Gaussian Mixture Model based Spatial Information Concept for Image Segmentation

Thanh Nguyen

University of Windsor

Follow this and additional works at: https://scholar.uwindsor.ca/etd

Recommended Citation

https://scholar.uwindsor.ca/etd/438

This online database contains the full-text of PhD dissertations and Masters’ theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000 ext. 3208.
Gaussian Mixture Model based Spatial Information Concept for Image Segmentation

by

Thanh Minh Nguyen

A Dissertation
Submitted to the Faculty of Graduate Studies
through Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada

2011

© 2011 Thanh Minh Nguyen
Author’s Declaration of Originality

I hereby declare that I am the sole author of this thesis. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.
Abstract

Segmentation of images has found widespread applications in image recognition systems. Over the last two decades, there has been a growing research interest in model-based technique. In this technique, standard Gaussian mixture model (GMM) is a well-known method for image segmentation. The model assumes a common prior distribution, which independently generates the pixel labels. In addition, the spatial relationship between neighboring pixels is not taken into account of the standard GMM. For this reason, its segmentation result is sensitive to noise. To reduce the sensitivity of the segmented result with respect to noise, Markov Random Field (MRF) models provide a powerful way to account for spatial dependencies between image pixels. However, their main drawback is that they are computationally expensive to implement.

Based on these considerations, in the first part of this thesis (Chapter 4), we propose an extension of the standard GMM for image segmentation, which utilizes a novel approach to incorporate the spatial relationships between neighboring pixels into the standard GMM. The proposed model is easy to implement and compared with the existing MRF models, requires lesser number of parameters. We also propose a new method to estimate the model
parameters in order to minimize the higher bound on the data negative log-likelihood, based on the gradient method. Experimental results obtained on noisy synthetic and real world grayscale images demonstrate the robustness, accuracy and effectiveness of the proposed model in image segmentation.

In the final part of this thesis (Chapter 5), another way to incorporate spatial information between the neighboring pixels into the GMM based on MRF is proposed. In comparison to other mixture models that are complex and computationally expensive, the proposed method is robust and fast to implement. In mixture models based on MRF, the M-step of the EM algorithm cannot be directly applied to the prior distribution for maximization of the log-likelihood with respect to the corresponding parameters. Compared with these models, our proposed method directly applies the EM algorithm to optimize the parameters, which makes it much simpler. Finally, our approach is used to segment many images with excellent results.
To my beloved Dad and Mom

&

My beloved wife, Dang Nguyen Thanh Yen
Acknowledgements

I am deeply indebted to my thesis advisor, Prof. Jonathan Wu, for his informed guidance, understanding, support and encouragement through the years. His depth and vision in this area are truly admirable, and he has been an inspiration to me in many ways.

I am grateful to my committee members Prof. Narayan Kar, Prof. Huapeng Wu, Prof. Dan Wu, and Prof. Peter X. Liu for reviewing my thesis and making many excellent suggestions to improve its substance and presentation. Their openness, encouragement and interest in my research were very important to me.

Thanks also go to Andria Ballo and Shelby Marchand from Department of Electrical and Computer Engineering for their reminders and assistance in various administrative tasks.

I am grateful that during the past four years, I get to know many other talented graduate students: Siddhant Ahuja, Mohammed Golam Sarwer, Baradarani Aryaz, Rashid Minhas, AbdulAdeel Mohammed, Dibyendu Mukherjee, and Ashirbani Saha. I want to thank them for their help, support, interest, and valuable hints. I would also like to thank other members
in our lab for encouragements and the joys and tears that we shared.

I thank my parents who always stand beside me. I could not complete this long journey without their never-ending love, support and prayer. I cannot find any word to express my deep appreciation to them. I just want to tell them how much I love them. I love you, Dad and Mom, and thanks for always being there for me.

Finally, I thank my best friend and beloved wife Dang Nguyen Thanh Yen, who always takes care of many things around me since I came here so that I could concentrate on this thesis. Her love has been a strong energy for my entire study.

This research has been supported in part by the Canada Research Chair Program, AUTO21 NCE, the NSERC Discovery grant, and OGS.

Thanh Minh Nguyen

The University of Windsor

August 2011
# Table of Contents

Author’s Declaration of Originality .................................................. iii

Abstract ............................................................................................. iv

Dedication ......................................................................................... vi

Acknowledgements ........................................................................... vii

List of Figures .................................................................................. xiii

List of Tables .................................................................................... xx

Symbols and Abbreviations .............................................................. xxi

1 Introduction .................................................................................. 1
  1.1 General Introduction ............................................................. 1
  1.2 Thesis Overview ..................................................................... 7

2 Standard Finite Mixture Model for Image Segmentation .......... 10
  2.1 Probability Distributions ..................................................... 10
2.1.1 The Gaussian Distribution .................................. 10
2.1.2 The Student’s-t distribution .................................. 12
2.2 Maximum likelihood for the Gaussian ......................... 14
2.3 Standard Finite Mixture Model ................................. 16
  2.3.1 Gaussian Mixture Model .................................. 16
  2.3.2 Student’s-t Mixture Model ............................... 19
2.4 The Expectation Maximization (EM) Algorithm .............. 22
  2.4.1 EM Algorithm for the Gaussian Mixture Model ........ 22
  2.4.2 Relation between EM and K-means ....................... 25
2.5 Gradient-Based Optimization Techniques ..................... 27
2.6 Image Segmentation Evaluation ............................... 30
2.7 Conclusions .................................................. 31

3 Gaussian Mixture Model based Markov Random Field ....... 33
  3.1 Introduction ................................................. 33
3.2 Gaussian Mixture Model based MRF for the Pixel Labels ... 34
    3.2.1 Markov Random Field Theory ........................ 34
    3.2.2 Hidden Markov models ............................... 36
3.3 Gaussian Mixture Model based MRF for the Priors of the Pixel Labels ............................................ 39
3.4 Conclusions .................................................. 44

4 An Extension of the Standard Mixture Model for Image Seg-
mentation .......................................................... 45
  4.1 Introduction ............................................... 45
4.2 Proposed Method ............................................. 46
4.3 Parameter Learning ....................................... 50
4.4 Experiments .................................................. 52
  4.4.1 Synthetic Images ........................................ 54
  4.4.2 Natural Images .......................................... 58
4.5 Conclusions .................................................. 65

5 Fast and Robust Spatially Constrained Gaussian Mixture Model for Image Segmentation 66
  5.1 Introduction .................................................. 66
  5.2 Proposed Method ........................................... 70
  5.3 Experiments .................................................. 74
    5.3.1 Segmentation of Synthetic Images ..................... 75
    5.3.2 Segmentation of Grayscale Natural Images ............ 81
    5.3.3 Segmentation of Colored Images ........................ 83
  5.4 Conclusions .................................................. 87

6 Conclusions .................................................... 89
  6.1 Summary of Main Contributions ............................. 89
  6.2 Future Directions ........................................... 91

References ......................................................... 94

Appendix A .......................................................... 110

Publications ....................................................... 112
List of Figures

2.1 Plot of the Gaussian distribution showing the mean $\mu$ and standard deviation $\sigma$. .......................... 11

2.2 Plot of Student’s-t distribution for $\mu = 0$ and $\Lambda = 1$ for various values of $v$. The limit $v \to \infty$ corresponds to a Gaussian distribution with mean $\mu$ and precision $\Lambda$. .................. 13

2.3 The graphical representation of a Gaussian mixture model for a set of $N$ pixel $x_i$. .......................... 17

2.4 Synthetic image, (a): original image, (128x128 image) (b): Corrupted original image with Gaussian noise (0 mean, 0.02 variance), (c): standard GMM. .......................... 18

2.5 Real world image (321x481 image resolution), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.02 variance), (c): standard GMM .......................... 18

2.6 Synthetic image, (a): original image, (128x128 image) (b): Corrupted original image with Gaussian noise (0 mean, 0.005 variance), (c): standard GMM, (c): standard SMM. ................. 21
2.7 EM algorithm for the mixture of gaussians, (a): The original 2D point set with the initial condition, (b): Result of GMM with this initial condition.

3.1 Synthetic image, (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.05 variance), (c): standard GMM, (d): SIMF [63].

3.2 Synthetic image, (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.01 variance), (c): standard GMM (time = 5.5s), (d): SVFMM [72] (time = 246.1s).

4.1 The first experiment (128x128 image resolution), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.05 variance), (c): K-means (MCR = 17.315%), (d): standard GMM (MCR = 33.982%), (e): SVFMM (MCR = 13.671%), (f): NEM (MCR = 17.034%), (g) ICM (MCR = 9.605%), (h): MODEF (MCR = 7.781%), (i): SIMF (MCR = 7.725%), (k): MEANF (MCR = 7.721%), (l) HMRF-FCM (MCR = 0.823%), (m): The proposed method (MCR = 0.653%).
4.2 The second experiment (128x128 image resolution), (a): original image, (b): Corrupted original image with mixed noise (salt and pepper noise (0.03%) + Gaussian noise (0 mean, 0.01 variance)), (c): K-means (MCR = 24.609%), (d): standard GMM (MCR = 40.209%), (e): SVFMM (MCR = 22.338%), (f): ICM (MCR = 22.001%), (g): MODEF (MCR = 12.744%), (h): SIMF (MCR = 13.307%), (i): MEANF (MCR = 12.200%), (k) FGFCM (MCR = 5.562%), (l) HMRF-FCM (MCR = 3.692%), (m): The proposed method (MCR = 2.789%).

4.3 Affect of the simple competitive selection, (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.03 variance), (c): MODEF (MCR = 3.512%), (d): MODEF with the simple competitive selection (MCR = 3.117%), (e): proposed method without the simple competitive selection (MCR = 0.199%), (f): proposed method with the simple competitive selection (MCR = 0.197%).

4.4 Images from the Berkeleys grayscale image segmentation dataset, (a): 135069, (b): 124084, (c): 58060, (d): 353013 with Gaussian noise (0 mean, 0.001 variance), (e): 239007, (f): 46076, (g): 15088 with Gaussian noise (0 mean, 0.005 variance), (h): 374067 with Gaussian noise (0 mean, 0.01 variance), (i): 302003 with Gaussian noise (0 mean, 0.01 variance).
4.5 Image segmentation results obtained by employing the proposed method, (a): 135069, (b): 124084, (c): 58060, (d): 353013 with Gaussian noise (0 mean, 0.001 variance), (e): 239007, (f): 46076, (g): 15088 with Gaussian noise (0 mean, 0.005 variance), (h): 374067 with Gaussian noise (0 mean, 0.01 variance), (i): 302003 with Gaussian noise (0 mean, 0.01 variance).

4.6 Grayscale image segmentation (55067), (a): original image, (b): K-means (PR = 0.879), (c): standard GMM (PR = 0.843), (d): SVFMM (PR = 0.882), (e): ICM (PR = 0.880), (f): MODEF (PR = 0.882), (g): MEANF (PR = 0.881), (h): FGFCM (PR = 0.879), (i): HMRF-FCM (PR = 0.887), (k): The proposed method (PR = 0.891).

4.7 Noisy grayscale image segmentation (24063), (a): original image, (b): corrupted original image with Gaussian noise (0 mean, 0.005 variance), (c): K-means (PR = 0.778), (d): standard GMM (PR = 0.765), (e): SVFMM (PR = 0.787), (f): MODEF (PR = 0.814), (g): MEANF (PR = 0.818), (h): FGFCM (PR = 0.796), (i): HMRF-FCM (PR = 0.815), (k): The proposed method (PR = 0.826).

4.8 Computational cost (in seconds) comparison, (a): original image, (b): K-means (0.7 sec), (c): standard GMM (36 sec), (d): ICM (169 sec), (e): MODEF (390 sec), (f): SIMF (432 sec), (g): MEANF (445 sec), (h): The proposed method (61 sec).
4.9 Minimization Progress of the negative log-likelihood function of the proposed algorithm, for the final experiment.

5.1 The first experiment (128x128 image resolution), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.03 variance), (c): K-means, (d): Standard GMM (MCR = 41.67%), (e): SVFMM (MCR = 23.28%), (f): CA–SVFMM (MCR = 20.29%), (g): ICM (MCR = 20.23%), (h): SIMF (MCR = 3.83%), (i) MEANF (MCR = 3.55%), (j): Proposed method (MCR = 1.13%).

5.2 Maximization progress of the log-likelihood of the proposed method of the first experiment.

5.3 The second experiment (128x128 image resolution), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.05 variance), (c): K-means, (d): Standard GMM (MCR = 35.02%), (e): SVFMM (MCR = 12.01%), (f): CA–SVFMM (MCR = 11.16%), (g): ICM (MCR = 7.65%), (h): SIMF (MCR = 5.65%), (i) MEANF (MCR = 5.94%), (j): Proposed method (MCR = 1.05%).
5.4 The third experiment (128x128 image resolution), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.03 variance), (c): K-means, (d): Standard GMM (MCR = 30.31%), (e): SVFMM (MCR = 18.11%), (f): CA-SVFMM (MCR = 17.73%), (g): ICM (MCR = 5.90%), (h): SIMF (MCR = 2.86%), (i) MEANF (MCR = 2.70%), (j): Proposed method (MCR = 0.21%).

5.5 The fourth experiment (256x256 image resolution), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.1 variance), (c): Proposed method in Chapter 4 (MCR = 0.10%, time = 8.9s), (j): Proposed method in Chapter 4 (MCR = 0.22%, time = 3.7s).


5.8 Grayscale natural image segmentation (374067), (a): original image, (b): Standard GMM, (c): SVFMM, (d): SIMF, (e): Proposed method. .................................................. 84


5.10 Color image segmentation (388016), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.0015 variance), (c): Standard GMM, (d): SVFMM, (e): CA–SVFMM, (f): ICM, (g) MEANF, (h): Proposed method. ...................... 86

# List of Tables

4.1 Comparison of the proposed method with other methods in term of MCR (%), for the first experiment. .................. 55
4.2 Comparison of the proposed method with other methods in term of MCR (%), for the second experiment. .................. 57
4.3 Comparison of image segmentation results on Berkeley’s grayscale image segmentation dataset: Probabilistic Rand (PR) Index. . 61
5.1 Computational cost (in seconds) comparison for the synthetic image in the first experiment. .................. 76
5.2 Computational cost (in seconds) comparison for the synthetic image in the second experiment. .................. 79
5.3 Comparison of the proposed method with other methods in term of MCR (%), for the third experiment, in the presence of varying levels of noise. .................. 79
5.4 Comparison of image segmentation results on Berkeley’s color image segmentation dataset: Probabilistic Rand (PR) Index. . 87
Symbols and Abbreviations

\( x_i \) : The \( i \)-th pixel of an image

\( N \) : The number of pixels in an image

\( \Omega_j \) : The \( j \)-th label in an image

\( K \) : The number of labels in an image

\( \mathcal{N}_i \) : The neighborhood pixels of the \( i \)-th pixel

\( \Phi(x_i | \Theta_j) \) : The Gaussian function

\( \mu_j \) : The mean of the Gaussian function \( \Phi(x_i | \Theta_j) \)

\( \sigma_j \) : The standard deviation of the Gaussian function

\( \Sigma_j \) : The covariance of the Gaussian function \( \Phi(x_i | \Theta_j) \), for the case of a \( D \)-dimensional vector \( x_i \)

\( \pi_j \) : The prior probability of all pixel belonging to the label \( \Omega_j \)

\( \pi_{ij} \) : The prior probability of the pixel \( x_i \) belonging to the label \( \Omega_j \)

\( z_{ij} \) : The posterior probability of the pixel \( x_i \) belonging to the label \( \Omega_j \)
\( f(x_i|\Pi, \Theta) \) : The density function at an observation \( x_i \)

\( p(X|\Pi, \Theta) \) : The joint conditional density

\( L(\Theta, \Pi|X) \) : The log-likelihood function

\( J(\Theta, \Pi|X) \) : The negative log-likelihood function

\( H(\Pi, \Theta|X) \) : The objective function

\( E(\Theta, \Pi) \) : The error function

GMM : Gaussian mixture model

SMM : Student’s-t mixture model

MRF : Markov random field

EM : Expectation maximization

SVFMM : Spatially variant finite mixture model

CA-SVFMM : Class adaptive spatially variant finite mixture model

NEM : Neighborhood expectation maximization

ICM : Iterated conditional model

MODEF : Mode field algorithm

SIMF : Simulated field algorithm

MEANF : Mean field algorithm

FGFCM : Fast generalized fuzzy c-means

HMRF-FCM : Hidden Markov random field based fuzzy c-means

MCR : The misclassification ratio

PR : The probabilistic rand index

\( \text{Dice}_j \) : Dice similarity coefficient for the label \( \Omega_j \)
Chapter 1

Introduction

1.1 General Introduction

In order to analyse the content of an image, it is often useful to construct a simpler representation of multiple segments. And the process to partition an image into non-overlapping regions that humans can easily separate is called image segmentation. In an image, various features can be used for segmentation process. These might be colour information that is used to create histograms, or information about the pixels that indicate boundaries or texture information.

Segmentation is an important step of low level vision. An accurately segmented image provides detailed information about the objects present in an image and their respective boundaries. There are many applications of segmentation. For example, in a vision guided tool tracker system [1], [2], [3], the robot needs to track the appropriate components in automotive manufacturing
environments, thereby increasing the productivity and profitability of automotive manufacturing enterprises and the global competitiveness. In the field of medical imaging [4], [5], [6], [7], segmentation plays an important role. Accurate medical image segmentation provides additional information that helps to prepare treatment scheme and to evaluate therapeutic effect. The applications of segmentation vary from the detection of synthetic aperture radar images [8], [9], video analysis [10], to magnetic resonance imaging (MRI) [11], and object detection [12], [13], [14] etc. In all these areas, the quality of the segmented output affects on the quality of the final output largely. However, automated segmentation [15] is still a very challenging research topic, due to overlapping intensities and low contrast in images, as well as noise perturbation.

Many previous works have been proposed for image segmentation, in particular by the method of threshold [16], [17], [18]. However, thresholding is significantly susceptible to low resolution, low contrast and signal to noise ratio. As for some part of the image, high intensity variation may correspond to edges of interest, while the other part may require high low variation. The selection of the threshold is very crucial. A bad choice of threshold [19] leads to a poor quality of the segmentation. Adaptive thresholding [20], [21], [22] often is taken as a solution to this. However, it cannot eliminate the problem of threshold selection [23].

In order to avoid the above-mentioned disadvantages, an artificial neural network [24], [25], [26] is applied for image segmentation. In [27], the authors clustered feature vectors extracted from an image using a neural network which
minimized the distance between the feature vectors. Although this approach worked well in the examples shown, it led to sub-optimal image segmentation. This is because the pixels in general are spatially correlated and the approach presented in this method did not incorporate any spatial information.

Many algorithms have been developed for image segmentation including graph-based methods [28], [29], mean shift based methods [30], [31], histogram-based methods [32], multi-scale segmentation [33], and clustering methods [34], [35], [36]. In clustering methods, K-means [37], [86] and fuzzy c-means [38] are two well-known methods that have been widely used for segmenting an image due to their simplicity and ease of implementation. However, one of their main drawbacks [39]–[42] is that these two methods ignore the spatial constraints in an image.

During the last decades, much attention has been given to model-based techniques [43], [44], [45], [100]–[105] to model the uncertainty in a probabilistic manner. In model-based techniques, standard Gaussian mixture model (GMM) [46]–[49] is a well-known method. It is a flexible and powerful statistical modeling tool for multivariate data. Many researchers have used it to study a number of key problems in the area of image segmentation [50], [51]. In standard GMM, each pixel $x_i$ is considered to be a random variable whose possibility density function $\Phi(x_i; \Theta_j)$ is a Gaussian function. The model assumes a common prior distribution $\pi_j$, which independently generates the pixel labels. In order to estimate the model parameters, expectation maximization (EM) algorithm [52]–[57] is employed to maximize the log-likelihood of the
given data set. The main advantage of the standard GMM is that it is easy to implement and requires a small number of parameters. The log-likelihood function that is used to estimate the parameters is inherently simple. However, one of the main drawbacks of this model is that the prior distribution $\pi_j$ has no dependence on the pixel index $i$. One of the other problems is that the spatial relationships between the neighboring pixels are not taken into its account [58]. Although the standard GMM is a well known and simple method for image segmentation, its segmentation result is thus sensitive to noise, varying illumination and other environmental factors such as wind, rain or camera shaking.

In order to reduce the segmentation sensitivity to noise, mixture models with Markov random field (MRF) have been employed for pixel labels [59], [62], [63], [64], [66]. The most important distinction is that in standard GMM, a common prior distribution $\pi_j$ for all pixels $x_i$ is evaluated, whereas, in these approaches, the prior distribution $\pi_{ij}$ varies for every pixel $x_i$ corresponding to each label $\Omega_j$ and depends on the neighboring pixels and their corresponding parameters [67]. This prior distribution $\pi_{ij}$ is a probability. Although these approaches can lead to an improved segmentation quality, they lack enough robustness with respect to noise. In addition, the computational cost of the MRF based methods remains quite high.

To incorporate the spatial relationships in a given image, several researchers have suggested the GMM model based on MRF [58], [68], [69], [70], [71], [72], [73], where an MRF models the joint distribution of the priors of each
pixel label, instead of the joint distribution of the pixel labels as in [59], [63], [64], [66]. These models work well for noisy image segmentation; however, in order to accurately evaluate the influence of the neighboring pixel labels during the learning step, the algorithm becomes complex and computationally expensive. In order to maximize the log-likelihood with respect to the parameters in [58], [72], [73], the M-step of EM algorithm [58], [72] cannot be applied directly to the prior distribution \( \pi_{ij} \). Therefore, various approximations have been introduced in order to tackle this problem. For example, the MAP algorithm in [58] cannot evaluate the prior distribution \( \pi_{ij} \) in a closed form, and thus the gradient projection algorithm was proposed to implement the M-step. In [72], [73], another method based on a closed form update equation was used to implement the M-step, and estimate the parameters. As compared to standard GMM based methods, the computational cost of these methods remains high. However, in addition to increased complexity, the final segmented image lacks adequate robustness to noise.

Based on these considerations, in the first part of this thesis, we propose an extension of the standard GMM [82], [83] for image segmentation, which utilizes a novel approach to incorporate the spatial relationships between neighboring pixels into the standard GMM. The proposed model is similar to the standard GMM and thus easy to implement, with the main difference that the prior distribution of each label \( \Omega_j \) is different for each pixel \( x_i \) and depends on its neighboring pixels. A new way to properly account for the relationship of neighboring pixels is introduced. In addition, the proposed method does not
require as many parameters as compared to the models based on MRF. To es-

timate the unknown parameters of the pixel’s prior distributions, as well as

the parameters of the distribution itself, instead of using the EM algorithm,

we use the gradient method to minimize a higher bound on the data nega-
tive log-likelihood. The proposed method has been applied for segmenting

synthetic and real world grayscale images. The performance of the proposed

model is compared with other methods based on standard GMM and MRF

models, there by demonstrating its robustness, accuracy and effectiveness.

In the final part of this thesis, a new mixture model for image segmen-
tation [84] is presented, which differs from the above methods in the follow-
ing manner. Firstly, our proposed method incorporates spatial relationships

amongst neighboring pixels in a simpler metric based on MRF. Therefore, the

proposed method is fast and easy to implement, compared with other mixture

models that are complex and computationally expensive. Generally, in above-

mentioned models based MRF, the M-step of the EM algorithm cannot be
directly applied for the maximization of the log-likelihood with respect to the

parameters. In our proposed method, we can directly apply the EM algorithm
to optimize the parameters, which makes it simpler. Finally, the proposed

model is quite robust with respect to noise, more accurate and effective as

compared to other GMM based methods.
1.2 Thesis Overview

In the Chapter 2, the first group of model-based techniques is described beginning with the using of standard GMM to solve the fully unsupervised segmentation problem. The advantages and disadvantages of the standard GMM are then discussed. Next, in order to estimate the parameters of the model, various techniques based on maximizing their likelihood are described, beginning with the EM algorithm, then continuing with the gradient-based optimization techniques. The criteria evaluation for unsupervised segmentation algorithms is addressed before concluding with some observations regarding the relevance of the reviewed literature to the direction of research presented in our thesis.

In Chapter 3, we describe the second group of model-based techniques for unsupervised segmentation. In order to take into account the spatial correlation between the neighboring pixels and reduce the sensitivity of the segmentation result with respect to noise, the mixture models presented here is based on the MRF for modeling and processing image data. In the second group of model-based techniques, the mixture models based on MRF can be divided into two types. In the first type, mixture models with MRF have been employed for pixel labels. In order to take into account the spatial correlation, an MRF model in the second type is used to model the joint distribution of the priors of each pixel label, instead of the joint distribution of the pixel labels as in first type. For each type of mixture models in this chapter, the limitations and advantages are also described. Moreover, the main differences between the first and the second group of model-based techniques are discussed. The
most important distinction is that in standard GMM of the first group, a common prior distribution for all pixels is evaluated, whereas, in the approaches of the second group, the prior distribution varies for every pixel corresponding to each label and depends on the neighboring pixels and their corresponding parameters.

Chapter 4 presents new unsupervised segmentation algorithms [82], [83]. In this chapter, we propose an extension of the standard GMM for image segmentation, which utilizes a novel approach to incorporate the spatial relationships between neighboring pixels into the standard GMM. The proposed model is easy to implement and compared with MRF models, requires fewer parameters. We also propose a new method to estimate the model parameters in order to minimize the higher bound on the data negative log-likelihood, based on the gradient method. Results are presented before conclusions are drawn.

Chapter 5 describes a new way to incorporate spatial information between the neighboring pixels into the GMM based on MRF [84]. In comparison to other mixture models that are complex and computationally expensive, the proposed method is fast and easy to implement. In mixture models based on MRF, the M-step of the EM algorithm cannot be directly applied to the prior distribution for maximization of the log-likelihood with respect to the corresponding parameters. Compared with these models, our proposed method directly applies the EM algorithm to optimize the parameters, which makes it much simpler. Finally, experimental results obtained by employing the pro-
posed method on many synthetic and real-world grayscale and colored images demonstrate its robustness, accuracy and effectiveness, as compared with other mixture models.

Chapter 6 concludes the thesis with suggestions for further work.
Chapter 2

Standard Finite Mixture Model for Image Segmentation

2.1 Probability Distributions

2.1.1 The Gaussian Distribution

In probability theory, one of the most important distributions for continuous variables is Gaussian distribution. It is historically called the law of errors and is considered the most popular probability distribution in practice, and is used throughout statistics. For the case of a single real-valued variable $x$, the Gaussian distribution has its own mean $\mu$ and standard deviation $\sigma$ and is defined by:

$$
\Phi(x|\Theta) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( - \frac{(x - \mu)^2}{2\sigma^2} \right)
$$

(2.1)
Where $\Theta = \{\mu, \sigma\}$. The graph of the Gaussian distribution [80] is shown in Figure 2.1.

![Gaussian distribution graph]

Figure 2.1: Plot of the Gaussian distribution showing the mean $\mu$ and standard deviation $\sigma$.

As shown in Eq.(2.1), we see that the Gaussian distribution satisfies the two requirements for a valid probability density.

$$\Phi(x|\Theta) > 0 \quad (2.2)$$

And,

$$\int_{-\infty}^{\infty} \Phi(x|\Theta) dx = 1 \quad (2.3)$$

Within the Gaussian distribution defined by Eq.(2.1), the average value of $x$ is

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \Phi(x|\Theta) x dx = \mu \quad (2.4)$$

And the variance of $x$ is given by

$$\text{var}[x^2] = \mathbb{E}[x^2] - (\mathbb{E}[x])^2 = \int_{-\infty}^{\infty} \Phi(x|\Theta) x^2 dx - \mu^2 = \sigma^2 \quad (2.5)$$
For the case of a $D$-dimensional vector $x$, each Gaussian distribution $\Phi(x|\Theta)$ can be written in the form:

$$\Phi(x|\Theta) = \frac{1}{(2\pi)^{D/2} \det|\Sigma|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right\}$$  \hspace{1cm} (2.6)

where $\Theta = (\mu, \Sigma)$. The $D$-dimensional vector $\mu$ is the mean, the $D\times D$ matrix $\Sigma$ is the covariance, and $|\Sigma|$ denotes the determinant of $\Sigma$.

### 2.1.2 The Student’s-t distribution

Student’s-t distribution is a continuous probability distribution that is heavily tailed than Gaussian. Hence, it is more prone to producing values that fall far from its mean. Student’s-t distribution plays an important role in a number of widely used statistical analysis and is used to estimate the mean of a normally distributed population in situations where the sample size is small. It has proven to be quite effective for image segmentation [106].

Student’s-t distribution is symmetric and bell-shaped, like the normal distribution, meaning that it has its own parameters $\Theta = \{ \mu, \Lambda, v \}$ with mean $\mu$, precision (inverse covariance) $\Lambda$ and degree of freedom $v$. For the case of a $D$-dimensional vector $x$, it has the probability density function.

$$S(x|\Theta) = \frac{\Gamma(v/2 + D/2)|\Lambda|^{1/2}}{\Gamma(v/2)(v\pi)^{D/2}} \left( 1 + \frac{\Delta^2}{v} \right)^{-(v+D)/2}$$  \hspace{1cm} (2.7)

where, $\Gamma(\cdot)$ is the Gamma function [80]:

$$\Gamma(y) = \int_0^\infty t^{y-1}e^{-t} dt$$  \hspace{1cm} (2.8)
and, $\Delta^2$ is the squared Mahalanobis distance from pixel $x$ to mean $\mu$.

$$\Delta^2 = (x - \mu)^T \Lambda (x - \mu)$$

(2.9)

The degree of freedom $v$ is illustrated in Figure 2.2. For the particular case of $v = 1$, the Student’s-t distribution reduces to the Cauchy distribution [80], while in the limit $v \to \infty$ the Student’s-t distribution becomes a Gaussian with mean $\mu$ and precision $\Lambda$. Eq.(2.7) is the multivariate form of Student’s-t distribution and satisfies the following properties

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} S(x | \Theta) x dx = \mu$$

(2.10)

And the variance of $x$ is given by

$$\text{var}[x^2] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \int_{-\infty}^{\infty} S(x | \Theta) x^2 dx - \mu^2 = \frac{v}{v - 2} \Lambda^{-1}$$

(2.11)
2.2 Maximum likelihood for the Gaussian

Let \( x_i \) denote an observation. Given a data set \( X = (x_1, x_2, \ldots, x_N) \) in which the observations \( x_i \) are assumed to be drawn independently from a Gaussian distribution, we can estimate the parameters of the Gaussian distribution by maximum likelihood [80]. The log-likelihood function is given by

\[
L(\Theta|X) = \frac{ND}{2} \log(2\pi) - \frac{N}{2} \log |\Sigma| - \frac{1}{2} \sum_{i=1}^{N} \frac{(x_i - \mu)^2}{\sigma^2}
\]  

(2.12)

where \( \Theta = \{\mu, \sigma\} \). After some manipulation, we see that the likelihood function depends on the data set only through the two quantities

\[
\sum_{i=1}^{N} x_i \quad \text{and} \quad \sum_{i=1}^{N} x_i^2
\]  

(2.13)

The next objective is to optimize the parameter set \( \Theta = \{\mu, \sigma\} \) in order to maximize the log-likelihood function in Eq.(2.12). Let us now consider the derivation of the function \( L(\Theta|X) \) with the means \( \mu \), we have:

\[
\frac{\partial L(\Theta|X)}{\partial \mu} = \sum_{i=1}^{N} \frac{x_i - \mu}{\sigma^2}
\]  

(2.14)

and setting this derivative to zero, we obtain the solution for the maximum likelihood estimate of the mean at the \((t+1)\) step. The mean of the observed set of data points is given by

\[
\mu^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]  

(2.15)

The maximization of log-likelihood function \( L(\Theta|X) \) in Eq.(2.12) with respect to \( \sigma \) is rather more involved. Setting the derivative of the function in Eq.(2.12)
with $\sigma$ at the $(t+1)$ iteration step, we have:

\[
\frac{\partial L(\Theta|X)}{\partial \sigma} = \sum_{i=1}^{N} \left( -\frac{1}{\sigma} + \frac{(x_i - \mu)^2}{\sigma^3} \right) \tag{2.16}
\]

The solution of $\frac{\partial L(\Theta|X)}{\partial \sigma} = 0$ yields the minimizer of $\sigma$ at the $(t+1)$ step.

The result is as expected and takes the form

\[
[\sigma^2]^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu^{(t+1)})^2 \tag{2.17}
\]

Note that the solution for $\mu^{(t+1)}$ in Eq.(2.15) does not depend on $\sigma^{(t+1)}$, and so we can first evaluate $\mu^{(t+1)}$ and then use this to evaluate $\sigma^{(t+1)}$. If we evaluate the expectations of the maximum likelihood solutions under the true distribution in Eq.(2.15) and Eq.(2.17), we obtain the following results

\[
\mathbb{E} \left[ \mu^{(t+1)} \right] = \mu \tag{2.18}
\]

and,

\[
\mathbb{E} \left[ [\sigma^2]^{(t+1)} \right] = \frac{N - 1}{N} \sigma^2 \tag{2.19}
\]

As shown in Eq.(2.18), the expectation of the maximum likelihood estimate for the mean is equal to the true mean. However, the maximum likelihood estimate for the covariance has an expectation that is less than the true value.

We can correct this bias by defining a different estimator $[\tilde{\sigma}^2]^{(t+1)}$ given by

\[
[\tilde{\sigma}^2]^{(t+1)} = \frac{1}{N - 1} \sum_{i=1}^{N} (x_i - \mu^{(t+1)})^2 \tag{2.20}
\]

From Eq.(2.19) and (2.20), the expectation of $[\tilde{\sigma}^2]^{(t+1)}$ is equal to $\sigma^2$. 

15
2.3 Standard Finite Mixture Model

2.3.1 Gaussian Mixture Model

Over the last few years, much attention has been given to the standard GMM. An advantage of the standard GMM is that it requires a small amount of parameters for learning. Another advantage is that these parameters can be efficiently estimated by adopting the EM algorithm to maximize the log-likelihood function.

Let $x_i; i=1,2,...,N$; denote the observation at the $i$-th pixel of an image. Labels are denoted by $\Omega_1, \Omega_2, ..., \Omega_K$. Consider the problem of estimating the posterior probability of $x_i$ belonging to label $\Omega_j$. If we assume that $x_i$ is drawn independently from the distribution, then the standard GMM [46], [75] assumes that the density function at an observation $x_i$ is given by:

$$f(x_i|\Pi, \Theta) = \sum_{j=1}^{K} \pi_j \Phi(x_i|\Theta_j) \tag{2.21}$$

The graphical representation of a Gaussian mixture model for a set of $N$ pixel $x_i$ is shown in Figure 2.3. Where $\Pi = \{\pi_1, \pi_2, ..., \pi_K\}$, and $\pi_j$ is the prior distribution of the pixel $x_i$ belonging to the label $\Omega_j$, which satisfies the constraints:

$$0 \leq \pi_j \leq 1 \text{ and } \sum_{j=1}^{K} \pi_j = 1 \tag{2.22}$$

Each Gaussian distribution $\Phi(x_i|\Theta_j)$ is called a component of the mixture. For the case of a single real-valued variable $x_i$, the Gaussian distribution
Figure 2.3: The graphical representation of a Gaussian mixture model for a set of \( N \) pixel \( x_i \).

has its own mean \( \mu_j \) and covariance \( \sigma_j \) and is defined by:

\[
\Phi(x_i|\Theta_j) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{(x_i - \mu_j)^2}{2\sigma^2} \right) \tag{2.23}
\]

where \( \Theta_j = \{\mu_j, \sigma_j\} \). The observation \( x_i \) in Eq.(2.21) is modeled as statistically independent. And the joint conditional density [58], [80] of the data set \( X = (x_1, x_2, ..., x_N) \) can be modeled as:

\[
p(X|\Pi, \Theta) = \prod_{i=1}^{N} f(x_i|\Pi, \Theta) = \prod_{i=1}^{N} \left[ \sum_{j=1}^{K} \pi_j \Phi(x_i|\Theta_j) \right] \tag{2.24}
\]

Given the joint conditional density from Eq.(2.24), the log-likelihood function of the standard GMM [80] is given by:

\[
L(\Theta, \Pi|X) = \sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_j \Phi(x_i|\Theta_j) \right\} \tag{2.25}
\]

Where \( \Theta = \{\Theta_j\}; \ j=1,2,\ldots,K \). As can be seen from the likelihood function in
Eq. (2.25), one of the biggest advantages of the standard GMM is that it has a simple form, and requires a small number of parameters.

However, the main drawback is that we cannot assign the same weight for every pixel belonging to the label \( \Omega_j \), as the pixels in the image vary in their intensity values and locations. Another limitation of the standard GMM is that the pixel \( x_i \) is considered to be an independent sample. Therefore, it does not take into account the spatial correlation between the neighboring pixels in the decision process. Segmentation is extremely sensitive to noise and illumination. The original image in Figure 2.4(a) is corrupted with a Gaussian noise (0 mean, 0.02 variance). The objective is to segment the image
in Figure 2.4(b) into four labels. Figure 2.4(c) show the segmentation results for standard GMM. We can see that some details are lost in the segmented image. Another image from the Berkeleys image segmentation dataset [98], as shown in Figure 2.5(a), is used in next experiment. The image shown in Figure 2.5(b) is derived from the original image by corrupting it with Gaussian noise (0 mean, 0.02 variance). The objective is to segment the noisy image into two labels. As can be seen, the segmentation accuracy of standard GMM method, along the object boundaries is quite poor.

2.3.2 Student’s-t Mixture Model

To improve the robustness of the algorithm to outliers, Student’s-t distribution has been used. The main advantage of the Student’s-t distribution is that it is heavily tailed than Gaussian, and hence finite mixture model of the longertailed multivariate Student’s-t distribution provides a much more robust approach to the standard GMM. It has proven to be quite effective for image segmentation [106]. In order to partition an image consisting of $N$ pixels into $K$ labels, standard Student’s-t mixture model (SMM) assumes that each pixel $x_i$ is independent of the label $\Omega_j$. The density function at a pixel $x_i$ is given by:

$$f(x_i|\Pi, \Theta) = \sum_{j=1}^{K} \pi_j S(x_i|\Theta_j)$$  \hspace{1cm} (2.26)

where, $\Pi = \{\pi_j\}; \ j=(1,2,...,K)$; is the set of prior distributions modeling the probability that pixel $x_i$ is in label $\Omega_j$, which satisfies the constraints in Eq.(2.22).
Each Student’s-t distribution $S(x_i|\Theta_j)$, called a component of the mixture, has its own parameters $\Theta_j = \{\mu_j, \Lambda_j, v_j\}$. The Student’s-t distribution $S(x_i|\Theta_j)$ is given by:

$$S(x_i|\Theta_j) = \frac{\Gamma(v_j/2 + D/2)|\Lambda_j|^{1/2}}{\Gamma(v_j/2)(v_j\pi)^{D/2}} \left(1 + \frac{\Delta^2_j}{v_j}\right)^{-(v_j + D)/2}$$

(2.27)

where, $\Gamma(\cdot)$ is the Gamma function. And, $\Delta^2$ is the squared Mahalanobis distance from pixel $x_i$ to mean $\mu_j$.

$$\Delta^2 = (x_i - \mu_j)^T \Lambda_j (x_i - \mu_j)$$

(2.28)

The joint conditional density of the data set $X = (x_1, x_2, \ldots, x_N)$ is modeled as:

$$p(X|\Pi, \Theta) = \prod_{i=1}^N f(x_i|\Pi, \Theta) = \prod_{i=1}^N \sum_{j=1}^K \pi_j S(x_i|\Theta_j)$$

(2.29)

Then, the log-likelihood function of the standard SMM is given by the following identity:

$$L(\Theta, \Pi|X) = \log p(X|\Pi, \Theta) = \sum_{i=1}^N \log \left\{\sum_{j=1}^K \pi_j S(x_i|\Theta_j)\right\}$$

(2.30)

Unfortunately, there is no closed form solution for maximizing the log-likelihood under a Student’s-t distribution. To overcome this problem, the Student’s-t distribution in previous SMM models [80] is represented as an infinite mixture of scaled Gaussians. In particular, we can write the Student’s-t distribution in the form.

$$S(x_i|\Theta_j) = \int_0^\infty \Phi(x_i|\mu_j, u_j\Lambda_j)G(u_j|v_j/2, v_j/2)du_j$$

(2.31)
where, $\Phi(x_i|\mu_j, u_j \Lambda_j)$ denotes the Gaussian distribution, and $\mathcal{G}(u_j|v_j/2, v_j/2)$ is the Gamma distribution. As apparent, the representation of the Student’s-t distribution as an infinite mixture of scaled Gaussians in Eq.(2.31) will correspond to an increase in complexity.

As shown from the log-likelihood function in Eq.(2.30), the pixel $x_i$ in SMM is regarded as the same as $x_i$ in GMM. Each pixel $x_i$ is considered independent of its neighbors. The spatial correlation between the neighboring pixels is not taken into account in the decision process. Moreover, the prior distribution $\pi_j$ does not depend on the pixel index and has the same value for all pixels.

![Figure 2.6](image)

Figure 2.6: Synthetic image, (a): original image, (128x128 image) (b): Corrupted original image with Gaussian noise (0 mean, 0.005 variance), (c): standard GMM, (d): standard SMM.

In Figure 2.6, we illustrate the performances of the standard GMM and SMM for image segmentation. An image of size 128x128 with four labels, as shown in Figure 2.6(a), is used in this example. Each square box in this image has a size of 64x64 pixel and has the same luminance value [0, 1/3, 2/3, 1]. The segmentation results of standard GMM, and standard SMM are shown Figure 2.6(c)–(d), respectively. In this example, compared with GMM
method, the standard SMM demonstrates a higher degree of robustness with respect to noise. However, as we see in Figure 2.6(d), the effect of noise on the result of standard SMM is still very high.

2.4 The Expectation Maximization (EM) Algorithm

2.4.1 EM Algorithm for the Gaussian Mixture Model

In order to maximize the likelihood function given in Eq.(2.25), we need to determine the parameters of the GMM. Various techniques [48], [76] have been previously developed to determine these parameters, based on maximizing their likelihood \( L(\Theta, \Pi|X) \) in Eq.(2.25), for a given data set. In [46], the well-known EM algorithm is used to approximate the maximum likelihood.

Let us begin by setting the derivatives of \( L(\Theta, \Pi|X) \) in Eq.(2.25) with respect to the means \( \mu_j \) of the Gaussian components to zero. We obtain

\[
\frac{\partial L(\Theta, \Pi|X)}{\partial \mu_j} = - \sum_{i=1}^{N} \frac{\pi_j \Phi(x_i|\Theta_j)}{\sum_{l=1}^{K} \pi_l \Phi(x_i|\Theta_l)} \frac{x_i - \mu_j}{\sigma_j^2} = 0 \quad (2.32)
\]

where we have made use of the form Eq.(2.23) for the Gaussian distribution.

Note that the posterior probabilities \( z_{ij} \) appear naturally on the right-hand side:

\[
z_{ij}^{(t)} = \frac{\pi_j^{(t)} \Phi(x_i|\Theta_j^{(t)})}{\sum_{l=1}^{K} \pi_l^{(t)} \Phi(x_i|\Theta_l^{(t)})} \quad (2.33)
\]
where $t$ indicates the iteration step. The solution of $\partial L(\Theta, \Pi|X)/\partial \mu_j = 0$ yields the minimum of $\mu_j$ at the $(t+1)$ iteration step:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} x_i}{\sum_{i=1}^{N} z_{ij}^{(t)}}$$

(2.34)

If we set the derivative of $L(\Theta, \Pi|X)$ in Eq.(2.25) with respect to $\sigma_j$, and follow a similar line of reasoning, we obtain

$$[\sigma_j^2]^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} (x_i - \mu_j^{(t+1)})^2}{\sum_{i=1}^{N} z_{ij}^{(t)}}$$

(2.35)

Finally, we maximize $L(\Theta, \Pi|X)$ with respect to the prior distribution $\pi_j$. Here we must take account of the constraint in Eq.(2.22), which requires the prior distribution $\pi_j$ to sum to one. This can be achieved by using a Lagrange multiplier $\eta$ and maximizing the following quantity:

$$\frac{\partial}{\partial \pi_j} \left[ L(\Theta, \Pi|X) - \eta \left( \sum_{j=1}^{K} \pi_j - 1 \right) \right] = 0$$

(2.36)

which gives

$$\sum_{i=1}^{N} \frac{\Phi(x_i|\Theta_j)}{\sum_{t=1}^{K} \pi_t \Phi(x_i|\Theta_t)} - \eta = 0$$

(2.37)

If we now multiply both sides by $\pi_j$ and use the constraint in Eq.(2.22), we find $\eta = N$. Using this to eliminate $\eta$ and rearranging we obtain

$$\pi_j^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} z_{ij}^{(t)}$$

(2.38)

We summarize the EM algorithm for Gaussian mixture model below:

**Step 1:** Initialize the parameters $\Xi = \{\Theta, \Pi\} = \{\mu_j, \sigma_j, \pi_j\}$: the means $\mu_j$,
covariance values $\sigma_j$ and prior distributions $\pi_j$.

**Step 2** (E-step): Evaluate the values $z_{ij}$ in Eq.(2.33) using the current parameter values.

**Step 3** (M-step): Re-estimate the parameters $\Xi = \{\Theta, \Pi\} = \{\mu_j, \sigma_j, \pi_j\}$.

- Update the means $\mu_j$ by using Eq.(2.34).
- Update covariance values $\sigma_j$ by using Eq.(2.35).
- Update prior distributions $\pi_j$ by using Eq.(2.38).

**Step 4**: Evaluate the log-likelihood $L(\Theta, \Pi|X)$ in Eq.(2.25) and check the convergence of either the log-likelihood function, or the parameter values. If the convergence criterion is not satisfied, then go to step 2.

After optimizing the parameters of the GMM and determining the posterior probability $z_{ij}$, Eq.(2.33) is used to assign labels to each pixel in the image. The performance of EM algorithm for Gaussian mixture model is shown in Figure 2.7. In this example, four hundred simulated points are generated from
four multivariate Gaussian distributions. Each component has one hundred data points. In Figure 2.7(a), we show the initial condition for EM algorithm. As shown in Figure 2.7(b), GMM is likely to successfully classify the data points.

2.4.2 Relation between EM and K-means

According to the K-means [86], each pixel $x_i$ in an image belongs to just one label. It is based on the minimization of the following objective function:

$$H = \sum_{i=1}^{N} \sum_{j=1}^{K} r_{ij} ||x_i - \mu_j||^2$$  \hspace{1cm} (2.39)

The objective function in Eq.(2.39) represents the sum of the squares of the distances of each pixel to its assigned vector $\mu_j$. The binary indicator variable $r_{ij}$ is expressed as

$$r_{ij} = \begin{cases} 
1 & \text{if } j = \arg\min_k ||x_i - \mu_k||^2 \\
0 & \text{otherwise}
\end{cases}  \hspace{1cm} (2.40)$$

The binary indicator variable $r_{ij}$ in Eq.(2.40) describes which of the $K$ labels the pixel $x_i$ is assigned to. So that if data point $x_i$ is assigned to label $\Omega_j$ then $r_{ij}=1$, and $r_{ij}=0$ for $j \neq k$. In Eq.(2.39), the term $||x_i - \mu_j||^2$ expresses the similarity between the data and the mean. The optimum is reached when the mean $\mu_j$ of the lable $\Omega_j$ is found such that the objective function $H$ in Eq.(2.39) is minimized.

Now consider the optimization of the $\mu_j$ with the $r_{ij}$ held fixed. The objective function $H$ is a quadratic function of $\mu_j$. The gradient of the function
\( H \) with respect to parameters \( \mu_j \) is given by:

\[
\frac{\partial H}{\partial \mu_j} = -2 \sum_{i=1}^{N} r_{ij}(x_i - \mu_j)
\] (2.41)

The solution of \( \partial H / \partial \mu_j = 0 \) yields the minimum of \( \mu_j \) at the \((t+1)\) iteration step:

\[
\mu_j = \frac{\sum_{i=1}^{N} r_{ij}x_i}{\sum_{i=1}^{N} r_{ij}}
\] (2.42)

As an illustration in Eq.(2.42), we can see that one of the main advantages of K-means method is that it is very simple and easy to implement. Many researchers have used it in studying a number of key problems in image segmentation. However, from the objective function in Eq.(2.39), the pixel \( x_i \) in K-means is considered an independent sample, and thus this method does not take into account the spatial correlation between the neighboring pixels in the decision process. For that reason, the segmentation result of this method is very sensitive to noise.

Comparing the mathematical expressions of K-means with the EM algorithm for Gaussian mixtures, we see that there is a close similarity [79], [80]. As shown in Eq.(2.34), the EM algorithm makes a soft assignment based on the posterior probabilities \( z_{ij} \). Whereas the K-means algorithm performs a hard assignment of data points to clusters based on the binary indicator variables \( r_{ij} \), in which each data point is associated uniquely with just one label as shown in Eq.(2.42). In fact, we see that the K-means algorithm can be derived as a particular limit of EM for Gaussian mixture as follows.
Consider a Gaussian mixture in which the covariance matrices of the mixture components are given by \( \varepsilon I \), where \( I \) is the identity matrix

\[
\Phi(x_i|\Theta_j) = \frac{1}{\sqrt{2\pi\varepsilon}} \exp \left( \frac{-||x_i - \mu_j||^2}{2\varepsilon} \right)
\]  (2.43)

Applying the EM algorithm in subsection 2.4.1, the posterior probabilities \( z_{ij} \) for a particular data point \( x_i \), are given by

\[
z_{ij} = \frac{\pi_j \exp \{-||x_i - \mu_j||^2/2\varepsilon\}}{\sum_{l=1}^{K} \pi_l \exp \{-||x_i - \mu_l||^2/2\varepsilon\}}
\]  (2.44)

If we consider the limit \( \varepsilon \to 0 \), we see that \( z_{ij} \to r_{ij} \), where \( r_{ij} \) is defined by Eq.(2.40). And the EM estimation equation for the mean \( \mu_j \), given by Eq.(2.34), then reduces to the K-means result in Eq.(2.42). Note that the K-means algorithm only estimates the means but not the covariances of the labels. Finally, in the limit \( \varepsilon \to 0 \), the expected complete data log-likelihood [80], is given by

\[
\mathbb{E}[L(\Theta, \Pi|X)] \to \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{K} r_{ij} ||x_i - \mu_j||^2 + \text{const}
\]  (2.45)

From Eq.(2.45), we observe that K-means is a special case of the EM algorithm for Gaussian mixtures.

### 2.5 Gradient-Based Optimization Techniques

In order to estimate the parameters of a Gaussian mixture model, we will consider another technique based on the gradient method [79] in this subsection.
First of all, the prior probabilities $\pi_j$ corresponding to the label $\Omega_j$ are chosen:

$$
\pi_j = \frac{\exp(\gamma_j)}{\sum_{k=1}^{K} \exp(\gamma_k)} \tag{2.46}
$$

The transformation given by Eq.(2.46) is called the softmax function, or normalized exponential, and ensures that, for $-\infty \leq \gamma_j \leq \infty$, the constraints in Eq.(2.22) are satisfied as required for probabilities.

Given the density function in Eq.(2.21) and the prior probabilities $\pi_j$ in Eq.(2.46), we need to optimize the parameter set $\Xi = \{\Theta, \Pi\} = \{\mu_j, \sigma_j, \gamma_j\}$ in order to maximize the log-likelihood function in Eq.(2.25). Since the logarithm is a monotonically increasing function, it is more convenient to consider the negative logarithm of the likelihood function [80], [82], [83] as an error function:

$$
J(\Theta, \Pi|X) = -L(\Theta, \Pi|X) = -\sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_j \Phi(x_i|\Theta_j) \right\} \tag{2.47}
$$

Applying the complete data [79], [80], [82], [83], minimizing the negative log-likelihood function in Eq.(2.47), is equivalent to minimizing the error function $E(\Xi^{(t)}|\Xi^{(t+1)})$:

$$
E(\Xi^{(t)}|\Xi^{(t+1)}) = -\sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \log \left\{ \pi_{ij}^{(t+1)} \Phi(x_i|\Theta_j^{(t+1)}) \right\} \tag{2.48}
$$

where

$$
z_{ij}^{(t)} = \frac{\pi_{ij}^{(t)} \Phi(x_i|\Theta_j^{(t)})}{\sum_{l=1}^{K} \pi_{ij}^{(t)} \Phi(x_i|\Theta_l^{(t)})}
$$

is the posterior probability at the iteration of the current step as shown in Eq.(2.33).
To minimize this error function, we apply the gradient descent algorithm [79] to adjust the parameters \( \Xi = \{ \Theta, \Pi \} = \{ \mu_j, \sigma_j, \gamma_j \} \). The change in the parameters is then given by:

\[
\Xi_{\text{new}} = \Xi_{\text{old}} - \eta \nabla E(\Xi_{\text{old}}) \tag{2.49}
\]

where, \( \nabla E(\Xi) = (\partial E/\partial \mu_j, \partial E/\partial \sigma_j, \partial E/\partial \gamma_j) \), \( \eta \) is the learning rate and its value is sufficiently small. The gradient of this error function \( E(\Xi) \) with respect to parameters \( \mu_j \) is given by:

\[
\frac{\partial E}{\partial \mu_j} = -\sum_{i=1}^{N} z_{ij}^{(t)} x_i - \mu_j \sigma_j^2 \tag{2.50}
\]

Similarly, the derivative of \( E(\Xi) \) with respect to \( \sigma_j \) is given by

\[
\frac{\partial E}{\partial \sigma_j} = -\sum_{i=1}^{N} z_{ij}^{(t)} \left( -\frac{1}{\sigma_j} + \frac{(x_i - \mu_j)^2}{\sigma_j^3} \right) \tag{2.51}
\]

The derivative of \( E(\Xi) \) with respect to \( \gamma_j \) is expressed as:

\[
\frac{\partial E}{\partial \gamma_j} = -\sum_{i=1}^{N} \left( z_{ij}^{(t)} - \pi_j^{(t)} \right) \tag{2.52}
\]

We summarize the gradient-based optimization techniques for Gaussian mixture model below:

**Step 1:** Initialize the parameters \( \Xi = \{ \mu_j, \sigma_j, \gamma_j \} \): the means \( \mu_j \), covariance values \( \sigma_j \) and the value of \( \gamma_j \).

**Step 2:** Evaluate the values \( z_{ij} \) in Eq.(2.33) using the current parameter values.

**Step 3:** Re-estimate the parameters \( \Xi = \{ \mu_j, \sigma_j, \gamma_j \} \) by using Eq.(2.49).

**Step 4:** Evaluate the log-likelihood \( L(\Theta, \Pi | X) \) in Eq.(2.25) and check the convergence of either the log-likelihood function, or the parameter values. If the convergence criterion is not satisfied, then go to step 2.
2.6 Image Segmentation Evaluation

Once the parameter-learning phase is complete, in order to assign labels to each pixel, the posterior probability $z_{ij}$ is used. For each pixel $x_i$, given the posterior probability $z_{ij}$ for all labels, in order to segment an image consisting of $N$ pixels into $K$ labels, a determination is made, whereby the pixel $x_i$ is assigned to the label with the largest posterior probability:

$$x_i \in \Omega_j : \text{IF } z_{ij} \geq z_{ik}; \ j, k = 1, 2, ..., K$$  \hspace{1cm} (2.53)

In order to evaluate the segmentation performance quantitatively, the misclassification ratio (MCR) [91] is employed:

$$\text{MCR} = \frac{\text{number of misclassified pixels}}{\text{total number of pixels}} \times 100$$  \hspace{1cm} (2.54)

The value of MCR is in the $[0–100]$ range, where lower values indicate better segmentation results.

Another technique to obtain an objective performance evaluation is to adopt the probabilistic rand (PR) index [93]. For each image, the multiple ground truths are available and are denoted as $G = \{G_1, G_2, ..., G_M\}$. The segmentation map under evaluation is denoted as $G_{\text{eval}}$. The PR index is given by:

$$\text{PR}(G, G_{\text{eval}}) = \frac{2}{M(M-1)} \sum_i \sum_{j>i} [c_{ij}p_{ij} + (1 - c_{ij})(1 - p_{ij})]$$  \hspace{1cm} (2.55)

where $M$ is the number of image pixels. $p_{ij}$ is the ground truth probability that pixels $i$ and $j$ belong to the same segment. If pixels $i$ and $j$ belong to the same segment in $G_{\text{eval}}$, the value of $c_{ij}$ is one. Otherwise, its value is zero.
The PR index takes a value in the interval [0–1]. A score of zero indicates a bad segmentation where every pixel pair in the test image has the opposite relationship as every pair in the ground truth segmentation. Otherwise, a score of one indicates a good result where every pixel pair in the test image has the same relationship as every pair in the ground truth segmentation.

In order to quantify the overlap between the segmented image and the ground truth for the label $\Omega_j$, the Dice similarity coefficient [94] is used:

$$\text{Dice}_j = \frac{2V_{ab}^j}{(V_a^j + V_b^j)} \times 100$$  \hspace{1cm} (2.56)

Where, $V_{ab}^j$ denotes the number of pixels that are assigned to label $\Omega_j$ by both the segmented image and ground truth. The number of pixels assigned to $\Omega_j$ by the segmented image and the ground truth are denoted by $V_a^j$ and $V_b^j$, respectively. The Dice index attains the value in the [0–100] range, where higher values indicate better segmentation results.

## 2.7 Conclusions

In this chapter, various criteria evaluations for unsupervised segmentation algorithms are addressed. Besides that, the standard GMM for image segmentation is presented. This mixture model is a well-known method that has been widely used as a tool for image segmentation. Its success is attributed to the fact that the model parameters can be efficiently estimated by adopting various techniques such as EM algorithm, or gradient-based optimization techniques. Other advantages are its simplicity and ease of implementation.
However, the major disadvantage of GMM is that the model assumes that each pixel is independent of its neighbors. It is well known that pixels in an image are similar in some sense and cannot be classified consistently based on feature attributes alone. Thus, the segmentation result of GMM is extremely sensitive to noise.
Chapter 3

Gaussian Mixture Model based
Markov Random Field

3.1 Introduction

As mentioned in Chapter 2, a major shortcoming of standard GMM is that it does not take into account the spatial dependencies in the image. Moreover, it does not use the prior knowledge that adjacent pixels most likely belong to the same cluster. In this family of model-based techniques, prior probabilities of label membership are considered constant for every pixel of an image. Thus, the performance of these methods is too sensitive to noise and image contrast levels.

A possible approach to overcome this problem is to impose spatial smoothness constraints to incorporate the spatial relationships between neighboring pixels [59], [60], [61]. Several mixture models based on MRF for pixel
labels are proposed in [59]–[64], [89], [90]. According to these approaches, prior probabilities are based on MRF to capture spatial information. The primary advantage of this family of mixture models is that it incorporates spatial information. Hence, it improves segmentation results, particularly when an image is corrupted by high levels of noise.

Another family of mixture models based on MRF for pixel label priors have been successfully applied to image segmentation [58], [72]–[74], [100]. Instead of imposing the smoothness constraint on the pixel label as in the above category, however, these methods aim to impose the smoothness constraint on the contextual mixing proportions. Their primary disadvantage, however, lies in its additional training complexity. The M-step of the EM algorithm in [72]–[74] cannot evaluate the prior distribution in a closed form, which therefore corresponds to an increase in the algorithm’s complexity. In [58] the gradient projection step was proposed to implement the M-step. Another reparatory projection step based on a closed form update equation was introduced [72] to guarantee that the prior probabilities are positive and sum to one.

### 3.2 Gaussian Mixture Model based MRF for the Pixel Labels

#### 3.2.1 Markov Random Field Theory

Markov random field (MRF) is a probability theory that provides a stochastic mathematical framework for analyzing the spatial constraint in an image. Let
$S$, $S = (1, 2, ..., N)$, index a discrete set of $N$ sites (pixels). And $L$, $L = (1, 2, ..., K)$ is a set of label. For every site $i \in S$, we consider a finite space $Z_i$ of states $z_i$, such as $Z_i = (z_i, z_i \in L)$. The space of the configurations of the state values of the considered sites set is denoted by the product space.

$$Z = \prod_{i=1}^{N} Z_i$$ (3.1)

Then, the $p(Z)$ is a random field, if the following condition is satisfied

$$p(Z) > 0, Z \in Z$$ (3.2)

Now, let us denote the neighborhood of site $i$ as $N_i$, $i \notin N_i$ and $i \in N_j \iff j \in N_i$. The neighborhood system on $S$ is defined as $N = (N_i, i \in S)$. Then, the previously considered random field $p(Z)$ is an MRF with respect to a neighborhood system $N$ [91], [92] if and only if

$$p(z_i | z_{S-\{i\}}) = p(z_i | z_{N_i})$$ (3.3)

Hammersley-Clifford theorem is proposed in [95], [96] to establish the equivalence of MRF and Gibbs random field. According to this theorem, a Gibbs distribution is equivalently characterized by a MRF and vice versa. Thus, an MRF given in Eq.(3.3) is rewritten as:

$$p(Z|\beta) = W^{-1} \exp (-U(Z|\beta))$$ (3.4)

where, $W$ is a normalizing constant called the partition function

$$W(\beta) = \sum_{z \in Z} \exp (-U(Z|\beta))$$ (3.5)
and \( U(Z|\beta) \) is an energy function. This energy function is a sum of clique potentials over all possible cliques of the form.

\[
U(Z|\beta) = \sum_{c \in C} V_c(Z|\beta)
\]  

(3.6)

\( V_c \) stands for the clique potential associated with the clique \( c \). And \( C \) is a label of subsets of the sites that contains sites that are all neighbors, and are known as cliques. \( \beta \) is a parameter of the clique potentials known as the inverse supercritical temperature. The computation of the term \( W \) in Eq.(3.5) involves all possible realizations \( Z \) of the MRF which is hardly ever feasible, in terms of computational requirements. To overcome this problem, an approximation of the likelihood in Eq.(3.4) is the pseudo-likelihood introduced by Besag [97] and defined as

\[
p(Z|\beta) = \prod_{j=1}^{N} p(z_i|z_N; \beta)
\]  

(3.7)

where, each term in the product is to compute

\[
p(z_i|z_N; \beta) = \frac{\exp \left( - \sum_{c \in C} V_c(Z|\beta) \right)}{\sum_{z_i} \exp \left( - \sum_{c \in C} V_c(Z|\beta) \right)}
\]  

(3.8)

The probability distribution in Eq.(3.8) is used to obtain estimates of a Markov random field parameters.

### 3.2.2 Hidden Markov models

Let us consider the problem of segmenting an image with \( N \) pixels, \( X = (x_1, x_2, ..., x_N) \), into \( K \) labels. In this model, the observations \( X \) are conditionally independent given \( Z \). According to the GMM based MRF for the pixel
label [59], [63], [64], [65], the density function at an observation \( x_i \) is given by:

\[
f(x_i | \Pi, \Theta) = \sum_{j=1}^{K} \pi_{ij} \Phi(x_i | \Theta_j)
\]  

(3.9)

and the prior distributions \( \pi_{ij} \) are defined by:

\[
\pi_{ij} = p(z_i | z_{N_i}; \beta)
\]  

(3.10)

where, the prior \( \pi_{ij} \) is different for each pixel \( i \) and depends on the neighbors of the pixel. For more details, please refer to [64], [89]. Thus far, the problem has focused on how to estimate the parameters to maximize the following log-likelihood function:

\[
L(\Theta, \Pi | X) = \sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij} \Phi(x_i | \Theta_j) \right\}
\]  

(3.11)

The iterative EM algorithm for estimating the parameters of the component densities is applied to estimate the parameters. The conditional expectation values \( z_{ij} \) of the hidden variables is computed as follows:

\[
z_{ij}^{(t)} = \frac{\pi_{ij}^{(t)} \Phi(x_i | \Theta_j^{(t)})}{\sum_{k=1}^{K} \pi_{ik}^{(t)} \Phi(x_i | \Theta_k^{(t)})}
\]  

(3.12)

The estimates of the means \( \mu_j \) and covariance matrices \( \Sigma_j \) yield:

\[
\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} x_i}{\sum_{i=1}^{N} z_{ij}^{(t)}}
\]  

(3.13)

and,

\[
\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} (x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{N} z_{ij}^{(t)}}
\]  

(3.14)
Finally, the estimate of the inverse temperature parameter $\beta$ yields

$$
\beta^{(t+1)} = \arg \max_{\beta} \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \log p(z_i = j|z_N^{(t)}; \beta)
$$

(3.15)

Figure 3.1: Synthetic image, (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.05 variance), (c): standard GMM, (d): SIMF [63].

From the log-likelihood function in Eq.(3.11), we can see that the main advantage of the GMM based MRF for the pixel labels is that it incorporates spatial dependencies between pixels. Compared to the standard GMM, this approach works well in noisy image segmentation. A synthetic image, shown in Figure 3.1(a), was used to test the effectiveness of this model. The objective is to segment the image into two labels. The image shown in Figure 3.1(b) is made from the original image by corrupting with Gaussian noise. In Figure 3.1(c) and Figure 3.1(d), we present the segmentation results obtained by employing standard GMM and SIMF [63] methods, respectively. As can be easily seen, the effect of noise on the performance of the SIMF is much less as compared to the standard GMM.
3.3 Gaussian Mixture Model based MRF for the Priors of the Pixel Labels

In order to reduce the sensitivity of the segmentation result with respect to noise, several researchers have suggested modifications to incorporate the local spatial interactions between the neighboring pixels. In the models in [58], [72]–[74], the pixel label priors are treated as random variables forming an MRF have been presented in. In [58], the authors proposed a spatially variant finite mixture model (SVFMM) for image segmentation. The model assumes that the density function at an observation $x_i$ is given by:

$$f(x_i | \Pi, \Theta) = \sum_{j=1}^{K} \pi_{ij} \Phi(x_i | \Theta_j)$$  \hspace{1cm} (3.16)

where the Gaussian distribution $\Phi(x_i | \Theta_j)$ is the same as Eq.(2.23). The prior distribution $\pi_{ij}$ of the pixel $x_i$ belonging to the label $\Omega_j$ should satisfy the following constraints:

$$0 \leq \pi_{ij} \leq 1 \text{ and } \sum_{j=1}^{K} \pi_{ij} = 1$$  \hspace{1cm} (3.17)

Note that the observation $x_i$ in Eq.(3.16) is modeled as statistically independent of the label $\Omega_j$. The joint conditional density [72] of the data set $X = (x_1, x_2, ..., x_N)$ can be modeled as:

$$p(X | \Pi, \Theta) = \prod_{i=1}^{N} f(x_i | \Pi, \Theta) = \prod_{i=1}^{N} \left[ \sum_{j=1}^{K} \pi_{ij} \Phi(x_i | \Theta_j) \right]$$  \hspace{1cm} (3.18)

Since the observation $x_i$ is considered to be independent given the pixel label, the spatial correlation between the neighboring pixels is not taken into account.
As a result, the segmented image is sensitive to noise, varying illumination and other environmental factors such as wind, rain or camera movements. To overcome this problem, MRF distribution [99] is applied to incorporate the spatial correlation amongst label values:

\[
p(\Pi) = W^{-1} \exp \left\{ -\frac{1}{T} U(\Pi) \right\} \tag{3.19}
\]

where, \( W \) is a normalizing constant, \( T \) is a temperature constant. And \( U(\Pi) \) is the smoothing prior. The posterior probability density function given by Bayes’ rules can be written as:

\[
p(\Pi, \Theta | X) \propto p(X | \Pi, \Theta) p(\Pi) \tag{3.20}
\]

By incorporating Eq.(3.20), the log-likelihood function can be derived as:

\[
L(\Pi, \Theta | X) = \log(p(\Pi, \Theta | X))
\]

\[
= \sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij} \Phi(x_i | \Theta_j) \right\} + \log p(\Pi)
\]

\[
= \sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij} \Phi(x_i | \Theta_j) \right\} - \log W - \frac{1}{T} U(\Pi) \tag{3.21}
\]

Depending on the type of energy \( U(\Pi) \) selected, we can have different kinds of models. In SVFMM method in [58], the value of \( T \) is set to one \((T=1)\), and the Gibbs function for the priors \( p(\Pi) \) is given by:

\[
p(\Pi) = \frac{1}{Z} \exp \left( -U(\Pi) \right) ; \text{ where : } U(\Pi) = \beta \sum_{i=1}^{N} \sum_{j=1}^{K} \sum_{m \in N_i} \left( \pi_{ij} - \pi_{mj} \right)^2 \tag{3.22}
\]
where, $\Pi$ is the parameter set; $\Pi = \{\pi_{ij}\}; i=1,2,...,N; j=1,2,...,K$. And $\beta$ is a scalar. The log-likelihood function (ignore the constant) is given by:

$$L(\Pi, \Theta|X) = \sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij} \Phi(x_{i}|\Theta_{j}) \right\} + \log p(\Pi) = \sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij} \Phi(x_{i}|\Theta_{j}) \right\} - \beta \sum_{i=1}^{N} \sum_{j=1}^{K} \sum_{m \in N_i} (\pi_{ij} - \pi_{mj})^2$$

(3.23)

Compared to the log-likelihood function of the standard GMM in Eq.(2.25), the log-likelihood function in Eq.(3.23) is quite complex. In order to maximize this likelihood with respect to the parameters $\Xi = \{\Theta, \Pi\} = (\mu_{j}, \Sigma_{j}, \pi_{ij})$, an iterative EM algorithm is adopted. Application of the complete data condition in [58], maximizing the log-likelihood function $L(\Pi, \Theta|X)$ in Eq.(3.23) will lead to an increase in the value of the objective function $H(\Pi, \Theta|X)$.

$$H(\Pi, \Theta|X) = \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \{ \log \pi_{ij} + \log \Phi(x_{i}|\Theta_{j}) \} - \beta \sum_{i=1}^{N} \sum_{j=1}^{K} \sum_{m \in N_i} (\pi_{ij} - \pi_{mj})^2$$

(3.24)

where the conditional expectation values $z_{ij}$ of the hidden variables can be computed as follows:

$$z_{ij}^{(t)} = \frac{\pi_{ij}^{(t)} \Phi(x_{i}|\Theta_{j}^{(t)})}{\sum_{k=1}^{K} \pi_{ik}^{(t)} \Phi(x_{i}|\Theta_{k}^{(t)})}$$

(3.25)

Let us now consider the derivation of the function $H(\Pi, \Theta|X)$ with the means $\mu_{j}$ at the $(t+1)$ iteration step. We have:

$$\frac{\partial H}{\partial \mu_{j}} = \sum_{i=1}^{N} z_{ij}^{(t)} \left[ -\frac{1}{2} (2\Sigma_{j}^{-1}\mu_{j} - 2\Sigma_{j}^{-1}x_{i}) \right]$$

(3.26)
The solution of $\frac{\partial J}{\partial \mu_j} = 0$ yields the minimizer of $\mu_j$ at the $(t+1)$ step:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} x_i}{\sum_{i=1}^{N} z_{ij}^{(t)}}$$  \hspace{1cm} (3.27)

Thus, setting the derivative of the function in Eq.(3.24) with respect to $\Sigma_j^{-1}$ at the $(t+1)$ iteration step we have:

$$\frac{\partial H}{\partial \Sigma_j} = \sum_{i=1}^{N} z_{ij}^{(t)} \left[ 1/2 \Sigma_j - 1/2 (x_i - \mu_j)(x_i - \mu_j)^T \right]$$  \hspace{1cm} (3.28)

and, equating it to zero yields:

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} (x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{N} z_{ij}^{(t)}}$$  \hspace{1cm} (3.29)

However, due to the complexity of the log-likelihood function in Eq.(3.23), the M-step of EM algorithm cannot evaluate the prior distribution $\pi_{ij}$ in a closed form. In order to maximize objective function $H(\Pi, \Theta|X)$ with respect $\pi_{ij}$ [58], [72], we set its derivative equal to zero and obtain the following quadratic expression:

$$4\beta N_i \left( \pi_{ij}^{(t+1)} \right)^2 - 4\beta \pi_{ij}^{(t+1)} \sum_{m \in N_i} \pi_{mj} - z_{ij}^{(t)} = 0$$  \hspace{1cm} (3.30)

where $N_i$ stands for the set of neighbors falling in a window around the pixel $x_i$. The above equation has two roots:

$$\pi_{ij}^{(t+1)} = \frac{\sum_{m \in N_i} \pi_{mj} \pm \sqrt{\left( \sum_{m \in N_i} \pi_{mj} \right)^2 + \frac{N_i}{\beta} z_i^{(t)}}}{2N_i}$$  \hspace{1cm} (3.31)

We select the root with the positive sign + since it yields $\pi_{ij} \geq 0$. The above equation provides a straightforward update for the values of label parameters.
\( \pi_{ij} \) of each pixel \( i \) at the M-step of every EM iteration. However, note that the prior distribution \( \pi_{ij} \) should satisfy the constraints in Eq.(3.17). Therefore, the algorithm becomes even more computationally complex. In [58], [72], [73], a large amount of computational power is utilized to solve the constrained optimization problem of the prior distribution \( \pi_{ij} \).

To illustrate the computational cost of this approach, an image (245x245 image resolution) with three labels as shown in Figure 3.2(a) is used. The image shown in Figure 3.2(b) is made from the original image by corrupting with Gaussian noise. All methods are initialized with the same initial condition and are performed on a PC (Core i3 with 4GB RAM) until convergence by using MATLAB in the Windows environment. As shown in Figure 3.2(d), although, SVFMM [72] demonstrates a higher degree of robustness with respect to noise, it is still low in terms of the speed. Note that SVFMM takes 246.1 seconds to segment this image. Compared to SVFMM, standard GMM is fast (5.5 seconds). However, the segmentation accuracy of standard GMM is quite poor as shown in Figure 3.2(c).

![Figure 3.2: Synthetic image, (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.01 variance), (c): standard GMM (time = 5.5s), (d): SVFMM [72] (time = 246.1s).](image)

43
Another limitation of this model, as mentioned in [58], is that it requires a greater number of parameters compared to the standard GMM. In order to segment an image consisting of \( N \) pixels into \( K \) labels, we have to deal with \( K \times (2+N) \) parameters (\( K \) parameters of \( \mu_j \), \( K \) parameters of \( \Sigma_j \) and \( NK \) parameters of \( \pi_{ij} \)). This implies that the larger the image, the more the number of parameters that we have to estimate.

### 3.4 Conclusions

In this chapter, mixture models based on the Markov random fields are presented. Compared with the standards GMM, the major difference is that instead of using the common prior distribution \( \pi_j \) for all pixels, the prior distribution \( \pi_{ij} \) of the mixture models based on the Markov random fields are different for each pixel and depends on the neighbors of the pixel of interest and the corresponding parameters. The spatial relationship between neighboring pixels is taken into account. Although the effect of noise on the final segmentation result is reduced, these mixture models lack enough robustness with respect to noise. Besides that, they are too complex, and require a large number of parameters compared to the standard GMM.
Chapter 4

An Extension of the Standard Mixture Model for Image Segmentation

4.1 Introduction

The key concept in the field of image segmentation is the imposing of spatial smoothness. Indeed, methods that do not utilize smoothing for image segmentation, such as standard GMM, standard SMM, K-means lead to very poor results. In the standard GMM method, it can be easily seen that the spatial relationship between neighboring pixels is not taken into account. For this reason, although the standard GMM is a well known and simple method for image segmentation, its segmentation result is sensitive to noise, varying illumination and other environmental factors such as wind, rain or camera
In this chapter, we propose a new model based on the standard GMM that applies to the image classification problem. Our approach differs from those discussed above by the following statements. Firstly, a unique approach accounting for the relationship amongst neighboring pixels is presented. The proposed model is quite similar to the standard GMM and thus, is easy to implement. Secondly, compared to the standard GMM, the main difference in the proposed method is that the prior distribution $\pi_{ij}$ of each label $\Omega_j$ is different for each pixel $x_i$ and depends on its neighboring pixels. Thirdly, compared to the models based on MRF, the proposed model is simple and requires fewer parameters.

The rest of this chapter is organized as follows. In section 4.2, we describe the details of the proposed algorithm. Learning algorithms for the proposed system are presented in section 4.3. In section 4.4, we present the experimental results and conclude with a discussion in section 4.5.

### 4.2 Proposed Method

First, we define a function that represents the weight of each $i$-th pixel for each label $\Omega_j$.

$$\xi_j(x_i) = \exp\left(-\frac{(x_i - c_j)^2}{2b_j^2}\right)$$  \hspace{1cm} (4.1)

where $c_j$ and $b_j$; $j=1,2,\ldots,K$; are parameters whose optimal values can be obtained by utilizing the methodology presented in the following section. For the neighborhood $\mathcal{N}_i$ of the $i$-th pixel, the weight function for each label $\Omega_j$ is
defined as:

$$\vartheta_j(x_i) = \left[ \sum_{m \in \mathcal{N}_i} \exp \left( -\frac{(x_m - c_j)^2}{2b^2_j} \right) \right]^\alpha$$  \hspace{1cm} (4.2)$$

where, $\alpha$ is a parameter, and $\mathcal{N}_i$ is the neighborhood of the $i$-th pixel (a $5 \times 5$ window is used in this method). Next, we propose a novel approach to incorporate the spatial relationships between neighboring pixels into the prior probability distribution $\pi_{ij}$. This prior distribution has different values for each pixel corresponding to each label $\Omega_j$ in the image, given by:

$$\pi_{ij} = \frac{\vartheta_j(x_i)}{\sum_{k=1}^{K} \vartheta_k(x_i)}$$  \hspace{1cm} (4.3)$$

The prior probability $\pi_{ij}$ in Eq.(4.3) is computed subject to the constraints $0 \leq \pi_{ij} \leq 1$ and $\sum_{j=1}^{K} \pi_{ij} = 1$. The density function at an observation $x_i$ is given by:

$$f(x_i|\Pi, \Theta) = \sum_{j=1}^{K} \pi_{ij} \Phi(x_i|\Theta_j)$$  \hspace{1cm} (4.4)$$

The log-likelihood function is given by:

$$L(\Theta, \Pi|X) = \sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij} \Phi(x_i|\Theta_j) \right\}$$  \hspace{1cm} (4.5)$$

When $\pi_{ij} = \pi_{cj}; \forall i, c = 1, 2, ..., N$; the log-likelihood function $L(\Theta, \Pi|X)$ in Eq.(4.5) is the same as that given in Eq.(2.25). Therefore, the standard GMM is a special case of the proposed method. Now, if we compare the log-likelihood function of the proposed method in Eq.(4.5) with the log-likelihood function in Eq.(3.11), we can see that it is very similar to the Gaussian mixture model based MRF for the pixel labels in Section 3.2 with two main differences. First, while the above-mentioned methods introduce a mean-field approximation of
the MRF for $\pi_{ij}$, we propose a new way to incorporate the spatial relationships between neighboring pixels into the prior distribution $\pi_{ij}$, as shown in Eq.(4.3). Considering the formulae in Eq.(4.2) and Eq.(4.3), it can be easily seen that the prior distribution $\pi_{ij}$ in the proposed method acts like a mean filter. For that reason, the image segmentation result obtained by employing the proposed method is robust with respect to noise. We also propose a new method to estimate the model parameters in order to minimize the higher bound on the data negative log-likelihood [83], [82], [7], [77], [78], based on the gradient method that offers a closed form M-step, with computational complexity similar to that of the M-step for Gaussian mixture model.

The next objective is to optimize the parameter set $\Xi = \{\Theta, \Pi\} = \{\mu_j, \sigma_j, c_j, b_j, \alpha\}$ to maximize the log-likelihood function in Eq.(4.5). Since the logarithm is a monotonically increasing function, it is more convenient to consider the negative logarithm of the likelihood function [79], [80] as an error function:

$$J(\Theta, \Pi | X) = -L(\Theta, \Pi | X) = -\sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij} \Phi(x_i | \Theta_j) \right\} \quad (4.6)$$

Applying the complete data in [46], with proper replacements for old parameter (at $t$ iteration step) values with the new ones (at $t+1$ iteration step) in Eq.(4.6), the change in the error function can be expressed as:

$$J(\Theta^{(t+1)}, \Pi^{(t+1)} | X) - J(\Theta^{(t)}, \Pi^{(t)} | X) = -\sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij}^{(t+1)} \Phi(x_i | \Theta_j^{(t+1)}) \times \frac{z_{ij}^{(t+1)}}{\sum_{k=1}^{K} \pi_{ik}^{(t)} \Phi(x_i | \Theta_k^{(t)})} \right\} \quad (4.7)$$
Note, that $z_{ij}^{(t)}$, as shown in Eq.(2.33), always satisfies the conditions: $z_{ij}^{(t)} \geq 0$ and $\sum_{j=1}^{K} z_{ij}^{(t)} = 1$. We can now apply the Jensen’s inequality [81] which states that, given a set of numbers $\lambda_j \geq 0$ and $\sum_{j=1}^{K} \lambda_j = 1$, we have:

$$\log \left( \sum_{j=1}^{K} \lambda_j y_j \right) \geq \sum_{j=1}^{K} \lambda_j \log (y_j) \tag{4.8}$$

From Eq.(4.8), the change in error function in Eq.(4.7) is given by:

$$J(\Theta^{(t+1)}, \Pi^{(t+1)}|X) - J(\Theta^{(t)}, \Pi^{(t)}|X) \leq - \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \log \left\{ \frac{\pi_{ij}^{(t+1)} \Phi(x_i|\Theta_j^{(t+1)})}{\sum_{k=1}^{K} \pi_{ik}^{(t)} \Phi(x_i|\Theta_k^{(t+1)})} \right\} \tag{4.9}$$

Thus, we have to minimize the log-likelihood function with respect to the new parameters (at the $t+1$ iteration step). Therefore, we can drop the terms that depend only on the old parameters (at the $t$ iteration step). The change in the error function can be written in the form:

$$E(\Theta^{(t)}, \Pi^{(t)}|\Theta^{(t+1)}, \Pi^{(t+1)}) = - \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \log \left\{ \pi_{ij}^{(t+1)} \Phi(x_i|\Theta_j^{(t+1)}) \right\} \tag{4.10}$$

The $E$ in Eq.(4.10) can be regarded as an error function. Therefore, maximizing the likelihood $L$ in Eq.(4.5) is then equivalent to minimizing $E$ in Eq.(4.10). The minimization of the error function $E$ with respect to the parameters $\Xi = \{\Theta, \Pi\} = \{\mu_j, \sigma_j, c_j, b_j, \alpha\}; j=1,2,\ldots,K$ will be discussed in detail in the next part. In order to assign labels to each pixel, the posterior probability $z_{ij}$ is used:

$$z_{ij} = \frac{\pi_{ij} \Phi(x_i|\Theta_j)}{\sum_{k=1}^{K} \pi_{ik} \Phi(x_i|\Theta_k)} \tag{4.11}$$
After labeling each pixel using Eq.(4.11), a simple competitive selection is carried out for each pixel, in order to remove the remaining noise. If the $i$-th pixel belongs to label $j; j=1,2,...,K$; and all its neighborhood pixels $\mathcal{N}_i$ belong to the label $k, k \neq j$ and $k=1,2,...,K$, then the $i$-th pixel is set to the label $k$. It is worth noticing that we only use one type of selection criteria in this method: if all the neighbors of a given pixel are assigned to a specific label, the considered pixel is also assigned to the same label. The effect of this simple competitive selection employed in the proposed method is shown in the section containing the experimental results.

### 4.3 Parameter Learning

Thus far, the discussion has focused on probability estimation used to determine the label $\Omega_j$ to which the pixel $x_i$ should be assigned. To generalize the posterior probability $z_{ij}$, we need to adjust the parameters $\Xi = \{\Theta, \Pi\} = (\mu_j, \sigma_j, c_j, b_j, \alpha); j=1,2,...,K$; to minimize the error function $E$ in Eq.(4.10), corresponding to maximizing log-likelihood function $L$ in Eq.(4.5). Note, that the total number of parameters required for the proposed method is only $4K+1$ ($K$ parameters of $\mu_j$, $K$ parameters of $\sigma_j$, $K$ parameters of $c_j$, $K$ parameters of $b_j$ and 1 parameter of $\alpha$), which is less than the number of parameters in the models based on MRF mentioned in the section 3.3.

In this part, instead of utilizing EM algorithm, we employ the gradient method [82], [83], [79], [80] for adjusting the parameters to minimize the error function $E$. The proposed algorithm can be summarized as follows:
**Step 1**: Initialize the parameters $\Xi$ by the following sub-steps:

+ Use K-means to initialize the mean $\mu_j$, and covariance $\sigma_j$. Then, select $c_j = \mu_j$, and $b_j = \sigma_j$. In this method, the initial value of $\alpha$ is set to 14.
+ While $t \leq T$, ($T=5$ in this method), repeat the following sub-steps:

  - Calculate the Gaussian distribution $\Phi(x_i|\Theta_j^{(t)})$ from Eq.(2.23) and the prior probability distribution $\pi_{ij}^{(t)}$ from Eq.(4.3). Next, calculate the posterior probability $z_{ij}^{(t)}$, given in Eq.(4.11):

    \[
    z_{ij}^{(t)} = \frac{\pi_{ij}^{(t)} \Phi(x_i|\Theta_j^{(t)})}{\sum_{k=1}^{K} \pi_{ik}^{(t)} \Phi(x_i|\Theta_k^{(t)})}
    \]

  - Update the parameters $\mu_j$ and $\sigma_j$ using Eq.(2.34) and (2.35):

    \[
    \mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} x_i}{\sum_{i=1}^{N} z_{ij}^{(t)}}; \quad [\sigma_j^{2(t+1)}] = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} (x_i - \mu_j^{(t+1)})^2}{\sum_{i=1}^{N} z_{ij}^{(t)}}
    \]

  - Set $c_j = \mu_j$, and $b_j = \sigma_j$.
  - Increase $t$ by 1.

After finishing Step 1, we obtain the initial values of the parameters $\Xi^{(t)} = (\mu_j, \sigma_j, c_j, b_j, \alpha)$.

**Step 2**: Evaluate $z_{ij}^{(t)}$ given by Eq.(4.11):

\[
z_{ij}^{(t)} = \frac{\pi_{ij}^{(t)} \Phi(x_i|\Theta_j^{(t)})}{\sum_{k=1}^{K} \pi_{ik}^{(t)} \Phi(x_i|\Theta_k^{(t)})}
\]
where, \( \Phi(t)(x_i|\Theta_j) \) and \( \pi_{ij}^{(t)} \) are calculated from Eq.(2.23) and Eq.(4.3), respectively.

**Step 3:** Update parameters \( \Xi = \{\Theta, \Pi\} = (\mu_j, \sigma_j, c_j, b_j, \alpha) \) to obtain the new parameters \( \Theta^{(t+1)} \), which can then be calculated and updated by using the gradient method [82], [83], [85].

\[
\Xi^{(t+1)} = \Xi^{(t)} - \eta \nabla E(\Xi^{(t)}) \tag{4.12}
\]

where, \( \nabla E(\Xi^{(t)}) = [\partial E/\partial \mu_j, \partial E/\partial \sigma_j, \partial E/\partial c_j, \partial E/\partial b_j, \partial E/\partial \alpha] \). Details of the formulae \( \partial E/\partial \mu_j, \partial E/\partial \sigma_j, \partial E/\partial c_j, \partial E/\partial b_j \) and \( \partial E/\partial \alpha \) used to update the parameters \( \mu_j, \sigma_j, c_j, b_j \) and \( \alpha \) are given in the Appendix A. \( \eta \) is the learning rate and its value is sufficiently small. In this method, we have selected \( \eta = 10^{-5} \).

**Step 4:** Check for convergence of either the negative log-likelihood function, or the parameter values. If the convergence criterion is not satisfied, then set \( \Xi^{(t)} = \Xi^{(t+1)} \), and return to step 2.

### 4.4 Experiments

In this section, the performance of the proposed algorithm is compared to the K-means [86], standard GMM [75], SVFMM [72], neighborhood expectation maximization (NEM) [48], [87], iterated conditional model (ICM) [88], mode field (MODEF), SIMF, MEANF [63] and [64], fast generalized fuzzy
c-means (FGFCM) [35], and hidden markov random field based fuzzy c-means (HMRF-FCM) [89]. The source code for the SVFMM algorithm can be downloaded from http://www.cs.uoi.gr/~kblekas/sw/MAPsegmentation.html. Parameter $\beta$ in SVFMM algorithms is assigned a value of 0.1. This method was implemented in the MATLAB environment. For the NEM, ICM, MODEF, SIMF and MEANF methods, we used a software implementation developed by the authors [63], [64] (for the Windows environment) publicly available at http://spacem3.gforge.inria.fr/. Standard GMM, SVFMM, NEM, ICM, MODEF, SIMF and MEANF methods are initialized by the K-means algorithm similar to the initialization of the proposed algorithm. SVFMM methods use a first order neighborhood system, while ICM, MODEF, SIMF and MEANF methods use a second order (8-neighbor) neighborhood system. For these methods, the standard isotropic Potts model is used and the temperature value $\beta$ is heuristically optimized. For the FGFCM method, we have selected $\lambda_s=3$ and $\lambda_g=6$. The optimally selected value of the fuzziness parameter $\lambda$ in the HMRF-FCM is set to five. These methods were run until convergence of the iteration steps.

To test the robustness of the proposed method, different types of noise with varying levels have been introduced to each image. The proposed method was implemented and tested on synthetic and real-world images. For the synthetic images, to compare the results obtained, the misclassification ratio (MCR) [91], as shown in Eq.(2.27), has been used, which is the number of misclassified pixels divided by the total number of pixels. For natural real-
world images, we employ the probabilistic rand (PR) index [93], as shown in Eq.(2.28), to compare the results of these methods. The proposed method was implemented and tested on a PC (Pentium 4, running at 3 GHz with 1GB of RAM) in MATLAB compiled visual C environment. The algorithms were tested using synthetic and real-world images.

4.4.1 Synthetic Images

Figure 4.1: The first experiment (128x128 image resolution), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.05 variance), (c): K-means (MCR = 17.315%), (d): standard GMM (MCR = 33.982%), (e): SVFMM (MCR = 13.671%), (f): NEM (MCR = 17.034%), (g): ICM (MCR = 9.605%), (h): MODEF (MCR = 7.781%), (i): SIMF (MCR = 7.725%), (k): MEANF (MCR = 7.721%), (l): HMRF-FCM (MCR = 0.823%), (m): The proposed method (MCR = 0.653%).

In the first experiment, a synthetic image (128x128 image resolution) similar to the one used in [91], as shown in Figure 4.1(a), was used to compare the performance of the proposed algorithm with others. The image has three labels (K=3) with luminance values [0, 0.5, 1]. The image shown in Figure 4.1(b) is obtained by corrupting the original image with Gaussian noise (0
Table 4.1: Comparison of the proposed method with other methods in terms of MCR (%), for the first experiment.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Gaussian noise (0 mean, var)</th>
<th>var=0.02</th>
<th>var=0.03</th>
<th>var=0.04</th>
<th>var=0.05</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td></td>
<td>5.212</td>
<td>9.985</td>
<td>13.721</td>
<td>17.315</td>
</tr>
<tr>
<td>Standard GMM</td>
<td></td>
<td>29.974</td>
<td>31.332</td>
<td>32.445</td>
<td>33.982</td>
</tr>
<tr>
<td>SVFMM</td>
<td></td>
<td>4.168</td>
<td>7.385</td>
<td>10.546</td>
<td>13.671</td>
</tr>
<tr>
<td>NEM</td>
<td></td>
<td>3.584</td>
<td>10.101</td>
<td>15.130</td>
<td>17.034</td>
</tr>
<tr>
<td>ICM</td>
<td></td>
<td>0.385</td>
<td>1.025</td>
<td>2.386</td>
<td>9.605</td>
</tr>
<tr>
<td>MODEF</td>
<td></td>
<td>0.598</td>
<td>1.837</td>
<td>3.757</td>
<td>7.781</td>
</tr>
<tr>
<td>SIMF</td>
<td></td>
<td>0.586</td>
<td>1.831</td>
<td>3.661</td>
<td>7.725</td>
</tr>
<tr>
<td>MEANF</td>
<td></td>
<td>0.585</td>
<td>1.800</td>
<td>3.659</td>
<td>7.721</td>
</tr>
<tr>
<td>HMRF-FCM</td>
<td></td>
<td>0.177</td>
<td>0.391</td>
<td>0.493</td>
<td>0.823</td>
</tr>
<tr>
<td>The proposed method</td>
<td></td>
<td>0.122</td>
<td>0.290</td>
<td>0.415</td>
<td>0.653</td>
</tr>
</tbody>
</table>

mean, 0.05 variance). Figure 4.1(c) presents the segmentation result obtained by employing K-means algorithm. This result is used in the initialization step for all the remaining methods. From Figure 4.1(d) to Figure 4.1(l), we present the segmentation results obtained by employing standard GMM, SVFMM, NEM, ICM, MODEF, SIMF, MEANF and HMRF-FCM, respectively. As can be seen, the accuracy of the standard GMM method is poor compared to SVFMM in Figure 4.1(e). However, the SVFMM method requires a large number of parameters in the process of estimation \(K(2+N)=49158\) parameters in this example). In Figure 4.1(l), the HMRF-FCM algorithm reduces the effect of noise significantly and can segment the image well. However, the proposed method in Figure 4.1(m) can segment the image with a better result. The results obtained with varying levels of noise are presented in Table 4.1. As can be seen, the proposed method has a lower MCR compared with the
other methods.

Figure 4.2: The second experiment (128x128 image resolution), (a): original image, (b): Corrupted original image with mixed noise (salt and pepper noise (0.03%) + Gaussian noise (0 mean, 0.01 variance)), (c): K-means (MCR = 24.609%), (d): standard GMM (MCR = 40.209%), (e): SVFMM (MCR = 22.338%), (f): ICM (MCR = 22.001%), (g): MODEF (MCR = 12.744%), (h): SIMF (MCR = 13.307%), (i): MEANF (MCR = 12.200%), (k) FGFCM (MCR = 5.562%), (l) HMRF-FCM (MCR = 3.692%), (m): The proposed method (MCR = 2.789%).

Figure 4.2(a) shows the synthetic image (128x128 image resolution) which consists of four labels ($K=4$) with luminance values [0, 1/3, 2/3, 1], used in the second experiment. The image shown in Figure 4.2(b) is obtained by corrupting the original image with mixed noise. First corrupting the original image with salt & pepper noise (noise=0.03), and then adding Gaussian noise (0 mean, 0.01 variance). Figure 4.2(d)–(l), we present the segmentation results of standard GMM, SVFMM, ICM, MODEF, SIMF MEANF, FGFCM and HMRF-FCM, respectively. Amongst these methods, FGFCM and HMRF-FCM classify the image with the lowest MCR. However, as compared with the other methods, the segmentation result obtained by employing the proposed
Table 4.2: Comparison of the proposed method with other methods in term of MCR (%), for the second experiment.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Gaussian noise (0 mean, var)</th>
<th>Mixed Noise: Salt &amp; Pepper Noise (sp) + Gaussian Noise (0 mean, var)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>var=0.01 var=0.03 sp=0.03 var=0.01</td>
<td>sp=0.03 var=0.03</td>
</tr>
<tr>
<td>Standard GMM</td>
<td>27.624 37.634</td>
<td>29.290 40.209</td>
</tr>
<tr>
<td>SVFMM</td>
<td>4.687 19.860</td>
<td>6.347 22.338</td>
</tr>
<tr>
<td>ICM</td>
<td>0.463 17.126</td>
<td>3.118 22.001</td>
</tr>
<tr>
<td>MODEF</td>
<td>0.512 5.120</td>
<td>3.088 12.744</td>
</tr>
<tr>
<td>SIMF</td>
<td>0.511 4.235</td>
<td>3.125 13.307</td>
</tr>
<tr>
<td>MEANF</td>
<td>0.494 4.095</td>
<td>3.082 12.200</td>
</tr>
<tr>
<td>FGFCM</td>
<td>0.262 2.777</td>
<td>0.676 5.562</td>
</tr>
<tr>
<td>HMRF-FCM</td>
<td>0.225 2.032</td>
<td>0.643 3.692</td>
</tr>
<tr>
<td>Proposed method</td>
<td>0.189 1.007</td>
<td>0.469 2.789</td>
</tr>
</tbody>
</table>

method, as shown in Figure 4.2(l), demonstrates a higher degree of robustness with respect to the given level of noise. Note that the initialization for all algorithms was carried out by using K-means algorithm as shown in Figure 4.2(c). Table 4.2 contains the results obtained for all methods, for varying levels of noise. As can be easily seen, the proposed method outperforms other methods, with a lower MCR.

In the third experiment, an image with four labels ($K=4$) with luminance values $[0, 1/3, 2/3, 1]$ as shown in Figure 4.3(a) was used to test the effect of simple competitive selection used in the proposed method. We also added this selection step to MODEF method to determine its effect on the result. The image shown in Figure 4.3(b) is obtained by corrupting the origi-
inal image with Gaussian noise (0 mean, 0.03 variance). The results for the MODEF method, and the proposed method, without and with the simple competitive selection are shown in Figure 4.3(c), Figure 4.3(d), Figure 4.3(e) and Figure 4.3(f), respectively. It can be easily seen that even though by employing simple competitive selection, there is a slight decrease in the MCR for both MODEF and the proposed methods, its main effect is mostly qualitative in nature. Thus, the significant quantitative difference between the MCRs for both these methods can be attributed to the higher degree of robustness of the proposed method with respect to noise.

Figure 4.3: Affect of the simple competitive selection, (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.03 variance), (c): MODEF (MCR = 3.512%), (d): MODEF with the simple competitive selection (MCR = 3.117%), (e): proposed method without the simple competitive selection (MCR = 0.199%), (f): proposed method with the simple competitive selection (MCR = 0.197%).

4.4.2 Natural Images

Analysis of real-world outdoor scenes is a challenging problem for image segmentation. It is quite hard to come up with a good model for objects such as flowers, trees, birds etc. In this set of experiments, we compare the performance of various algorithms on real world images obtained from the Berkeley’s
Figure 4.4: Images from the Berkeleys grayscale image segmentation dataset, (a): 135069, (b): 124084, (c): 58060, (d): 353013 with Gaussian noise (0 mean, 0.001 variance), (e): 239007, (f): 46076, (g): 15088 with Gaussian noise (0 mean, 0.005 variance), (h): 374067 with Gaussian noise (0 mean, 0.01 variance), (i): 302003 with Gaussian noise (0 mean, 0.01 variance).

image segmentation dataset [98]. This set of grayscale images includes natural images along with their ground truth segmentation results provided by human subjects. PR index, previously introduced in [93], was used to obtain the performance evaluation results in this subsection. It contains values that are between 0 and 1, with values closer to 1 indicating a good result.

In the first experiment, a set of real world grayscale images with and without artificial noise were used to evaluate the performance of the proposed method against K-means, standard GMM, MODEF, SIMF, MEANF, FGFCM, HMRF-FCM methods. Table 4.3 contains the cumulative results obtained for all methods, for the given set of real world images. As can be easily seen, on average, the proposed method outperforms other methods with a
Figure 4.5: Image segmentation results obtained by employing the proposed method, (a): 135069, (b): 124084, (c): 58060, (d): 353013 with Gaussian noise (0 mean, 0.001 variance), (e): 239007, (f): 46076, (g): 15088 with Gaussian noise (0 mean, 0.005 variance), (h): 374067 with Gaussian noise (0 mean, 0.01 variance), (i): 302003 with Gaussian noise (0 mean, 0.01 variance).

higher PR. Figure 4.5 presents the segmentation results obtained by employing the proposed method for the given set of images shown in Figure 4.4.

In the second experiment, we tried to segment the real-world grayscale image (481x321) as shown in Figure 4.6(a) into four labels (K=4): “tree”, “near mountain”, “far mountain”, “sky”. Images in Figure 4.6(b), Figure 4.6(c), Figure 4.6(e) and Figure 4.6(h) show the result obtained by using the K-means, standard GMM, ICM and FGFCM methods. As can be seen, the accuracy of segmentation for these methods is quite poor. The sharp edge between the “near mountain” and the “far mountain”, and between the “far mountain” and the “sky” is lost because of noise. Extraction accuracies of the SVFMM, MODEF and MEANF methods are shown in Figure 4.6(d), Figure
Table 4.3: Comparison of image segmentation results on Berkeley’s grayscale image segmentation dataset: Probabilistic Rand (PR) Index.

<table>
<thead>
<tr>
<th>Image</th>
<th>Noise (var)</th>
<th>lab</th>
<th>Sta. GMM</th>
<th>SIMF</th>
<th>MEANF</th>
<th>FGF CM</th>
<th>HMRF-FCM</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td>135069</td>
<td>-</td>
<td>2</td>
<td>0.982</td>
<td>0.983</td>
<td>0.983</td>
<td>0.549</td>
<td>0.984</td>
<td>0.985</td>
</tr>
<tr>
<td>124084</td>
<td>-</td>
<td>3</td>
<td>0.499</td>
<td>0.514</td>
<td>0.513</td>
<td>0.521</td>
<td>0.526</td>
<td>0.558</td>
</tr>
<tr>
<td>69020</td>
<td>-</td>
<td>3</td>
<td>0.486</td>
<td>0.555</td>
<td>0.556</td>
<td>0.554</td>
<td>0.559</td>
<td>0.605</td>
</tr>
<tr>
<td>12003</td>
<td>-</td>
<td>3</td>
<td>0.515</td>
<td>0.614</td>
<td>0.615</td>
<td>0.605</td>
<td>0.618</td>
<td>0.623</td>
</tr>
<tr>
<td>58060</td>
<td>-</td>
<td>3</td>
<td>0.568</td>
<td>0.618</td>
<td>0.611</td>
<td>0.571</td>
<td>0.615</td>
<td>0.622</td>
</tr>
<tr>
<td>239007</td>
<td>-</td>
<td>3</td>
<td>0.652</td>
<td>0.662</td>
<td>0.660</td>
<td>0.655</td>
<td>0.668</td>
<td>0.671</td>
</tr>
<tr>
<td>46076</td>
<td>-</td>
<td>4</td>
<td>0.812</td>
<td>0.826</td>
<td>0.824</td>
<td>0.807</td>
<td>0.826</td>
<td>0.828</td>
</tr>
<tr>
<td>55067</td>
<td>-</td>
<td>3</td>
<td>0.843</td>
<td>0.881</td>
<td>0.881</td>
<td>0.879</td>
<td>0.888</td>
<td>0.891</td>
</tr>
<tr>
<td>353013</td>
<td>0.001</td>
<td>3</td>
<td>0.680</td>
<td>0.736</td>
<td>0.722</td>
<td>0.687</td>
<td>0.740</td>
<td>0.742</td>
</tr>
<tr>
<td>310007</td>
<td>0.005</td>
<td>7</td>
<td>0.628</td>
<td>0.765</td>
<td>0.764</td>
<td>0.648</td>
<td>0.776</td>
<td>0.784</td>
</tr>
<tr>
<td>61060</td>
<td>0.005</td>
<td>3</td>
<td>0.586</td>
<td>0.678</td>
<td>0.682</td>
<td>0.632</td>
<td>0.679</td>
<td>0.681</td>
</tr>
<tr>
<td>15088</td>
<td>0.005</td>
<td>3</td>
<td>0.844</td>
<td>0.860</td>
<td>0.867</td>
<td>0.844</td>
<td>0.869</td>
<td>0.871</td>
</tr>
<tr>
<td>24063</td>
<td>0.005</td>
<td>3</td>
<td>0.765</td>
<td>0.815</td>
<td>0.818</td>
<td>0.796</td>
<td>0.815</td>
<td>0.826</td>
</tr>
<tr>
<td>374067</td>
<td>0.01</td>
<td>4</td>
<td>0.646</td>
<td>0.706</td>
<td>0.704</td>
<td>0.673</td>
<td>0.705</td>
<td>0.708</td>
</tr>
<tr>
<td>302003</td>
<td>0.01</td>
<td>3</td>
<td>0.686</td>
<td>0.718</td>
<td>0.717</td>
<td>0.710</td>
<td>0.715</td>
<td>0.727</td>
</tr>
<tr>
<td>Mean</td>
<td>-</td>
<td>-</td>
<td>0.679</td>
<td>0.729</td>
<td>0.728</td>
<td>0.675</td>
<td>0.732</td>
<td>0.741</td>
</tr>
</tbody>
</table>

4.6(f) and Figure 4.6(g). In the left hand side of the results, looking closely in the “tree” area, it can be seen that there is a small portion of pixels that have been misclassified. HMRF-FCM methods in Figure 4.6(i) can produce a better segmentation. The sharp edge between mountains is clearly defined, and there was no missing region as in the MEANF method. However, the proposed method in Figure 4.6(k), can better classify with more detail along the sharp edge between “near mountain” and “far mountain”, as compared with HMRF-FCM methods.

In order to further test the accuracy and determine the efficiency of the
Figure 4.6: Grayscale image segmentation (55067), (a): original image, (b): K-means (PR = 0.879), (c): standard GMM (PR = 0.843), (d): SVFMM (PR = 0.882), (e): ICM (PR = 0.880), (f): MODEF (PR = 0.882), (g): MEANF (PR = 0.881), (h): FGFCM (PR = 0.879), (i): HMRF-FCM (PR = 0.887), (k): The proposed method (PR = 0.891).

proposed algorithm, another real word grayscale image was used. The image shown in Figure 4.7(b) is obtained by corrupting the original image in Figure 4.7(a) with Gaussian noise (0 mean, 0.005 variance). Figure 4.7(c)–(k), present the segmentation results obtained by employing K-means, standard GMM, SVFMM, MODEF, MEANF, FGFCM, HMRF-FCM and the proposed method, respectively. From visual inspection of the results, K-means, standard GMM, SVFMM methods are unable to segment the image successfully. FGFCM method demonstrates a better performance compared to the standard GMM method. However, the effect of noise on the final segmented image is still quite high. MODEF, MEANF, HMRF-FCM algorithms can produce a better segmentation. However, compared with these algorithms, the effect of noise in the top right hand side of the final segmented image, obtained by employing the proposed method is far less. Moreover, we also notice that the details of the balcony are better preserved by the proposed method compared
Figure 4.7: Noisy grayscale image segmentation (24063), (a): original image, (b): corrupted original image with Gaussian noise (0 mean, 0.005 variance), (c): K-means (PR = 0.778), (d): standard GMM (PR = 0.765), (e): SVFMM (PR = 0.787), (f): MODEF (PR = 0.814), (g): MEANF (PR = 0.818), (h): FGFCM (PR = 0.796), (i): HMRF-FCM (PR = 0.815), (k): The proposed method (PR = 0.826).

In the final experiment, a real world image (481x321 image resolution), as shown in Figure 4.8(a), is used for evaluating the computational cost. The objective is to segment this image into two labels ($K=2$). The K-means method was implemented in Visual C environment. For the GMM, ICM, MODEF, SIMF, MEANF methods, we used the software implementation developed by the authors [7], [14] (for Windows environment, C++ Language) publicly available at http://spacem3.gforge.inria.fr/. The proposed method was implemented in MATLAB compiled visual C environment. All experiments were performed on a PC (Pentium 4, running at 3 GHz with 1GB of RAM). These methods were run until convergence. The segmentation results obtained by employing K-means, standard GMM, ICM, MODEF, SIMF, MEANF and the proposed method are shown in Figure 9(b)–(h), respectively.
Once more, the proposed method demonstrates robustness with respect to noise, yielding a better segmentation result. K-means is the quickest method taking 0.7 seconds, while the slowest is MEANF taking 445 seconds. The proposed method takes 61 seconds and converges after 8 iterations, as shown in Figure 4.9. However, as shown Figure 4.8, the proposed method performs quite well in segmenting the image into two labels.

Figure 4.8: Computational cost (in seconds) comparison, (a): original image, (b): K-means (0.7 sec), (c): standard GMM (36 sec), (d): ICM (169 sec), (e): MODEF (390 sec), (f): SIMF (432 sec), (g): MEANF (445 sec), (h): The proposed method (61 sec).

Figure 4.9: Minimization Progress of the negative log-likelihood function of the proposed algorithm, for the final experiment.
4.5 Conclusions

In this chapter, we have presented an extension of the standard GMM for grayscale image segmentation. We have proposed a novel approach to incorporate the spatial relationships between neighboring pixels into the standard GMM model. Differing from the standard GMM, the prior distribution in the proposed model is different for each pixel and depends on the neighbors of the pixel and their corresponding parameters. The proposed model is easy to implement and requires fewer parameters compared to the models based on MRF. We have presented a new way to estimate the unknown parameters of the proposed model, based on the gradient method. The proposed method has been tested with many noisy synthetic and real-world images, thereby demonstrating the excellent performance of the proposed model in segmenting the grayscale images.
Chapter 5

Fast and Robust Spatially Constrained Gaussian Mixture Model for Image Segmentation

5.1 Introduction

In this chapter, a new mixture model for image segmentation is presented. We propose a new way to incorporate spatial information between the neighboring pixels into the Gaussian mixture model (GMM) based on Markov random field (MRF). In comparison to other mixture models that are complex and computationally expensive, the proposed method is fast and easy to implement. In mixture models based on MRF, the M-step of the expectation maximization (EM) algorithm cannot be directly applied to the prior distribution $\pi_{ij}$ for maximization of the log-likelihood with respect to the corresponding pa-
rameters. Compared with these models, our proposed method directly applies the EM algorithm to optimize the parameters, which makes it much simpler. Finally, experimental results obtained by employing the proposed method on many synthetic and real-world grayscale and colored images demonstrate its robustness, accuracy and effectiveness, as compared with other mixture models.

In order to partition an image consisting of \(N\) pixels into \(K\) labels, GMM [58] assumes that each observation \(x_i\) is considered independent of the label \(\Omega_j\). The density function \(f(x_i|\Pi, \Theta)\) at an observation \(x_i\) in Eq.(3.16) is rewritten by:

\[
f(x_i|\Pi, \Theta) = \sum_{j=1}^{K} \pi_{ij} \Phi(x_i|\Theta_j)
\]

where \(\Pi = \{\pi_{ij}\}; \ i=1,\ldots,N; \ j=1,2,\ldots,K;\) is the set of prior distributions modeling the probability that pixel \(x_i\) is in label \(\Omega_j\), which satisfies the constraints in Eq.(3.17). And, \(\Phi(x_i|\Theta_j)\) is the Gaussian distribution, called a component of the mixture.

For the case of a \(D\)-dimensional vector \(x_i\), each Gaussian distribution \(\Phi(x_i|\Theta_j)\) in Eq.(2.6) can be written in the form:

\[
\Phi(x_i|\Theta_j) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma_j|^{1/2}} \exp \left\{ -\frac{1}{2} (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j) \right\}
\]

(5.1)

where \(\Theta = \{\Theta_j\}\) and \(\Theta_j = \{\mu_j, \Sigma_j\}; \ i=1,2,\ldots,N.\) The \(D\)-dimensional vector \(\mu_j\) is the mean, the \(D\times D\) matrix \(\Sigma_j\) is the covariance, and \(|\Sigma_j|\) denotes the determinant of \(\Sigma_j\). Note that the observation \(x_i\) is modeled as statistically independent, the joint conditional density in Eq.(3.18) of the data set
\[ X = (x_1, x_2, ..., x_N) \] can be modeled as:

\[
p(X|\Pi, \Theta) = \prod_{i=1}^{N} f(x_i|\Pi, \Theta) = \prod_{i=1}^{N} \sum_{j=1}^{K} \pi_{ij} \Phi(x_i|\Theta_j)
\]

Since the observation \( x_i \) is considered to be independent given the pixel label, the spatial correlation between the neighboring pixels is not taken into account. As a result, the segmented image is sensitive to noise. To overcome this problem, MRF distribution [99], as shown in Eq.(3.19), is applied to incorporate the spatial correlation amongst label values:

\[
p(\Pi) = W^{-1} \exp \left\{ -\frac{1}{T} U(\Pi) \right\}
\]

By using the Eq.(3.21), the log-likelihood function can be derived as:

\[
L(\Theta, \Pi|X) = \log \left( p(\Pi, \Theta|X) \right) \\
= \sum_{i=1}^{N} \log \left( \sum_{j=1}^{K} \pi_{ij} \Phi(x_i|\Theta_j) \right) + \log p(\Pi)
= \sum_{i=1}^{N} \log \left( \sum_{j=1}^{K} \pi_{ij} \Phi(x_i|\Theta_j) \right) - \log W - \frac{1}{T} U(\Pi)
\]

Depending on the type of energy \( U(\Pi) \), we can have different kinds of models. In the Bayesian auto-logistic model [99], the function \( U(\Pi) \) is chosen to incorporate the spatial correlation, as shown:

\[
U(\Pi) = \sum_{i=1}^{N} \sum_{j=1}^{K} \alpha_{ij} \pi_{ij} + \sum_{i=1}^{N} \sum_{j=1}^{K} \beta_{ijm} \pi_{ij} \pi_{mj}
\]

where \( \alpha_{ij} \) and \( \beta_{ijm} \) form the parameter set. Although this model involves many parameters, its segmentation is not sufficiently robust to noise.

Other mixture models based on MRF for pixel labeling have been successfully applied to image segmentation [58], [72], [73], [100] and different ways
are adopted to select the energy $U(\Pi)$. In [58], $U(\Pi)$ is given by:

$$U(\Pi) = \beta \sum_{i=1}^{N} \sum_{m \in N_i} \sum_{j=1}^{K} (\pi_{ij} - \pi_{mj})^2$$  \hspace{1cm} (5.3)$$

while, in another MRF model based method [72], $U(\Pi)$ is given by:

$$U(\Pi) = \beta \sum_{i=1}^{N} \sum_{m \in N_i} \left[ 1 + \left( \sum_{j=1}^{K} (\pi_{ij} - \pi_{mj})^2 \right)^{-1} \right]^{-1}$$  \hspace{1cm} (5.4)$$

where, $\beta$ in Eq.(5.3) and Eq.(5.4) represent a constant value.

In [73], spatial information is taken into account, and $U(\Pi)$ is given as:

$$U(\Pi) = \sum_{i=1}^{N} \sum_{j=1}^{K} \sum_{s=1}^{S} \left[ \frac{1}{2} \log (\beta_{js}^2) - \frac{1}{2} \left( \frac{\sum_{m \in \partial_i} (\pi_{ij} - \pi_{mj})^2}{\beta_{js}^2} \right) \right]$$  \hspace{1cm} (5.5)$$

where, $S$ is the total number of the considered directions. In the general case, $S$ is equal to four ($S=4$: horizontal, vertical and 2 diagonal directions). $\beta_{js}$ in Eq.(5.5) is a variable parameter.

As shown in Eq.(5.3), Eq.(5.4), and Eq.(5.5), the incorporation of local information adds complexity. In order to maximize the likelihood with respect to the parameters $\Pi$ and $\Theta$, an iterative EM algorithm can be applied. However, due to the complexity of the log-likelihood function, the M-step of EM algorithm cannot be applied directly to the prior distribution $\pi_{ij}$. Note, that the prior distribution $\pi_{ij}$ should satisfy the constraints in Eq.(3.17). Thus, the resulting algorithms are computationally complex and utilize large amounts of computational power to solve the constrained optimization problem of the prior distribution $\pi_{ij}$. For details regarding the maximization of the log-likelihood function, we refer the readers to [58], [72], [73], [100].
5.2 Proposed Method

Various mixture models differ based on the way they derive the strength of the smoothing prior $U(\Pi)$. In [99], given in Eq.(5.2), the smoothing prior $U(\Pi)$ has a simple form, thus, it is easy to optimize the parameter set $\{\Pi, \Theta\}$ to maximize the log-likelihood function. However, one of its main drawbacks is that the segmentation result is not robust to noise. Models in [58], [72], [73] represented by Eq.(5.3)–(5.5), make use of a complex smoothing prior. Their primary disadvantage lies in its additional training complexity. The M-step of the EM algorithm cannot be applied directly to the prior distribution, which therefore corresponds to an increase in the algorithms’ complexity. In order to overcome these disadvantages, we introduce a novel factor $G_{ij}$ defined as:

$$G_{ij}^{(t)} = \exp \left[ \beta \frac{1}{2N_i} \sum_{m \in N_i} (z_{mj}^{(t)} + \pi_{mj}^{(t)}) \right]$$

(5.6)

where, $z_{mj}$ is the posterior probability. $\beta$ is the temperature value that controls the smoothing prior. In this method, it has been set to 12 ($\beta = 12$). $\partial_i$ is the neighborhood of the $i$-th pixel, including itself. A square window of size 5x5 is used in this method. $N_i$ is the number of neighboring pixels around the pixel $x_i$ in this window. By taking a closer look at Eq.(5.6), it can be visualized that the factor $G_{ij}$ in the proposed method acts as a mean filter. The goal is to simply replace each pixel value with the average value of its neighbors, including itself. The main advantage of $G_{ij}$ is the ease of implementation, and incorporation of the spatial relationships amongst neighboring pixels in a simpler metric.
Next, we propose a novel approach to incorporate the spatial information into the smoothing prior. The new smoothing prior $U(\Pi)$ is given by:

$$U(\Pi) = -\sum_{i=1}^{N} \sum_{j=1}^{K} G_{ij}^{(t)} \log \pi_{ij}^{(t+1)}$$

(5.7)

where, $t$ indicates the iteration step. The MRF distribution $p(\Pi)$ is given by:

$$p(\Pi) = W^{-1} \exp \left\{ \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{K} G_{ij}^{(t)} \log \pi_{ij}^{(t+1)} \right\}$$

(5.8)

Given the MRF distribution $p(\Pi)$, the log-likelihood function is written in the form.

$$L(\Theta, \Pi|X) = \sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij} \Phi(x_i|\Theta_j) \right\} - \log W + \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{K} G_{ij}^{(t)} \log \pi_{ij}^{(t+1)}$$

(5.9)

Application of the complete data condition in [46], [58], maximizing the log-likelihood function $L(\Theta, \Pi|X)$ in Eq.(5.9) will lead to an increase in the value of the objective function $H(\Theta, \Pi|X)$.

$$H(\Theta, \Pi|X) = \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \left\{ \log \pi_{ij}^{(t+1)} + \log \Phi(x_i|\Theta_j^{(t+1)}) \right\} - \log W + \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{K} G_{ij}^{(t)} \log \pi_{ij}^{(t+1)}$$

(5.10)

The conditional expectation values $z_{ij}$ of the hidden variables, as shown in Eq.(2.33), is rewritten as follows:

$$z_{ij}^{(t)} = \frac{\pi_{ij}^{(t)} \Phi(x_i|\Theta_j^{(t)})}{\sum_{k=1}^{K} \pi_{ik}^{(t)} \Phi(x_i|\Theta_k^{(t)})}$$

The next objective is to optimize the parameter set $\{\Pi, \Theta\}$ in order to maximize the objective function $H(\Theta, \Pi|X)$ in Eq.(5.10). Similar to the MRF
based methods [58], [72], [73], [100], $W$ and $T$ in Eq.(5.10) are set equal to one ($W=1$, $T=1$). From Eq.(5.10), the new objective function is given by:

$$H(\Theta, \Pi|X) = \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \left\{ \log \pi_{ij}^{(t+1)} + \log \Phi(x_i|\Theta_j^{(t+1)}) \right\} + \sum_{i=1}^{N} \sum_{j=1}^{K} G_{ij}^{(t)} \log \pi_{ij}^{(t+1)}$$  

(5.11)

From Eq.(5.1), the function in Eq.(5.11) can be rewritten as:

$$H(\Theta, \Pi|X) = \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \left\{ \log \pi_{ij}^{(t+1)} - \frac{1}{2} \log \left( 2\pi \right) - \frac{1}{2} \log |\Sigma_j^{(t+1)}| \right\} + \sum_{i=1}^{N} \sum_{j=1}^{K} G_{ij}^{(t)} \log \pi_{ij}^{(t+1)}$$  

(5.12)

To maximize this function, the EM algorithm [52], [56] is applied. Let us now consider the derivation of the function $H(\Theta, \Pi|X)$ with the means $\mu_j$ at the $(t+1)$ iteration step. We have:

$$\frac{\partial H}{\partial \mu_j^{(t+1)}} = \sum_{i=1}^{N} z_{ij}^{(t)} \left[ -\frac{1}{2} \left( 2\Sigma_j^{(t+1)} \mu_j^{(t+1)} - 2\Sigma_j^{(t+1)} x_i \right) \right]$$  

(5.13)

The solution of $\partial H/\partial \mu_j = 0$ yields the minimizer of $\mu_j$ at the $(t+1)$ step:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} x_i}{\sum_{i=1}^{N} z_{ij}^{(t)}}$$  

(5.14)

Thus, setting the derivative of the function $H(\Theta, \Pi|X)$ in Eq.(5.11) with respect to $\Sigma_j^{-1}$ at the $(t+1)$ iteration step we have:

$$\frac{\partial H}{\partial \Sigma_j^{-(t+1)}} = \sum_{i=1}^{N} z_{ij}^{(t)} \left[ \frac{1}{2} \Sigma_j^{(t+1)} - \frac{1}{2} (x_i - \mu_j^{(t+1)}) (x_i - \mu_j^{(t+1)})^T \right]$$  

(5.15)
and, equating it to zero yields:

$$\Sigma^{(t+1)}_j = \frac{\sum_{i=1}^{N} z_{ij}^{(t)} (x_i - \mu_j^{(t+1)}) (x_i - \mu_j^{(t+1)})^T}{\sum_{i=1}^{N} z_{ij}^{(t)}}$$  \hspace{1cm} (5.16)$$

An important consideration is that the prior distribution should satisfy the constraints in Eq.(3.17). In order to enforce these constraints, we use the Lagrange’s multiplier $\eta_i$ for each data point:

$$\frac{\partial}{\partial \pi^{(t+1)}_{ij}} \left[ H - \sum_{i=1}^{N} \eta_i \left( \sum_{j=1}^{K} \pi_{ij}^{(t+1)} - 1 \right) \right] = 0  \hspace{1cm} (5.17)$$

Eq.(5.17) can be rewritten in the following form:

$$\frac{z_{ij}^{(t)}}{\pi_{ij}^{(t+1)}} + \frac{G_{ij}^{(t)}}{\pi_{ij}^{(t+1)}} - \eta_i = 0  \hspace{1cm} (5.18)$$

The constraint $\sum_{j=1}^{K} \pi_{ij} = 1$ enables the Lagrange multiplier $\eta_i$ to satisfy the following condition:

$$\eta_i = 1 + \sum_{j=1}^{K} G_{ij}^{(t)}  \hspace{1cm} (5.19)$$

The necessary condition for determining the prior distribution $\pi_{ij}$ at the $(t+1)$ iteration step becomes:

$$\pi_{ij}^{(t+1)} = \frac{z_{ij}^{(t)} + G_{ij}^{(t)}}{\sum_{k=1}^{K} \left( z_{ik}^{(t)} + G_{ik}^{(t)} \right)}  \hspace{1cm} (5.20)$$

So far, the discussion has focused on estimating $\{\Pi, \Theta\}$ of the model, in order to assign a label $\Omega_j$ to the pixel $x_i$. The various steps of the proposed mixture model incorporating spatial information based on MRF can be summarized as follows:
Step 1: Initialize the parameters \( \{ \Pi, \Theta \} \): the means \( \mu_j \), covariance values \( \Sigma_j \) and prior distributions \( \pi_{ij} \).

Step 2 (E-step):

+ Evaluate the values \( z_{ij} \) in Eq.(2.33) using the current parameter values.

+ Update the factor \( G_{ij} \) by using Eq.(5.6).

Step 3 (M-step): Re-estimate the parameters \( \{ \Pi, \Theta \} \).

+ Update the means \( \mu_j \) by using Eq.(5.14).

+ Update covariance values \( \Sigma_j \) by using Eq.(5.16).

+ Update prior distributions \( \pi_{ij} \) by using Eq.(5.20).

Step 4: Evaluate the log-likelihood in Eq.(5.9) and check the convergence of either the log-likelihood function, or the parameter values. If the convergence criterion is not satisfied, then go to step 2.

Once the parameter-learning phase is complete, every pixel \( x_i \) is assigned to the label with the largest posterior probability \( z_{ij} \) by using Eq.(2.53). In the next section, we will demonstrate the robustness, accuracy and effectiveness of the proposed model, as compared with other GMM based approaches.

5.3 Experiments

In this section, the performance of the proposed algorithm is compared with algorithms based on the K-means algorithm, standard GMM, and mixture models based on MRF such as ICM, SIMF and MEANF. We also compare the results with SVFMM, and the class-adaptive spatially variant finite mixture
model (CA-SVFMM) [73]. For ICM, SIMF and MEANF methods, a standard isotropic Potts model with a second order (8-neighbor) neighborhood system is utilized, where the temperature value $\beta$ is automatically optimized. All these methods were run until convergence. Parameter $\beta$ in SVFMM algorithms is manually set a value of 0.1 based on user experience. This method was implemented in the MATLAB environment. The proposed method was implemented and tested on both synthetic and real-world images. For synthetic images, in order to compare the results obtained, MCR has been used, which is given by the number of misclassified pixels divided by the total number of pixels. All these methods were implemented and tested on a PC (Pentium 4, running at 3 GHz with 1GB of RAM).

5.3.1 Segmentation of Synthetic Images

In the first experiment, a synthetic image (128x128 image resolution) as shown in Figure 5.1(a), was used to compare the performance of the proposed algorithm with others. The image has four labels with luminance values $[0, 1/3, 2/3, 1]$. The image shown in Figure 5.1(b) was obtained by corrupting the original image with Gaussian noise (0 mean, 0.03 variance). Figure 5.1(c) presents the segmentation result obtained by employing the K-means algorithm, which is used during the initialization step for all the remaining methods. In Figure 5.1(d) to Figure 5.1(i), we present the segmentation results obtained by employing standard GMM, SVFMM, CA-SVFMM, ICM, SIMF and MEANF methods respectively. Under given conditions the segmentation accuracy of
the standard GMM method is quite poor. SIMF (MCR = 3.83%) in Figure 5.1(h) and MEANF (MCR = 3.55%) in Figure 5.1(i) demonstrate better classification, and segment the image well. However, the proposed method in Figure 5.1(j) segments the image better with the lowest MCR = 1.13%.

Figure 5.1: The first experiment (128x128 image resolution), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.03 variance), (c): K-means, (d): Standard GMM (MCR = 41.67%), (e): SVFMM (MCR = 23.28%), (f): CA-SVFMM (MCR = 20.29%), (g): ICM (MCR = 20.23%), (h): SIMF (MCR = 3.83%), (i) MEANF (MCR = 3.55%), (j): Proposed method (MCR = 1.13%).

<table>
<thead>
<tr>
<th>Methods</th>
<th>Sta. GMM</th>
<th>SVFMM</th>
<th>CA–S VFMM</th>
<th>ICM</th>
<th>SIMF</th>
<th>MEANF</th>
<th>our method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (sec)</td>
<td>3.2</td>
<td>28.3</td>
<td>218.1</td>
<td>85.4</td>
<td>163.2</td>
<td>121.8</td>
<td>4.9</td>
</tr>
<tr>
<td>MCR (%)</td>
<td>41.67</td>
<td>23.28</td>
<td>20.29</td>
<td>20.23</td>
<td>3.83</td>
<td>3.55</td>
<td>1.13</td>
</tr>
</tbody>
</table>

Moreover, due to the inherent simplicity of the proposed algorithm, it has low computational cost. In this experiment, for ICM, SIMF and MEANF
methods, we used the software implementation (for Linux environment, C++ Language) publicly available at http://mistis.inrialpes.fr/software/SEMMS.html. The source code for the SVFMM algorithm written in MATLAB environment can be downloaded from http://www.cs.uoi.gr/~kblekas/sw/MAPsegmentation.html. Standard GMM, CA-SVFMM and the proposed method were implemented using MATLAB in the Windows environment. All experiments were performed on a PC (Pentium 4, running at 3 GHz with 1GB of RAM) until convergence. Table 5.1 lists the computation time and the misclassification ratio for each of the aforementioned methods. As shown in Table 5.1, standard GMM takes the least amount of time for segmentation, while the slowest one is CA-SVFMM. Although the proposed method comes second (convergence after 4.9 seconds and 50 iterations, as shown in Figure 5.2) in terms of the speed, it has the lowest MCR, and demonstrates a higher degree of robustness with respect to noise.
In Figure 5.3, we show the segmentation results of a synthetic image corrupted by Gaussian noise. The original image with luminance values $[0, 0.5, 1]$ is shown in Figure 5.3(a). The image in Figure 5.3(b) is obtained by adding Gaussian noise (0 mean, 0.05 variance) to the original image. Figure 5.3(d) to Figure 5.3(i), show the segmentation results for standard GMM, SVFMM, CA-SVFMM, ICM, SIMF, MEANF and the proposed method, respectively. Amongst these methods, the proposed method demonstrates a higher degree of robustness to a specified noise level. All algorithms were initialized using the K-means algorithm, as shown in Figure 5.3(c). Results in Table 5.2 confirm that the proposed method is quite fast with the lowest MCR.

The effect of varying noise level on the performance of different methods
Table 5.2: Computational cost (in seconds) comparison for the synthetic image in the second experiment.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Time (sec)</th>
<th>MCR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>1.2</td>
<td>35.02</td>
</tr>
<tr>
<td>SVFMM</td>
<td>117.1</td>
<td>12.01</td>
</tr>
<tr>
<td>CA–SVFMM</td>
<td>147.1</td>
<td>11.16</td>
</tr>
<tr>
<td>ICM</td>
<td>78.3</td>
<td>7.65</td>
</tr>
<tr>
<td>SIMF</td>
<td>108.2</td>
<td>5.65</td>
</tr>
<tr>
<td>MEANF</td>
<td>97.7</td>
<td>5.94</td>
</tr>
<tr>
<td>our method</td>
<td>3.1</td>
<td>1.05</td>
</tr>
</tbody>
</table>

Table 5.3: Comparison of the proposed method with other methods in term of MCR (%), for the third experiment, in the presence of varying levels of noise.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Gaussian Noise (0 mean, var=0.03)</th>
<th>Mixed Noise: Salt &amp; Pepper Noise (sp=0.01, var=0.03) + Gaussian Noise (0 mean, var=0.03)</th>
<th>Mixed Noise: Salt &amp; Pepper Noise (sp=0.02, var=0.05) + Gaussian Noise (0 mean, var=0.05)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard GMM</td>
<td>30.31</td>
<td>31.23</td>
<td>38.22</td>
</tr>
<tr>
<td>SVFMM</td>
<td>18.11</td>
<td>18.88</td>
<td>28.40</td>
</tr>
<tr>
<td>CA-SVFMM</td>
<td>17.73</td>
<td>18.25</td>
<td>28.58</td>
</tr>
<tr>
<td>ICM</td>
<td>5.90</td>
<td>11.65</td>
<td>29.88</td>
</tr>
<tr>
<td>SIMF</td>
<td>2.86</td>
<td>8.16</td>
<td>22.03</td>
</tr>
<tr>
<td>MEANF</td>
<td>2.70</td>
<td>7.14</td>
<td>21.94</td>
</tr>
<tr>
<td>Proposed method</td>
<td>0.21</td>
<td>0.35</td>
<td>0.42</td>
</tr>
</tbody>
</table>

is evaluated in the third experiment. A synthetic image with luminance values [0, 1/3, 2/3, 1] from Figure 5.4(a) is used. This original image is corrupted with varying levels of noise. The goal is to segment the corrupted image into four labels. The results are presented in Table 5.3. ICM method works well when an image is corrupted by low level of noise. However, the segmentation results are poor when the noise level is increased. The effect of noise on the performance of SIMF and MEANF methods is much less when we compared to
ICM. However, comparing to the proposed method, both SIMF and MEANF lose most of the sharpness and details in the segmented image as shown in Figure 5.4. In order to further test the accuracy of the proposed algorithm. The original image is corrupted with varying levels of mixed noise. Firstly, the original image is corrupted with salt & pepper noise, and later Gaussian noise is added. As shown in Table 5.3, the proposed method demonstrates a higher degree of robustness with respect to the given noise level.

In order to further test the accuracy of the proposed methods in Chapter 4 and Chapter 5, we generated a different image that contains two labels with luminance values (0, 1). In this experiment, the original image in Figure 5.5(a) is corrupted with a Gaussian noise (0 mean, 0.1 variance). Figure
Figure 5.5: The fourth experiment (256x256 image resolution), (a): original image, (b): Corrupted original image with Gaussian noise (0 mean, 0.1 variance), (c): Proposed method in Chapter 4 (MCR = 0.10%, time = 8.9s), (j): Proposed method in Chapter 4 (MCR = 0.22%, time = 3.7s).

5.5(c)-(d) show the segmentation the proposed methods in Chapter 4 and Chapter 5, respectively. We can see that the accuracy obtained by employing the proposed method in Chapter 4 (MCR=0.10%) is lower compared to the proposed method in Chapter 5. However, our method in Chapter 4 has high computational cost (8.9 seconds). As shown in Figure 5.5(d), the proposed method in Chapter 5 takes only 3.7 seconds.

5.3.2 Segmentation of Grayscale Natural Images

A real-world grayscale image from the Berkeley’s image segmentation dataset [98] was used to compare the proposed algorithm with other algorithms. The objective is to segment the image into two labels: “buffalo” and “water”. The main difficulty in this experiment is that the effect of noise on the label “water” is high. Figure 5.6(c) to Figure 5.6(g) show the results obtained by implementing SVFMM, CA-SVFMM, SIMF, MEANF, and the proposed method. The initialization for all these algorithms was carried out by using the standard GMM algorithm, as shown in Figure 5.6(b). As can be seen, the segmenta-
tion accuracy of SVFMM, CA-SVFMM, SIMF, MEANF methods, along the object boundaries is quite poor. In this experiment, only the proposed was successfully able to segment the image into “buffalo” and “water” regions. The maximization progress of the log-likelihood of the proposed method is shown in Figure 5.6(h).

Figure 5.6: Grayscale natural image segmentation (80099), (a): original image, (b): Standard GMM, (c): SVFMM, (d): CA–SVFMM, (e): SIMF, (f) MEANF, (g): Proposed method, (h): Maximization progress of the log-likelihood of the proposed method of this experiment.

Another real world grayscale image is used to test the efficiency and effectiveness of our proposed algorithm. We observe that the effect of noise in the bottom left-hand side of this image is high. The objective is to segment the original image, as shown in Figure 5.7(a), into two labels. As can be seen from the results in Figure 5.7, Standard GMM, SVFMM, CA-SVFMM, SIMF, MEANF could not successfully segment this image, and there is a high degree of misclassification. Compared with these methods, the proposed method, as shown in Figure 5.7(g), successfully segments the image.
Figure 5.7: Grayscale natural image segmentation (86016), (a): original image, (b): Standard GMM, (c): SVFMM, (d): CA–SVFMM, (e): SIMF, (f) MEANF, (g): Proposed method, (h): Maximization progress of the log-likelihood of the proposed method.

A grayscale real-world image, shown in Figure 5.8(a), was segmented into three labels: “road”, “grass”, and “tree”. A low segmentation accuracy is obtained with standard GMM, as shown in Figure 5.8(b). This result is used as the initialization step for all the remaining methods. SVFMM and SIMF methods in Figure 5.8(c) and Figure 5.8(d) can produce a better segmentation, even though the effect of noise remains quite high. Compared with these methods, we find that the proposed method, shown in Figure 5.8(f) successfully segment all objects, and the effect of noise on the final segmented image is quite low.

5.3.3 Segmentation of Colored Images

In Figure 5.9, we show the segmentation results of a real-world color image. The orginal image, as shown in Figure 5.9(a), was used for segmentation into
three labels: “snow”, “sky”, and “others”. The initialization for all algorithms was carried out by using the standard GMM as shown in Figure 5.9(b). Extraction accuracies of the SVFMM, CA-SVFMM, SIMF and MEANF methods are shown in Figure 5.9(c) to Figure 5.9(f). As shown in Figure 5.9, the segmentation accuracy for SIMF and MEANF methods is quite poor. CA-SVFMM can produce a better segmentation; however, the edge between the “snow” and the “sky” is lost. A closer inspection of the “sky” area indicates that a small portion of pixels have been misclassified. The proposed method in Figure 5.9(g), can better classify with more detail along the sharp edge between “snow” and the “sky”, as compared with the CA-SVFMM method. Figure 5.9(h) shows the maximization progress of the log-likelihood of the proposed method in this experiment.

In order to further test the accuracy of the proposed algorithm, another real word color image is used. The image shown in Figure 5.10(b) is obtained by corrupting the original image in Figure 5.10(a) with Gaussian noise (0 mean, 0.0015 variance). The objective is to segment the image into three la-
Figure 5.9: Color image segmentation (310007), (a): original image, (b): Standard GMM, (c): SVFMM, (d): CA–SVFMM, (e): SIMF, (f) MEANF, (g): Proposed method, (h): Maximization progress of the log-likelihood of the proposed method.

Figure 5.10(c) presents the segmentation results obtained by employing the standard GMM method. The segmented image is used to initialize the remaining methods. A visual inspection of the results indicate that SVFMM in Figure 5.10(d), CA-SVFMM in Figure 5.10(e), ICM in Figure 5.10(f), and MEANF in Figure 5.10(g) are unable to segment the image successfully. The effect of noise on the final segmented image is highly noticeable. These methods lose much of the image sharpness and salient details in and around the “hands” and “legs” region of the girl in the image. Compared with these algorithms, the proposed method leads to a smoother segmentation. The effect of noise on the final segmented image, obtained by employing the proposed method is far less.

A set of real world color images are used to evaluate the performance of the proposed method against SVFMM, CA-SVFMM, ICM, SIMF, MEANF
methods. Table 5.4 contains the cumulative results obtained for all methods, for the given set of real world images. As can be easily seen, on average, the proposed method outperforms other methods with a higher PR. Figure 5.6 shows some of the other real-world images used for segmentation by employing SVFMM, MEANF and the proposed method, respectively. The first row shows the original images, followed by the corresponding segmented images in the second, third, and the last row. Figure 5.11 clearly indicates that our proposed method achieves a better segmentation accuracy.
Table 5.4: Comparison of image segmentation results on Berkeley’s color image segmentation dataset: Probabilistic Rand (PR) Index.

<table>
<thead>
<tr>
<th>Image</th>
<th>labels</th>
<th>SVFMM</th>
<th>CA-SVFMM</th>
<th>ICM</th>
<th>SIMF</th>
<th>MEANF</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>86000</td>
<td>4</td>
<td>0.811</td>
<td>0.814</td>
<td>0.794</td>
<td>0.773</td>
<td>0.785</td>
<td>0.825</td>
</tr>
<tr>
<td>101085</td>
<td>2</td>
<td>0.598</td>
<td>0.600</td>
<td>0.572</td>
<td>0.566</td>
<td>0.569</td>
<td>0.603</td>
</tr>
<tr>
<td>100080</td>
<td>5</td>
<td>0.802</td>
<td>0.801</td>
<td>0.778</td>
<td>0.768</td>
<td>0.767</td>
<td>0.811</td>
</tr>
<tr>
<td>105053</td>
<td>2</td>
<td>0.778</td>
<td>0.782</td>
<td>0.756</td>
<td>0.650</td>
<td>0.689</td>
<td>0.817</td>
</tr>
<tr>
<td>241004</td>
<td>5</td>
<td>0.839</td>
<td>0.841</td>
<td>0.838</td>
<td>0.841</td>
<td>0.842</td>
<td>0.841</td>
</tr>
<tr>
<td>175043</td>
<td>2</td>
<td>0.800</td>
<td>0.801</td>
<td>0.798</td>
<td>0.713</td>
<td>0.722</td>
<td>0.802</td>
</tr>
<tr>
<td>374067</td>
<td>4</td>
<td>0.815</td>
<td>0.814</td>
<td>0.801</td>
<td>0.781</td>
<td>0.783</td>
<td>0.818</td>
</tr>
<tr>
<td>24063</td>
<td>3</td>
<td>0.840</td>
<td>0.839</td>
<td>0.838</td>
<td>0.839</td>
<td>0.839</td>
<td>0.841</td>
</tr>
<tr>
<td>106025</td>
<td>4</td>
<td>0.832</td>
<td>0.833</td>
<td>0.824</td>
<td>0.796</td>
<td>0.803</td>
<td>0.844</td>
</tr>
<tr>
<td>147091</td>
<td>3</td>
<td>0.813</td>
<td>0.815</td>
<td>0.770</td>
<td>0.773</td>
<td>0.775</td>
<td>0.824</td>
</tr>
<tr>
<td>277095</td>
<td>3</td>
<td>0.811</td>
<td>0.812</td>
<td>0.773</td>
<td>0.637</td>
<td>0.690</td>
<td>0.814</td>
</tr>
<tr>
<td>113009</td>
<td>3</td>
<td>0.671</td>
<td>0.670</td>
<td>0.645</td>
<td>0.614</td>
<td>0.622</td>
<td>0.676</td>
</tr>
<tr>
<td>296007</td>
<td>3</td>
<td>0.839</td>
<td>0.840</td>
<td>0.841</td>
<td>0.832</td>
<td>0.841</td>
<td>0.841</td>
</tr>
<tr>
<td>189080</td>
<td>4</td>
<td>0.861</td>
<td>0.862</td>
<td>0.814</td>
<td>0.805</td>
<td>0.807</td>
<td>0.867</td>
</tr>
<tr>
<td>38092</td>
<td>4</td>
<td>0.871</td>
<td>0.873</td>
<td>0.820</td>
<td>0.820</td>
<td>0.817</td>
<td>0.879</td>
</tr>
<tr>
<td>41004</td>
<td>3</td>
<td>0.908</td>
<td>0.909</td>
<td>0.891</td>
<td>0.848</td>
<td>0.872</td>
<td>0.917</td>
</tr>
<tr>
<td>Mean</td>
<td>-</td>
<td>0.806</td>
<td>0.807</td>
<td>0.785</td>
<td>0.754</td>
<td>0.764</td>
<td>0.814</td>
</tr>
</tbody>
</table>

5.4 Conclusions

We have presented a new mixture model for image segmentation that incorporates the spatial relationships based on MRF. Compared with other MRF based mixture models, our proposed method directly applies the EM algorithm to optimize the parameters, making it simple, fast, and easy to implement. The proposed method has been tested with many synthetic and real world grayscale and colored images, thereby demonstrating the excellent performance in noisy conditions, as compared to other mixture model based approaches.
Figure 5.11: (first row): original image, (second row): SVFMM, (third row): MEANF, (last row): Proposed method.
Chapter 6

Conclusions

6.1 Summary of Main Contributions

In this thesis, new fully unsupervised image segmentation algorithms based on mixture model with spatial neighborhood relationships have been described. The proposed methods have been tested with both synthetic and real images, and have been shown to be robust with respect to noise, efficient with respect to the number of parameters used, and sufficiently accurate with respect to the classification results.

A review of the various model-based techniques used to achieve segmentation is described. In Chapter 2, the first group of model-based techniques is described, beginning with the using of standard GMM to solve the unsupervised segmentation problem. The advantages and disadvantages of the standard GMM are then discussed. In order to estimate the parameters of the model, various techniques on maximizing their likelihood are described,
beginning with the EM algorithm, then continuing with the gradient-based optimization techniques.

The second group of model-based techniques for unsupervised segmentation is described in Chapter 3. These methods are based on the MRF to take into account the spatial correlation between the neighboring pixels and to reduce the sensitivity of the segmentation result with respect to noise. In this chapter, two types of models based on MRF are presented. In the first type, mixture models with MRF have been employed for pixel labels. In order to take into account the spatial correlation, an MRF model in the second type is used to model the joint distribution of the priors of each pixel label, instead of the joint distribution of the pixel labels as in first type. The main differences between the first and the second group of model-based techniques are also discussed in this chapter.

In Chapter 4, we propose an extension of the standard GMM for image segmentation, which utilizes a novel approach to incorporate the spatial relationships between neighboring pixels into the standard GMM. The proposed model is easy to implement and compared with the existing MRF models, requires a fewer parameters. We also propose a new method to estimate the model parameters in order to minimize the higher bound on the data negative log-likelihood, based on the gradient method. Experimental results obtained on noisy synthetic and real world grayscale images demonstrate the robustness, accuracy and effectiveness of the proposed model in image segmentation, as compared to other methods based on standard GMM and MRF models.
In Chapter 5, another way to incorporate spatial information between the neighboring pixels into the GMM based on MRF is proposed. In comparison to other mixture models that are complex and computationally expensive, the proposed method is robust and fast to implement. In mixture models based on MRF, the M-step of the EM algorithm cannot be directly applied to the prior distribution \( \pi_{ij} \) for maximization of the log-likelihood with respect to the corresponding parameters. Compared with these models, our proposed method directly applies the EM algorithm to optimize the parameters, which makes it much simpler. Finally, our approach is used to segment many synthetic and real-world grayscale and colored images with excellent results.

6.2 Future Directions

Several extensions are possible for future work. In this thesis, the number of labels \( K \) is currently set by the user based on prior knowledge. This is an open question and remains the subject of current research. One possible improvement on the current models is to investigate the ways to automatically optimize this parameter.

Another limitation is that a constant \( \beta \) is used throughout the image and for every label. A constant value of \( \beta \) reduces the impact of noise in homogeneous tissues but negatively affects segmentation along the border of two tissues. In the highly noisy image, it can erase the detail of the boundary of two tissues. One possible solution to overcome this problem is to use a different value of \( \beta \) throughout the image.
Another possible extension of this work is to study the use of non-linear smoothing filter to improve the quality of image segmentation. In the context of this thesis, one of the simplest linear filters is used. Our spatial process is based on the mean smoothing filter, where we replace each pixel value in an image with the average value of its neighbors.

Recently, to improve the robustness of the algorithm to outliers, finite Student’s-t mixture model (SMM) has been proposed. The main advantage of the Student’s-t distribution is that it is heavily tailed than Gaussian, and hence finite mixture model of the longertailed multivariate Student’s-t distribution provides a much more robust approach to the GMM. It has proven to be quite effective for image segmentation. However, in the existing SMM, the prior distribution does not depend on the pixel index and has the same value for all pixels. Moreover, it does not take into account the spatial constraints in an image. Additionally, in order to estimate the model parameters by adopting the EM algorithm, the Student’s-t distribution in the previous SMM is represented as an infinite mixture of scaled Gaussians, which corresponds to an increase in the algorithms’ complexity. A combination between finite Student’s-t mixture model with our methods to develop a new model could be considered in future work.

Segmentation of medical images such as computed tomography and magnetic resonance imaging is an important diagnostic imaging technique. Accurate segmentation of these images offers an opportunity to provide more information about the objects and their boundaries for clinical investigation.
However, fully automatic segmentation of medical images into a number of non-overlapping regions i.e., gray matter, white matter, cerebrospinal fluid, and background is difficult because they are frequently corrupted by high levels of noise, and poor contrast along boundaries. The proposed models could be applied for segmenting medical images in the future.
References


[21] Nakagawa Y., and Rosenfeld A., “Some experiments on variable thresh-

[22] Yanowitz S. D., and Bruckstein A. M., “A new method for image seg-


ministic network for texture segmentation,” *IEEE Transactions on Acous-


Appendix A

We apply the gradient descent technique to update the parameters $\Xi = \{\Theta, \Pi\} = (\mu_j, \sigma_j, c_j, b_j, \alpha), \ j = (1, 2, ..., K)$. The error function in Eq. (4.10) can be re-written as:

$$E(\Theta^{(t)}, \Pi^{(t)}|\Theta^{(t+1)}, \Pi^{(t+1)}) = - \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \{ \log \pi_{ij} + \log \Phi(x_i|\Theta_j) \}$$  \hspace{1cm} (A.1)

where

$$\pi_{ij} = \left[ \sum_{m \in N_i} \exp \left( - \frac{(x_m - c_j)^2}{2b_j^2} \right) \right]^\alpha \left/ \sum_{k=1}^{K} \left[ \sum_{m \in N_i} \exp \left( - \frac{(x_m - c_k)^2}{2b_k^2} \right) \right]^\alpha \right.$$  

and

$$\Phi(x_i|\Theta_j) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp \left( - \frac{(x_i - \mu_j)^2}{2\sigma_j^2} \right)$$

as shown in Eq.(4.3) and Eq.(2.23), respectively. The derivative of $E$ with respect to $\mu_j$ is given by:

$$\frac{\partial E}{\partial \mu_j} = - \sum_{i=1}^{N} z_{ij}^{(t)} \frac{x_i - \mu_j}{\sigma_j^2}$$  \hspace{1cm} (A.2)

and similarly, the derivative of $E$ with respect to $\sigma_j$, is given by:

$$\frac{\partial E}{\partial \sigma_j} = - \sum_{i=1}^{N} z_{ij}^{(t)} \left( - \frac{1}{\sigma_j} + \frac{(x_i - \mu_j)^2}{\sigma_j^3} \right)$$  \hspace{1cm} (A.3)
The derivative of $E$ with respect to $c_j$ can be expressed as:

$$
\frac{\partial E}{\partial c_j} = - \sum_{i=1}^{N} z_{ij}^{(t)} \left\{ \frac{\alpha \sum_{m \in \mathcal{N}_i} \frac{(x_m - c_j)}{b_j^2} \exp \left( - \frac{(x_m - c_j)^2}{2b_j^2} \right)}{\sum_{m \in \mathcal{N}_i} \exp \left( - \frac{(x_m - c_j)^2}{2b_j^2} \right)} \right\} 
$$

$$
+ \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik}^{(t)} \left\{ \frac{\alpha \left[ \sum_{m \in \mathcal{N}_i} \exp \left( - \frac{(x_m - c_k)^2}{2b_k^2} \right) \right]^{\alpha-1} \sum_{m \in \mathcal{N}_i} \frac{(x_m - c_j)}{b_j^2} \exp \left( - \frac{(x_m - c_j)^2}{2b_j^2} \right)}{\sum_{l=1}^{K} \sum_{m \in \mathcal{N}_i} \exp \left( - \frac{(x_m - c_l)^2}{2b_l^2} \right) \alpha} \right\}
$$

(A.4)

Now, considering the derivative of the term $E$ with respect to $b_j$, we have:

$$
\frac{\partial E}{\partial b_j} = - \sum_{i=1}^{N} z_{ij}^{(t)} \left\{ \frac{\alpha \sum_{m \in \mathcal{N}_i} \frac{(x_m - c_j)^2}{b_j^2} \exp \left( - \frac{(x_m - c_j)^2}{2b_j^2} \right)}{\sum_{m \in \mathcal{N}_i} \exp \left( - \frac{(x_m - c_j)^2}{2b_j^2} \right)} \right\} 
$$

$$
+ \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik}^{(t)} \left\{ \frac{\alpha \left[ \sum_{m \in \mathcal{N}_i} \exp \left( - \frac{(x_m - c_k)^2}{2b_k^2} \right) \right]^{\alpha-1} \sum_{m \in \mathcal{N}_i} \frac{(x_m - c_j)^2}{b_j^2} \exp \left( - \frac{(x_m - c_j)^2}{2b_j^2} \right)}{\sum_{l=1}^{K} \sum_{m \in \mathcal{N}_i} \exp \left( - \frac{(x_m - c_l)^2}{2b_l^2} \right) \alpha} \right\}
$$

(A.5)

The derivative of $E$ with respect to $\alpha$ is given by:

$$
\frac{\partial E}{\partial \alpha} = - \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \log \left( \sum_{m \in \mathcal{N}_i} \exp \left( - \frac{(x_m - c_j)^2}{2b_j^2} \right) \right) 
$$

$$
+ \sum_{i=1}^{N} \sum_{k=1}^{K} z_{ik}^{(t)} \left\{ \frac{K}{\sum_{l=1}^{K} \sum_{m \in \mathcal{N}_i} \exp \left( - \frac{(x_m - c_k)^2}{2b_k^2} \right) \alpha} \log \left( \sum_{m \in \mathcal{N}_i} \exp \left( - \frac{(x_m - c_j)^2}{2b_j^2} \right) \right) \right\}
$$

(A.6)
Publications


Vita Auctoris

Thanh Minh Nguyen was born in 1981 in Vietnam. He received the B.E. degree in electrical and electronic engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in April 2004, and the MA.Sc. degree from the Department of Electrical Engineering, Dayeh University, Changhua, Taiwan, in June 2006. He received a Doctor of Philosophy degree from the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada, in August 2011. His current research interests include computer vision and pattern recognition.