.E. Santamaria, and J.A. Moran. Moving object detection and tracking system : a real-time implementation. In *Symposium on Signal and Image Processing*, pages 375–378, 1997.

[156] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2000.

[157] D.A. Socolinsky, A. Selinger, and J.D. Neuheisel. Face recognition with visible and thermal infrared imagery. *Computer Vision and Image Understanding*, 91(1-2):72–114, 2003.

[158] C. Dai, Y. Zheng, and X. Li. Pedestrian detection and tracking in infrared imagery using shape and appearance. *Computer Vision and Image Understanding*, 106(2-3):288–299, 2007.

[159] I. Pavlidis and P. Symosek. The imaging issue in an automatic face/disguise detection system. In *Computer Vision Beyond the Visible Spectrum*, pages 15–24, 2000.

[160] D.B. Rensch and R.K. Long. Comparative studies of extinction and backscattering by aerosols, fog, and rain at 10.6μ and 0.63μ. *Applied Optic*, 9(7):1563–1573, 1970.

[161] M.A. Naboulsi, H. Sizun, and F. de Fornel. Fog attenuation prediction for optical and infrared waves. *Optical Engineering*, 43(2):319–329, 2004.

[162] A. Torabi, G. Masse, and G.-A. Bilodeau. Feedback scheme for thermal-visible video registration, sensor fusion, and people tracking. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 15–22, 2010.

[163] A. Leykin and R. Hammoud. Pedestrian tracking by fusion of thermal-visible surveillance videos. *Machine Vision and Applications*, 21(4):587–595, 2008.

[164] O. Tuzel, F.M. Porikli, and P. Meer. Pedestrian detection via classification on riemannian manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1713–1727, 2008.

[165] J. Davis and M. Keck. A two-stage template approach to person detection in thermal imagery. In *IEEE Workshops on Applications of Computer Vision, WACV/MOTIONS*, pages 364–369, 2005.

[166] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[167] D. Rowe, I. Reid, J. Gonzàlez, and J.J. Villanueva. Unconstrained multiplepeople tracking. In *Lecture Notes in Computer Science*, pages 505–514, 2006.

[168] J. Davis and V. Sharma. Background-subtraction using contour-based fusion of thermal and visible imagery. *Computer Vision and Image Understanding*, 106(2-3):162–182, 2007.

[169] A. Shimada, D. Arita, and R. Taniguchi. Dynamic control of adaptive mixture-of-gaussians background model. In *IEEE International Conference on Video and Signal Based Surveillance*, page 5, 2006.

[170] R. Tan, H. Huo, J. Qian, and T. Fang. Traffic video segmentation using adaptive-k gaussian mixture model. In *International Conference on Intelligent Computing in Pattern Analysis/Synthesis*, pages 125–134, 2006.

[171] R. Collins, Y. Liu, and M. Leordeanu. Online selection of discriminative tracking features. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 27(10):1631–1643, 2005.

[172] A.K. Jain. Advances in mathematical models for image processing. In *Proceedings of the IEEE*, pages 502–528, 1981.

[173] W. Choi, K. Shahid, and S. Savarese. Learning context for collective activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3273–3280, 2011.

[174] M.J. Black and D.J. Fleet. Probabilistic detection and tracking of motion boundaries. *International Journal of Computer Vision*, 38(3):231–245, 2000.

[175] N. Bouguila and W. ElGuebaly. Discrete data clustering using finite mixture models. *Pattern Recognition*, 42(1):33–42, 2009.

[176] G.J. McLachlan and D. Peel. Mixfit: an algorithm for the automatic fitting and testing of normal mixture models. In *International Conference on Pattern Recognition (ICPR), Volume 1*, pages 553–557, 1998.

[177] J. Goldberger, H. Greenspan, and J. Dreyfuss. Simplifying mixture models using the unscented transform. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 30(8):1496–1502, 2008.

[178] R.P. Browne, P.D. McNicholas, and M.D. Sparling. Model-based learning using a mixture of mixtures of gaussian and uniform distributions. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 34(4):814–817, 2012.

[179] M.R. Whiteley, B.M. Welsh, and M.C. Roggemann. Limitations of gaussian assumptions for the irradiance distribution in digital imagery: Nonstationary image ensemble considerations. *Journal of the Optical Society of America. A*, 15(4):802–810, 1998.

[180] A. Hyvärinen and P.O. Hoyer. Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

[181] P.N. Bennett. Using asymmetric distributions to improve text classifier probability estimates. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 111–118, 2003.

[182] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. New York: Wiley, 1997.

[183] L. Xu. Rival penalized competitive learning, finite mixture, and multisets clustering. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, pages 2525–2530, 1998.

[184] A. Kolcz, X. Sun, and J.K. Kalita. Efficient handling of high-dimensional feature spaces by randomized classifier ensembles. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 307–313, 2002.

[185] E. Bart and S. Ullman. Cross-generalization: Learning novel classes from a single example by feature replacement. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 672–679, 2005.

[186] T. Kinh and P. Viola. Boosting image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 228–235, 2000.

[187] C.-Y. Tsai and C.-C. Chiu. Developing a feature weight self-adjustment mechanism for a k-means clustering algorithm. *Computational Statistics & Data Analysis*, 52(10):4658–4672, 2008.

[188] Y.S. Kim, W.N. Street, and F. Menczer. Feature selection in unsupervised learning via evolutionary search. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 365–369, 2000.

[189] Y.-M. Cheung and H. Zeng. Feature weighted rival penalized em for gaussian mixture clustering: Automatic feature and model selections in a single paradigm. In *International Conference on Computational Intelligence and Security (CIS)*, pages 1018–1028, 2006.

[190] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. What do we see in a glance of a scene? *Journal of Vision*, 7(1):1–29, 2007.

[191] Z. Su, H.-J. Zhang, S.Li, and S.Ma. Relevance feedback in content-based image retrieval: Bayesian framework, features subspaces, and progressive learning. *IEEE Trans. on Image Processing*, 12(8):924–937, 2003.

[192] S. Boutemedjet, D. Ziou, and N. Bouguila. A graphical model for content based image suggestion and feature selection. In *Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 30–41, 2007.

[193] A. Singhal, J. Luo, and W. Zhu. Probabilistic spatial context models for scene content understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Volume 1*, pages 235–241, 2003.

[194] Y. LeCun, J.H. Fu, and L. Bottou. Probabilistic spatial context models for scene content understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Volume 2*, pages 97–104, 2004.

[195] Y. Wu, I. Kozintsev, J.-Y Bouguet, and C. Dulong. Sampling strategies for active learning in personal photo retrieval. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 529–532, 2006.

[196] N. Rasiwasia, J.C. Pereira, E. Coviello, G. Doyle, G.R.G. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *International Conference on Multimedia (MM)*, pages 251–260, 2010.

[197] A. Gupta and L.S. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *European Conference on Computer Vision (ECCV)*, pages 16–29, 2008.

[198] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2033–2040, 2006.

[199] A. Rocha and S. Goldenstein. Progressive randomization: Seeing the unseen. *Computer Vision and Image Understanding*, 114(3):349–362, 2010.

[200] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision (ECCV)*, pages 1–22, 2004.

[201] L. Fei-Fei, A. Iyer, C. Koch, and P. Perona. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.

[202] N. Serrano, A. Savakis, and J. Luo. Improved scene classification using efficient low-level features and semantic cues. *Pattern Recognition*, 37(9):1773–1784, 2004.

[203] A. Vailaya, M. Figueiredo, A. Jain, and H.-J. Zhang. Image classication for content-based indexing. *IEEE Transactions On Image Processing*, 10(1):117–130, 2001.

[204] M. Szummer and R.W. Picard. Indoor-outdoor image classification. In *International Workshop on Content-Based Access of Image and Video Databases (CAIVD)*, page 42, 1998.

[205] L. Lu, K. Toyama, and G.D. Hager. A two level approach for scene recognition.

In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 688–695, 2005.

[206] L.-J. Li and L. Fei-Fei. What, where and who? classifying event by scene and object recognition. In *IEEE International Conference in Computer Vision (ICCV)*, pages 1–8, 2007.

[207] A. Bosch, A. Zisserman, and X. Muoz. Scene classification via plsa. In *Workshop on Statistical Learning in Computer Vision, European Conference on Computer Vision (ECCV)*, pages 517–530, 2006.

[208] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 524–531, 2005.

[209] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178, 2006.

[210] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.

[211] M. Blei and L. Fei-Fei. Simultaneous image classification and annotation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 1903–1910, 2009.

[212] D. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1150–1157, 1999.

[213] L. Fei-Fei and E.P. Xing. Large margin learning of up-stream scene understanding models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2586–2594, 2010.

[214] L.-J. Li, H. Su, E.P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1378–1386, 2010.

[215] C. Darwin. *The Expression of the Emotions in Man and Animals*. D. Appleton and Company, New York, 1872.

[216] M. Suwa, N. Sugie, and K. Fujimora. A preliminary note on pattern recognition of human emotional expression. In *Joint Conference on Pattern Recognition*, pages 408–410, 1987.

[217] C. Shana, S. Gong, and P. W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803–816, 2009.

[218] K. Yu, Z. Wang, L. Zhuo, J. Wang, Z. Chi, and D. Feng. Learning realistic facial expressions from web images. *Pattern Recognition*, 46(8):2144–2155, 2013.

[219] A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions On Pattern Analysis and Machine intelligence*, 20(1):23–38, 1996.

[220] Y. Li, S. Gong, J. Sherrah, and H. Liddell. Support vector machine based multi-view face detection and recognition. *Image and Vision Computing*, 22(5):413–427, 2004.

[221] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2106–2112, 2011.

[222] P. Viola and M.J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[223] M.S. Bartlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel, and J. Movellan. Recognizing facial expression: machine learning and application to spotaneous behavior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 568–573, 2005.

[224] M.S. Bartlett, G. Littlewort, I. Fasel, and R. Movellan. Real time face detection and facial expression recognition: development and application to human

computer interaction. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, page 53, 2003.

[225] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[226] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[227] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. In *International Conference on Image and Signal Processing (ICISP)*, pages 236–243, 2008.

[228] V. Ojansivu, E. Rahtu, and J. Heikkilä. Rotation invariant blur insensitive texture analysis using local phase quantization. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008.

[229] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[230] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *ACM International Conference on Image and Video Retrieval (CIVR)*, pages 401–408, 2007.

[231] J. Chen, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao. Wld: a robust local image descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1705–1720, 2010.

[232] X. Feng, M. Pietikinen, and T. Hadid. Facial expression recognition with local binary patterns and linear programming. *Pattern Recognition and Image Analysis*, 15(2):546–548, 2005.

[233] A. Dhall, R. Goecke, S. Lucey, and Tom Gedeon. Collecting large, richly annotated facial-expression databases from movies. *IEEE MultiMedia*, 19(3):34–41, 2012.

Appendix A

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

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

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

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

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

$$\frac{\partial^2 L(\Theta, Z, \mathcal{X})}{\partial \sigma^2_{r_{jk}}} = \sum_{i=1}^{N} \frac{\hat{Z}_{ij}}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} - 3\sum_{i=1, X_{ik}\geq\mu_{jk}}^{N} \frac{\hat{Z}_{ij}(X_{ik}-\mu_{jk})^2}{\sigma^4_{r_{jk}}} \tag{A.5}$$

Appendix B

In this appendix, we develop the solutions for equations (3.24, 3.25, 3.26) used in the MML algorithm

$$-\frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \mu_{jk}^2} = \sum_{i=l, X_{ik}<\mu_{jk}}^{l+n_j-1} \frac{1}{\sigma_{l_{jk}}^2} + \sum_{i=l, X_{ik}\geq\mu_{jk}}^{l+n_j-1} \frac{1}{\sigma_{r_{jk}}^2} \tag{B.1}$$

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

$$\frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \sigma_{l_{jk}}^2} = \sum_{i=l}^{l+n_j-1} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})^2} - 3 \sum_{i=l, X_{ik}<\mu_{jk}}^{l+n_j-1} \frac{(X_{ik} - \mu_{jk})^2}{\sigma_{l_{jk}}^4} \tag{B.3}$$

$$\frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \sigma_{l_{jk_1}}\sigma_{l_{jk_2}}} = 0 \tag{B.4}$$

$$\frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \sigma_{r_{jk}}^2} = \sum_{i=l}^{l+n_j-1} \frac{1}{(\sigma_{l_{jk}} + \sigma_{r_{jk}})^2} - 3 \sum_{i=l, X_{ik}\geq\mu_{jk}}^{l+n_j-1} \frac{(X_{ik} - \mu_{jk})^2}{\sigma_{r_{jk}}^4} \tag{B.5}$$

$$\frac{\partial^2 \log(p(\mathcal{X}_j|\Theta))}{\partial \sigma_{r_{jk_1}}\sigma_{r_{jk_2}}} = 0 \tag{B.6}$$

Appendix C

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Appendix D

The gradients with respect to $\sigma_{l_{jk}}$ and $\sigma_{r_{jk}}$ for the AGM model:

$$\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{l_{jk}}} = \sum_{i=1}^{N} r_{ijk} \vartheta_{l_{ijk}} \tag{D.1}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma^2_{l_{jk}}} = \sum_{i=1}^{N} r_{ijk} \vartheta_{l_{ijk}} \left[\varrho_{l_{ijk}} + \vartheta_{l_{ijk}} \frac{(1-\omega_k) p(X_{ik}|\lambda_k)}{\zeta_{ijk}} \right] \tag{D.2}$$

$$\frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma_{r_{jk}}} = \sum_{i=1}^{N} r_{ijk} \vartheta_{r_{ijk}} \tag{D.3}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \sigma^2_{r_{jk}}} = \sum_{i=1}^{N} r_{ijk} \vartheta_{r_{ijk}} \left[\varrho_{r_{ijk}} + \vartheta_{r_{ijk}} \frac{(1-\omega_k) p(X_{ik}|\lambda_k)}{\zeta_{ijk}} \right] \tag{D.4}$$

where

$$\vartheta_{l_{ijk}} = \begin{cases} \frac{(X_{ik}-\mu_{jk})^2}{\sigma^3_{l_{jk}}} - \frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})} & \text{if } X_{ik} < \mu_{jk} \\ -\frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})} & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{D.5}$$

$$\vartheta_{r_{ijk}} = \begin{cases} -\frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})} & \text{if } X_{ik} < \mu_{jk} \\ \frac{(X_{ik}-\mu_{jk})^2}{\sigma^3_{r_{jk}}} - \frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})} & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{D.6}$$

$$\varrho_{l_{ijk}} = \begin{cases} \frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} - \frac{3(X_{ik}-\mu_{jk})^2}{\sigma^4_{l_{jk}}} & \text{if } X_{ik} < \mu_{jk} \\ \frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{D.7}$$

$$\varrho_{r_{ijk}} = \begin{cases} \frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} & \text{if } X_{ik} < \mu_{jk} \\ \frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} - \frac{3(X_{ik}-\mu_{jk})^2}{\sigma^4_{r_{jk}}} & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{D.8}$$

The gradients with respect to μ_{jk}, β_{jk}, $\sigma_{l_{jk}}$, and $\sigma_{r_{jk}}$ for the AGGM model:

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \mu_{jk}} &= \sum_{i=1}^{N} r_{ijk} A(\beta_{jk}) \\ &\times \begin{cases} -\frac{\beta_{jk}}{\sigma_{l_{jk}}}\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta_{jk}-1} & \text{if } X_{ik} < \mu_{jk} \\ \frac{\beta_{jk}}{\sigma_{r_{jk}}}\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}-1} & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \end{aligned} \tag{D.9}$$

$$\begin{aligned} \frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \mu^2_{jk}} &= -\sum_{i=1}^{N} r_{ijk} A(\beta_{jk}) \beta_{jk} \\ &\times \begin{cases} \frac{\beta_{jk}-1}{\sigma^2_{l_{jk}}}\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta_{jk}-2} & \text{if } X_{ik} < \mu_{jk} \\ \frac{\beta_{jk}-1}{\sigma^2_{r_{jk}}}\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}-2} & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \end{aligned} \tag{D.10}$$

$$\begin{aligned} \frac{\partial \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \beta_{jk}} &= \sum_{i=1}^{N} r_{ijk} \\ &\times \left\{ \frac{1}{\beta_{jk}} + \frac{3}{2}\left(\frac{\psi(1/\beta_{jk}) - \psi(3/\beta_{jk})}{\beta^2_{jk}}\right) - f_{ijk} \right\} \end{aligned} \tag{D.11}$$

$$\begin{aligned} \frac{\partial^2 \mathcal{L}(\mathcal{X}, \Theta_M, Z, \varphi)}{\partial \beta^2_{jk}} &= \sum_{i=1}^{N} r_{ijk} \bigg\{ \frac{-1}{\beta^2_{jk}} \\ &+ \frac{3}{2}\left(\frac{3\psi'(3/\beta_{jk}) - \psi'(1/\beta_{jk})}{\beta^4_{jk}}\right) - 3\left(\frac{\psi(1/\beta_{jk}) - \psi(3/\beta_{jk})}{\beta^3_{jk}}\right) - t_{ijk} \bigg\} \end{aligned} \tag{D.12}$$

$$\frac{\partial \mathcal{L}(\mathcal{X},\Theta_M,Z,\varphi)}{\partial \sigma_{l_{jk}}} = \sum_{i=1}^{N} r_{ijk}\bigg\{ o_{l_{ijk}} - \frac{1}{\sigma_{l_{jk}}+\sigma_{r_{jk}}}\bigg\} \tag{D.13}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X},\Theta_M,Z,\varphi)}{\partial \sigma_{l_{jk}}^2} = \sum_{i=1}^{N} r_{ijk}\bigg\{ \frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} - s_{l_{ijk}}\bigg\} \tag{D.14}$$

$$\frac{\partial \mathcal{L}(\mathcal{X},\Theta_M,Z,\varphi)}{\partial \sigma_{r_{jk}}} = \sum_{i=1}^{N} r_{ijk}\bigg\{ o_{r_{ijk}} - \frac{1}{\sigma_{l_{jk}}+\sigma_{r_{jk}}}\bigg\} \tag{D.15}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X},\Theta_M,Z,\varphi)}{\partial \sigma_{r_{jk}}^2} = \sum_{i=1}^{N} r_{ijk}\bigg\{ \frac{1}{(\sigma_{l_{jk}}+\sigma_{r_{jk}})^2} - s_{r_{ijk}}\bigg\} \tag{D.16}$$

where

$$f_{ijk} = A(\beta_{jk}) c_{ijk}^{\beta_{jk}} \bigg\{ \ln c_{ijk} + d_{jk} + e_{jk} \bigg\} \tag{D.17}$$

$$t_{ijk} \;\; = \;\; A(\beta_{jk}) c_{ijk}^{\beta_{jk}} \bigg\{ \frac{9\psi'(3/\beta_{jk}) - \psi'(1/\beta_{jk})}{2\beta_{jk}^3} + \bigg[\ln c_{ijk} + d_{jk} + e_{jk} \bigg]^2 \bigg\} \tag{D.18}$$

$$o_{l_{ijk}} = \begin{cases} A(\beta_{jk}) \frac{\beta_{jk}}{\sigma_{l_{jk}}} c_{ijk}^{\beta_{jk}} & \text{if } X_{ik} < \mu_{jk} \\ 0 & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{D.19}$$

$$o_{r_{ijk}} = \begin{cases} 0 & \text{if } X_{ik} < \mu_{jk} \\ A(\beta_{jk}) \frac{\beta_{jk}}{\sigma_{r_{jk}}} c_{ijk}^{\beta_{jk}} & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{D.20}$$

$$s_{l_{ijk}} = o_{l_{ijk}} \frac{(\beta_{jk}+1)}{\sigma_{l_{jk}}} \tag{D.21}$$

$$s_{r_{ijk}} = o_{r_{ijk}} \frac{(\beta_{jk} + 1)}{\sigma_{r_{jk}}} \tag{D.22}$$

and

$$c_{ijk} = \begin{cases} \left(\frac{\mu_{jk} - X_{ik}}{\sigma_{l_{jk}}}\right) & \text{if } X_{ik} < \mu_{jk} \\ \left(\frac{X_{ik} - \mu_{jk}}{\sigma_{r_{jk}}}\right) & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{D.23}$$

$$d_{jk} = \frac{\psi(1/\beta_{jk}) - 3\psi(3/\beta_{jk})}{2\beta_{jk}} \tag{D.24}$$

$$e_{jk} = \frac{1}{2} \ln \left(\frac{\Gamma(3/\beta_{jk})}{\Gamma(1/\beta_{jk})} \right) \tag{D.25}$$

Appendix E

The derivations of $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \mu_{jk}}$, $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial S_{l_{jk}}}$, and $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial S_{r_{jk}}}$ for the AGM model:

$$\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \mu_{jk}} = p(X_{ik}|\xi_{jk}^{old})V_{ijk} \tag{E.1}$$

$$\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial S_{l_{jk}}} = p(X_{ik}|\xi_{jk}^{old})W_{l_{ijk}} \tag{E.2}$$

$$\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial S_{r_{jk}}} = p(X_{ik}|\xi_{jk}^{old})W_{r_{ijk}} \tag{E.3}$$

$$V_{ijk} = \begin{cases} S_{l_{jk}}(X_{ik} - \mu_{jk}) & \text{if } X_{ik} < \mu_{jk} \\ S_{r_{jk}}(X_{ik} - \mu_{jk}) & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{E.4}$$

$$W_{l_{ijk}} = \begin{cases} \frac{1}{2}\left[\frac{S_{r_{jk}}^{1/2}}{S_{l_{jk}}(S_{l_{jk}}^{1/2}+S_{r_{jk}}^{1/2})} - (X_{ik} - \mu_{jk})^2\right] & \text{if } X_{ik} < \mu_{jk} \\ \frac{S_{r_{jk}}^{1/2}}{2S_{l_{jk}}(S_{l_{jk}}^{1/2}+S_{r_{jk}}^{1/2})} & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{E.5}$$

$$W_{r_{ijk}} = \begin{cases} \frac{S_{l_{jk}}^{1/2}}{2S_{r_{jk}}(S_{l_{jk}}^{1/2}+S_{r_{jk}}^{1/2})} & \text{if } X_{ik} < \mu_{jk} \\ \frac{1}{2}\left[\frac{S_{l_{jk}}^{1/2}}{S_{r_{jk}}(S_{l_{jk}}^{1/2}+S_{r_{jk}}^{1/2})} - (X_{ik} - \mu_{jk})^2\right] & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{E.6}$$

The derivations of $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \mu_{jk}}$, $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \beta_{jk}}$, $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \sigma_{l_{jk}}}$, and $\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \sigma_{r_{jk}}}$ for the AGGM model

$$\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \mu_{jk}} = p(X_{ik}|\xi_{jk}^{old})\kappa_{ijk} \tag{E.7}$$

$$\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \beta_{jk}} = p(X_{ik}|\xi_{jk}^{old})\rho_{ijk} \tag{E.8}$$

$$\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \sigma_{l_{jk}}} = p(X_{ik}|\xi_{jk}^{old})\tau_{l_{ijk}} \tag{E.9}$$

$$\frac{\partial p(X_{ik}|\xi_{jk}^{old})}{\partial \sigma_{r_{jk}}} = p(X_{ik}|\xi_{jk}^{old})\tau_{r_{ijk}} \tag{E.10}$$

and

$$\kappa_{ijk} = \begin{cases} -A(\beta_{jk})\frac{\beta_{jk}}{\sigma_{l_{jk}}}\left(\frac{\mu_{jk}-X_{ik}}{\sigma_{l_{jk}}}\right)^{\beta_{jk}-1} & \text{if } X_{ik} < \mu_{jk} \\ A(\beta_{jk})\frac{\beta_{jk}}{\sigma_{r_{jk}}}\left(\frac{X_{ik}-\mu_{jk}}{\sigma_{r_{jk}}}\right)^{\beta_{jk}-1} & \text{if } X_{ik} \geq \mu_{jk} \end{cases} \tag{E.11}$$

$$\rho_{ijk} = \frac{1}{\beta_{jk}} + \frac{3}{2}\left(\frac{\psi(1/\beta_{jk}) - \psi(3/\beta_{jk})}{\beta_{jk}^2}\right) - f_{ijk} \tag{E.12}$$

$$\tau_{l_{ijk}} = o_{l_{ijk}} - \frac{1}{\sigma_{l_{jk}} + \sigma_{r_{jk}}} \tag{E.13}$$

$$\tau_{r_{ijk}} = o_{r_{ijk}} - \frac{1}{\sigma_{l_{jk}} + \sigma_{r_{jk}}} \tag{E.14}$$