Models for images and video foreground segmentation using finite mixtures of generalized Gaussians

Aissa Boulmerka

Ecole nationale Supérieure d'Informatique

THESE

En vue de l'obtention du Grade de

DOCTEUR en INFORMATIQUE

Présentée et soutenue le 23/05/2016 par

BOULMERKA Aïssa

Dirigée par Prof. AIT-AOUDIA Samy (ESI, Algérie), et
Co-dirigée par Prof. ALLILI Mohand Saïd (UQO, Canada)

Titre :

Models for images and video foreground segmentation using finite mixtures of generalized Gaussians

JURY

Président	M. HIDOUCI Walid Khaled	Professeur, ESI, Algérie
Examinateurs	M. DJEDI Nourredine	Professeur, U. Biskra, Algérie
	M. MOUSSAOUI Abdelouahab	Professeur, U. Sétif, Algérie
	M. LARABI Slimane	Professeur, USTHB, Algérie
	M. NACEREDDINE Nafaa	Professeur, CRTI Ex CSC, Algérie
Directeur	M. AIT-AOUDIA Samy	Professeur, ESI, Algérie

Acknowledgements

I would like to express my deepest gratitude to my supervisor Prof. Samy Ait-Aoudia at the École Nationale Supèrieure d'Informatique (ESI), for believing in me and giving me this opportunity, for his support, and for his wise advice.

I also want to thank my co-supervisor Prof. Mohand-Saïd Allili from the Université du Québec en Outaouais (UQO), Canada, for his support, ideas, and encouraging. I am very grateful to him for inviting and co-supervising me during my visits to his lab LARIVIA (LAboratoire de Recherche en Imagerie et en VIsion Artificielle), UQO, during two months (March 2012 and November 2012) and 18 months (from October 2013 to March 2015).

I would address my gratitude to the members of my examination committee who accepted to examine this thesis, Prof. HIDOUCI Walid Khaled (ESI), Prof. DJEDI Nourredine (U. Biskra), Prof. MOUSSAOUI Abdelouahab (U. Setif), M. LARABI Slimane and Prof. NACEREDDINE Nafaa (CRTI Ex CSC).

This work would have not been possible without the funding and support provided by several institutions, starting at the regional level with the University Center of Mila, national with the Ministry of Higher Education and Scientific Research along with the École Nationale Supèrieure d'Informatique (ESI) and going international with the LARIVIA lab at the Université du Québec en Outaouais (UQO), Canada.

Special thanks to my colleagues during my work at LARIVIA lab for making such a great experience. Especially, I would like to thank Ouiza Ouyed, Marouene Mejri, Daniel Yapi, Idir Filali, and Siham Bacha for their help and fruitful discussions.

Finally, I would like to thank my friends, who have followed my work and offer me both support and encouragement. Special thanks to my family, to my mother who has given me love and support all the days of my life and who is always there for me. Thanks to my wife for her understanding and patience during all the years of this work.

Abstract

In this thesis, we deal with the *foreground segmentation* (FS) problem that is a key issue to numerous computer vision applications such as document analysis, object recognition, and video surveillance. Two problems can arise from the foreground segmentation depending on the input data: *image foreground segmentation* vs. *video foreground segmentation*. Many approaches have been proposed to address both of these problems, among of them is the histogram-based approach. However, most of the histogram-based approaches assume a unimodal Gaussian histogram shape for data classes and make use of parameters to estimate their distributions. Consequently, the efficiency these techniques could be affected by the fact that the data distribution is multi-modal and/or non-Gaussian. Moreover, addressing the problem of video foreground segmentation is not a trivial task especially in some challenging situations such as illumination changes, cast shadows, dynamic backgrounds, and pan-tilt-zoom (PTZ). These challenges have been widely studied to be able to make the current approaches more robust to such scenarios.

To address the FS in images and videos, we use the mixture of generalized Gaussians (MoGG's) for modeling the histogram of data (image or video). The merits of using the MoGG include: (i) An additional degree of freedom that controls its kurtosis. (ii) Histogram modes, ranging from sharply peaked to flat ones, can be accurately represented using this model. (iii) Skewed and multi-modal classes are explicitly represented using mixtures of GGDs. Furthermore, an online mixture of generalized Gaussian (MoGG) model is used to model the temporal information represented by the pixel history in image sequences. The former model is enriched by integrating temporal co-occurrence of background/foreground classes to deal with complex background dynamics. Besides, spatial analysis is introduced to deal with shadows, stopping objects, and PTZ camera effects.

The proposed algorithms have been developed to run in real-world environments with near real-time performance. Experiments on the available datasets show that the proposed algorithms significantly enhance results (both qualitatively and quantitatively) compared to the other state-of-the-art techniques.

Keywords: Foreground segmentation, Multi-class thresholding, Mixture of Generalized Gaussian Distributions (MoGG), Background subtraction, Temporal information, Spatial information.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

The following acronyms and abbreviations are used in this work:

AIC	Akaike Information Criterion
BS	Background Subtraction
EM	Expectation Maximisation
ER	misclassification Error Rate
FMM	Finite Mixture Model
FS	Foreground Segmentation
fps	Frames Per Second
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
ICM	Iterated Conditional Modes
KDE	Kernel Density Estimate
LBP	Local Binary Pattern
MAP	Maximum a posteriori
MET	Kittler and Illingworths Minimum Error Thresholding
MLE	Maximum Likelihood Estimate
MoGG	Mixture of Generalized Gaussians
MRF	Markov Random Field

NDT	Non Destructive Testing
PCA	Principal Component Analysis
PDF	Probability Density Function
PTZ	Pan-Tilt-Zoom
ROC	Receiver Operating Characteristic
ROI	Region of Interest
STLBP	Spatio-Temporal Local Binary Pattern
SVM	Support Vector Machine

1 Introduction

Computer vision is a branch of computer science dedicated to developing algorithms for processing visual information that can be obtained using various techniques such as digital cameras, infra-red, ultrasound, X-ray, and microscopes. The computer vision field has rapidly progressed thanks to the processing power, memory, and storage capacity of computers that has drastically increased. Another reason for the recent advances in the computer vision area of research is the increased aid given by diverse disciplines including statistics, machine learning, neuroscience, mathematics, physics, physiology, and biology.

Computer vision is used nowadays in a wide spectrum of applications, such as medical imaging, optical character recognition (OCR), industrial automation, video surveillance, gesture recognition, object recognition, and automotive driver assistance (see David Lowe's Web site of industrial vision applications and products related to computer vision [78].)

From a historical point of view, the list in Figure 1.1 shows popular computer vision areas. The future of computer vision is inspiring. Our understanding increases by the day, and it is likely that vision products will become more frequent in the next decade. As noted by Szeliski [127]: "It may be many years before computers can name and outline all of the objects in a photograph with the same skill as a two-year-old child.".

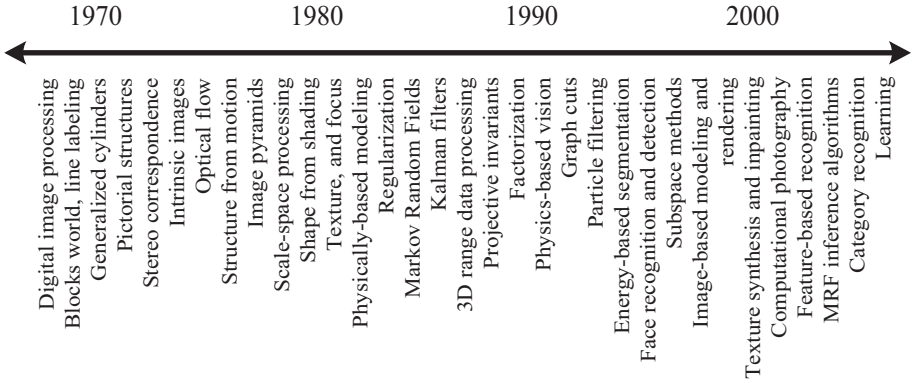


FIGURE 1.1: A chronological listing of popular topics of research in the field of computer vision [127].

1.1 Foreground segmentation (FS)

Foreground segmentation is a fundamental and critical task in many computer vision applications such as object recognition in images and action in videos and has been extensively studied for several decades. Foreground segmentation can be formulated as an *unsupervised binary classification* problem that classifies the pixels of the input images (videos) into foreground objects or background area. Several approaches have been proposed to deal with this problem. One of the most simple and efficient approaches is based on machine learning and statistical modeling. In such an approach, *generative models* are often used to find the probability distribution expressed as a parametric model, and then uses this distribution to make predictions for the input image (video) pixels. For example, data can be modeled as a finite mixture of distributions such as the *Gaussian mixture model (GMM)* where the finite mixture parameters can be estimated using the expectation-maximization (EM) algorithm [31]. The components of the resulting mixture model are then evaluated to determine which pixels are most likely to be labeled as foreground/background.

Two types of visual input data can be distinguished: (i) *still images* and (ii) *video sequences* which are images that change with time. In still images applications, foreground segmentation consists of dividing the image into two regions (classes), namely the foreground and the background region. This is a primary task in many applications, including (i) object recognition [11, 156], (ii) defect detection [94], and document analysis and recognition [116] in images. Regarding the video case, foreground segmentation helps to extract moving objects from the static background. This is an integral part of many video processing applications

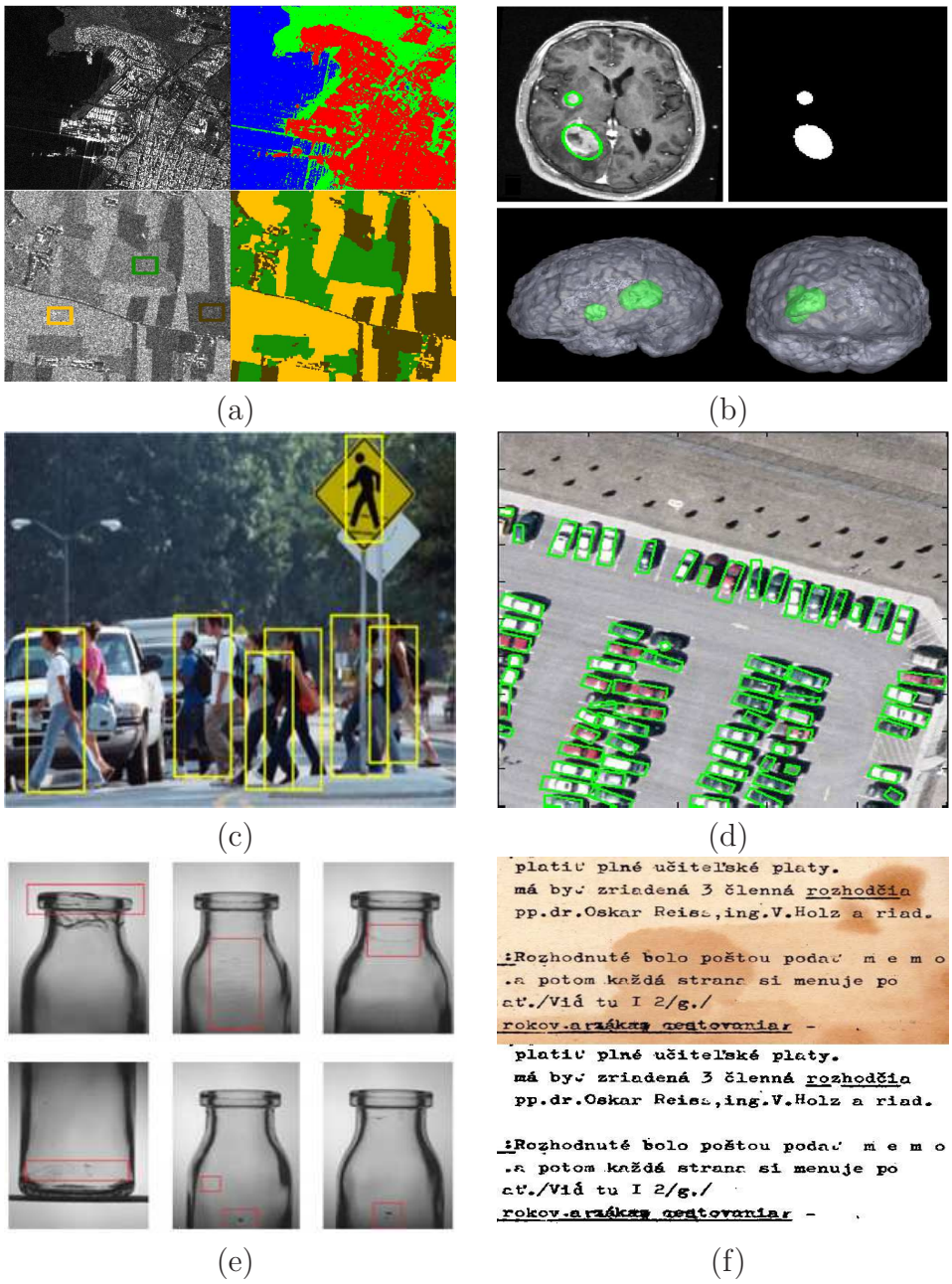


FIGURE 1.2: Some popular image segmentation applications. (a) SAR images for remote sensing. (b) MRI brain tumor detection and segmentation. (c) Pedestrian detection. (d) Object detection (cars). (e) Defect detection for glass bottles. (f) Document analysis and recognition.

such as (i) video surveillance [18], (ii) anomaly detection [146], and (iii) scene analysis and recognition [28]. Figure 1.2 shows some applications of still image and video foreground segmentation. In what follows, we present an overview of these applications.

1.1.1 Foreground segmentation in still images

Several approaches have been proposed to perform foreground segmentation of images also known as *image (histogram) thresholding*. Histogram thresholding is the most widely used method to deal with the image foreground segmentation. This approach is based on analysis of the histogram of a gray level image, searching for an optimal threshold value that divides the histogram into two classes (the foreground class $\mathcal{C}_1$ and background class $\mathcal{C}_2$). For instance, the standard Otsu's method [97], minimum-error thresholding (MET) method [66], and Xue and Titterington [148] method are based on fitting a finite mixture model. An extensive survey of histogram thresholding methods is given by Sezgin & Sankur [118]. The MET method is ranked among the best in the study conducted by [118]. The Otsu's method is implemented by many image processing tools such as OpenCV [1] and MATLAB Image Processing Toolbox [2].

It can be noted, however, that the above methods assume a uni-modal histogram shape for each class which is fitted using a Gaussian distribution. Consequently, the multi-modality aspect that characterizes the background and/or foreground distributions in real-world images has not been considered adequately. Another limitation of the standard histogram-based approach lies in the assumption that class data is Gaussian. In real-world images, one can find different shapes of histogram modes such as skewed and heavy-tailed ones, making the assumption of Gaussian-distributed classes not realistic.

1.1.2 Foreground segmentation in video sequences

As mentioned above, foreground segmentation is a crucial task in the field of video processing as it can significantly influence the performance of video applications such as detecting moving people in a video surveillance system. Among methods used for foreground segmentation of videos is by modeling the background also known as *background subtraction (BS)*. BS aims to separate the moving foreground objects from the static background by using an adaptive background model. A background subtraction process can be performed in two stages. First, the reference model of the background is established using an initial set of training frames

[1] http://www.opencv.org
[2] http://www.mathworks.com

without moving objects (i.e. frames that contain the background only). Then, each new frame is compared with the reference model to determine pixels representing the foreground objects and those representing the background.

The approaches for BS in the literature can be classified into one of two categories *parametric statistical models* such as Gaussian mixture models (GMMs) [3, 61, 120, 125] and *non-parametric statistical models* such as kernel density estimation (KDE) [8, 33, 52, 96]. The main idea behind these methods is to model statistically the background through pixel history and then using prediction to detect moving objects through probability estimation. However, addressing background subtraction is not a trivial task particularly in some situations and uncontrolled environments such as airports, museums, motorways, and outdoors environments. The captured videos in such environments usually contain varying illumination, image noise, cast shadows, dynamic backgrounds, and camera jitter., etc. This makes the BS methods generating plenty of false positives. These challenges have been extensively studied by the video processing community to make BS more robust. Indeed, most of the proposed algorithms are dedicated to dealing with one or two challenges but give poor performance for other challenges [22]. Finally, building effective systems for such environments (e.g. airports, and outdoors) is still a tremendous challenge because no method can deal with all challenges in an efficient way such as the human visual system (HVS).

1.2 Contributions

In the light of the above discussion, foreground segmentation (FS) poses a significant challenge for computer vision applications and developing a universal method for FS is still elusive. However, one can develop methods that can deal with as many challenges as possible and be robust with noise and outliers. Inspired by the recent development of machine learning research, we propose to deal with FS using new statistical methods based on parametric distributions. The key goal of this thesis is to address foreground segmentation using new statistical thechniques. As mentioned above, two problems can be raised regarding the input data: (1) the foreground segmentation of still images, and (2) the foreground segmentation of video sequences.

The first contribution of this work involves addressing the problem of image segmentation by proposing a model that deals with multi-modal classes with arbitrarily shaped modes. The state-of-art techniques of image segmentation based on using single probability density functions (pdf's) are generalized to mixtures of generalized Gaussian distributions (MoGG's). It is well known that the generalized Gaussian distribution (GGD) is more robust to fit the model than the simple Gaussian distribution. Especially, when the shape of the underlying distribution

is skew or heavy-tailed or contaminated by outliers. The merits of employing the mixtures of generalized Gaussian include: (i) an additional degree of freedom that controls its kurtosis. (ii) histogram modes, ranging from sharply peaked to flat ones, can be accurately represented using this model. (iii) skewed and multi-modal classes are accurately represented using mixtures of GGDs.

In the second contribution, we propose a new approach to overcoming some background modeling challenges and uncontrolled situations in video foreground segmentation. The background subtraction approach is adopted due to its adaptability and computational efficiency compared to the other existing methods. The proposed foreground segmentation approach is capable of dealing with image sequences containing several challenges such as background dynamics, shadows and illumination variations, and pan-tilt-zoom (PTZ) camera effects. The proposed approach combines temporal and spatial information to perform pixel-level background subtraction. An online mixture of generalized Gaussian (MoGG) model is employed to model the temporal information of each pixel location. To cope with complex background dynamics, the MoGG model is enhanced by introducing background/foreground co-occurrence analysis. Spatial information is introduced through multi-scale inter-frame correlation analysis and histogram matching to dissociate changes due to shadows and illumination changes, along with frame displacement estimation to deal with PTZ camera motion effects.

1.3 Outline of the thesis

This thesis includes a general introduction and four Chapters. Chapter 2 provides background theory on foreground segmentation, related works in this area of research as well as the datasets used in this work. Chapter 3 covers a new approach to multi-class image histogram segmentation based on the generalized mixture of Gaussian distributions, along with some experimentations conducted on synthetic data and real-world image segmentation. Chapter 4 is dedicated to a proposed approach based on the mixture of generalized Gaussian distributions and combining temporal and spatial information to perform efficient foreground segmentation of video data. Finally, Chapter 5 summarizes our concluding remarks and provides a list of possible future directions. These four chapters are summarized as follows:

- **Chapter 2: Background Theory and Related Works.** This chapter provides the background theory of finite mixture modeling, including (1) the Gaussian Mixture Model (GMM), (2) the Mixture of Generalized Gaussian (MoGG) Model, and (3) the Expectation-Maximisation (EM) algorithm and its variants, which are in turn employed to estimate the mixture parameters. Related works to image segmentation (particularly image thresholding) are

also introduced. Finally, the state-of-the-art of the video foreground segmentation approaches is introduced, followed by the foreground segmentation evaluation and existing datasets.

- **Chapter 3: A Generalized Multiclass Histogram Thresholding Approach Based on Mixture Modelling.** This chapter presents a new approach to multi-class thresholding-based segmentation. It considerably improves existing thresholding methods by efficiently modeling non-Gaussian and multi-modal class-conditional distributions using mixtures of generalized Gaussian distributions (MoGG). The proposed approach seamlessly: (1) extends the standard Otsu's method to arbitrary numbers of thresholds and (2) extends the Kittler and Illingworth minimum error thresholding to non-Gaussian and multi-modal class-conditional data. MoGGs enable efficient representation of heavy-tailed data and multi-modal histograms with flat or sharply shaped peaks. Experiments on synthetic data and real-world image foreground segmentation show the performance of the proposed approach with a comparison to recent state-of-the-art techniques.

- **Chapter 4: Foreground Segmentation in Videos Combining General Gaussian Mixture Modeling and Spatial Information.** This chapter presents a new statistical approach combining temporal and spatial information for robust background subtraction (BS) in videos. Temporal information is modeled by coupling finite mixtures of Generalized Gaussian (MoGG) distributions with foreground/background co-occurrence analysis. Spatial information is modeled by combining multi-scale inter-frame correlation analysis and histogram matching. We propose an algorithm that efficiently combines both information to cope with several BS challenges, such as cast shadows, illumination changes, and various complex background dynamics. In addition, global video information is used through a novel technique to deal with pan-tilt-zoom (PTZ) camera effects. Experiments with comparison with recent state-of-the-art methods have been conducted on standard datasets. Obtained results have shown that our approach outperforms several recent state-of-the-art methods on the aforementioned challenges while maintaining comparable computational time.

- **Chapter 5: Conclusion and Future Directions.** This chapter summarizes the contributions of this dissertation and suggests new avenues and improvements for future research.

2

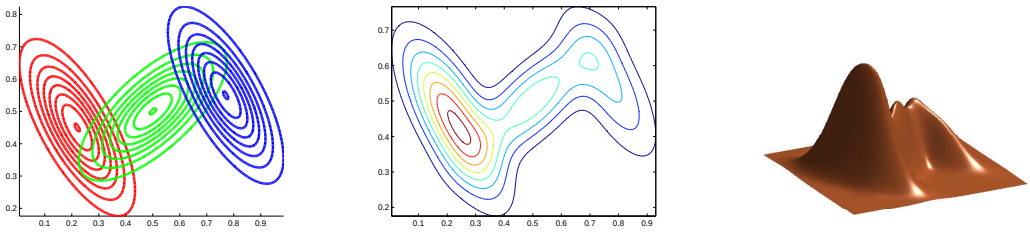
Background Theory and Related Works

2.1 Overview

In this Chapter, we provide an overview of the relevant theory used in this thesis along with the related works in foreground segmentation. Firstly, we describe the Mixture of Generalized Gaussian (MoGG) model, which is employed in the statistical modeling of the proposed algorithms, followed by the Expectation-Maximization (EM) algorithm which is employed to estimate the MoGG parameters. Secondly, we introduce the related works to still-image foreground segmentation. Finally, the related works in video foreground segmentation are discussed.

2.2 Finite mixture models and EM algorithm

Finite mixture models (FMM) provides a mathematical framework to the statistical modeling of a various random phenomena [86]. This modeling approach has been successfully applied in several fields including computer vision, pattern recognition, machine learning, and statistical analysis to name a few. In many applications, their parameters are estimated by maximum likelihood, typically using the Expectation-Maximization (EM) algorithm [13]. To understand the role of finite mixture models (FMM) and Expectation-Maximization (EM) algorithm in image and video foreground segmentation methods, it pays to examine some preliminary issues.



(a) (b) (b)

FIGURE 2.1: A mixture of three Gaussian distributions in a two-dimensional space. (a) Contours of the probability for each component in the mixture. (b) Contours of the mixture probability density function f. (c) A surface plot of the mixture density. Based on [13].

2.2.1 Definitions

Let $X = \{X_1, ..., X_N\}$ be a random sample of size N, where each point X_i is a d-dimensional vector ($X_i \in \mathbb{R}^d$) generated by the probability density function (*pdf*) f on $\mathbb{R}^d$ space and let $x = (x_1, ..., x_N)$ be the realization of the random vector X, where x_i is the observed value of the random vector X_i. We suppose that the *pdf* $f(X_i)$ of X_i can be written in the form

$$f(x_i|\Theta) = \sum_{j=1}^{K} \omega_j f_j(x_i|\theta_j), \tag{2.1}$$

Where θ_j represents the set of parameters of each density function f_j and Θ is a vector containing all the unknown parameters governing this mixture model can be written as

$$\Theta = (\omega_1, ..., \omega_K, \theta_1, ..., \theta_K), \tag{2.2}$$

And where here ω_j are nonnegative quantities that sum to one

$$0 < \omega_j \leq 1 (j = 1, ..., K), \tag{2.3}$$

and

$$\sum_{j=1}^{K} \omega_j = 1, \tag{2.4}$$

The quantities $\omega_1, ..., \omega_K$ are called the mixing proportions or weights. The $f_j(x_i)$ density is known as the component density of the mixture. The function $f(x_i|\Theta)$ is referred as a K-component finite mixture distribution or simply a mixture distribution. The number of components K can be known or unknown according to the application. If K is unknown, it has to be inferred from the available data, along with the weights parameters and the specific parameters of the *pdf* f_j. For example, Figure 2.1 illustrates a mixture model composed of three Gaussian distributions, where the random variables are considered to be in a two-dimensional space.

In [81], the authors demonstrate that the family of the mixture of Gaussian distributions is very flexible and large one. In fact, they gave fifteen examples of univariate mixtures of Gaussian densities, corresponding to various combinations of the components, as listed in Table 2.1. The reproduced plots in Figure 2.2 show a wide variety of density shapes based on the model of the mixture of Gaussian distributions.

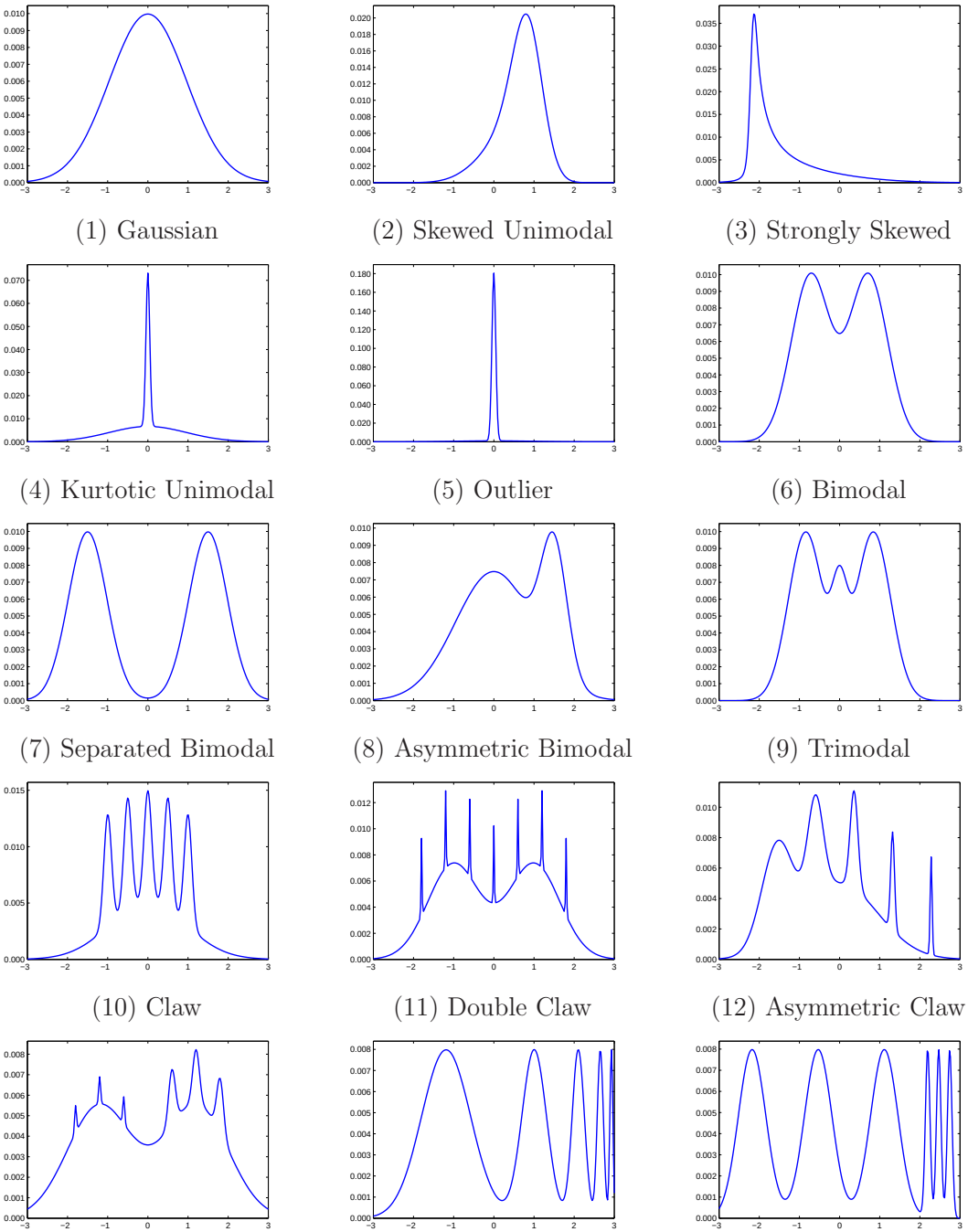
2.2.2 Expectation-Maximization (EM) algorithm

The expectation-maximization (EM) algorithm was introduced by Dempster *et al.* [31] for general latent variable models, many applications to finite mixture models were also mentioned. The use of the EM algorithm for the estimation of mixture models has been studied in detail by Redner and Walker [108]. In addition, a very inspiring general level tutorial on the EM algorithm in the context of finite mixtures of Poisson distributions is provided in [88], whereas McLachlan *et al.* [86] gives full details for a broad range of finite mixture models.

The log-likelihood function for a finite mixture model is defined as following

$$\log L(\Theta) = \log\Big\{ \prod_{i=1}^{N} f(x_i|\Theta) \Big\} = \sum_{i=1}^{N} \log\Big\{ \sum_{j=1}^{K} \omega_j f_j(x_i|\theta_j) \Big\}, \tag{2.5}$$

To implement the EM algorithm, we consider the problem of maximizing the likelihood for the complete-data set $Y = \{X, Z\}$, where $Z = \{Z_1, ..., Z_n\}$ is a latent variable that corresponds to the missing data. In particular, $z_{i,j}$ is a binary coding variable defined as



(1) Gaussian (2) Skewed Unimodal (3) Strongly Skewed

(4) Kurtotic Unimodal (5) Outlier (6) Bimodal

(7) Separated Bimodal (8) Asymmetric Bimodal (9) Trimodal

(10) Claw (11) Double Claw (12) Asymmetric Claw

(13) Asm. Db. Claw (14) Smooth Comb (15) Discrete Comb

FIGURE 2.2: Plots of the fifteen mixture of Gaussian densities proposed in [81].

Density	f
1. Gaussian	$\eta(0,1)$
2. Skewed unimodal	$\frac{1}{5}\eta(0,1)+\frac{1}{5}\eta(\frac{1}{2},(\frac{2}{3})^2)+\frac{3}{5}\eta(\frac{13}{15},(\frac{5}{9})^2)$
3. Strongly skewed	$\sum_{i=0}^{7}\frac{1}{8}\eta(3\{(\frac{2}{3})^i-1\},(\frac{2}{3})^{2i})$
4. Kurtotic unimodal	$\frac{2}{3}\eta(0,1)+\frac{1}{3}\eta(0,(\frac{1}{10})^2)$
5. Outlier	$\frac{1}{10}\eta(0,1)+\frac{9}{10}\eta(0,(\frac{1}{10})^2)$
6. Bimodal	$\frac{1}{2}\eta(-1,(\frac{2}{3})^2)+\frac{1}{2}\eta(1,(\frac{2}{3})^2)$
7. Separated bimodal	$\frac{1}{2}\eta(-\frac{3}{2},(\frac{1}{2})^2)+\frac{1}{2}\eta(\frac{3}{2},(\frac{1}{2})^2)$
8. Skewed bimodal	$\frac{3}{4}\eta(0,1)+\frac{1}{4}\eta(\frac{3}{2},(\frac{1}{3})^2)$
9. Trimodal	$\frac{9}{20}\eta(-\frac{6}{5},(\frac{3}{5})^2)+\frac{9}{20}\eta(\frac{6}{5},(\frac{3}{5})^2)+\frac{1}{10}\eta(0,(\frac{1}{4})^2)$
10. Claw	$\frac{1}{2}\eta(0,1)+\sum_{i=0}^{4}\frac{1}{10}\eta(i/2-1,(\frac{1}{10})^2)$
11. Double claw	$\frac{49}{100}\eta(-1,(\frac{2}{3})^2)+\frac{49}{100}\eta(1,(\frac{2}{3})^2)+\sum_{i=0}^{6}\frac{1}{350}\eta((i-3)/2,(\frac{1}{100})^2)$
12. Asymmetric claw	$\frac{1}{2}\eta(0,1)+\sum_{i=-2}^{2}(2^{1-i}/31)\eta(i+\frac{1}{2},(2^{-i}/10)^2)$
13. Asymmetric double claw	$\sum_{i=0}^{1}\frac{46}{100}\eta(2i-1,(\frac{2}{3})^2)+\sum_{i=1}^{3}\frac{1}{300}\eta(-i/2,(\frac{1}{100})^2)$ $+\sum_{i=1}^{3}\frac{7}{300}\eta(i/2,(\frac{7}{100})^2)$
14. Smooth comb	$\sum_{i=0}^{5}(2^{5-i}/63)\eta((65-96(\frac{1}{2})^i)/21,(\frac{32}{63})^2)/2^{2i})$
15. Discrete comb	$\sum_{i=0}^{2}\frac{2}{7}\eta((12i-15)/7,(\frac{2}{7})^2)+\sum_{i=8}^{10}\frac{1}{21}\eta(2i/7,(\frac{1}{21})^2)$

TABLE 2.1: Parameters for fifteen examples of the mixture of Gaussian density where $\eta(\mu,\sigma)$ represents the Gaussian distribution with mean μ and variance σ^2.

$$Z_i=(z_{i,1},...,z_{i,K}) \text{ where } z_{i,j}=\begin{cases}1 & \text{if } x_i\in \text{ the component } C_j\\ 0 & \text{otherwise}\end{cases} \tag{2.6}$$

The complete-data likelihood function is defined as following

$$L_c(\Theta|Y)=\prod_{i=1}^{N}\prod_{j=1}^{K}{\omega_j}^{z_{i,j}}f_j(x_i|\theta_j)^{z_{i,j}}, \tag{2.7}$$

Taking the logarithm, we obtain the complete-data log likelihood given by

$$\log L_c(\Theta|Y) = \sum_{i=1}^{N}\sum_{j=1}^{K} z_{i,j}(\log\omega_j + \log f_j(x_i|\theta_j)), \tag{2.8}$$

Starting from $\hat{\Theta}^{(0)}$, the EM algorithm iterates between two steps: an E-step, where the conditional expectation of $\log(L_c(\Theta|Y))$ is computed, given the current data along with the current parameters, and an M-step in which parameters that maximize the expected complete-data log likelihood function, obtained from the E-step are determined. Under fairly mild regularity conditions, the EM algorithm converges to a local maximum of the mixture likelihood function [31]. For mixture models, the E-step leads for $m = 1$ to the following estimator of $z_{i,j}$

$$\hat{z}_{i,j}^{(m)} = \frac{\hat{\omega}_j^{(m-1)} f_j(x_i|\hat{\theta}_j^{(m-1)})}{\sum_{k=1}^{K} \hat{\omega}_k^{(m-1)} f_k(x_i|\hat{\theta}_k^{(m-1)})}, \tag{2.9}$$

and the M-step involves maximizing

$$\sum_{i=1}^{N}\sum_{j=1}^{K} \hat{z}_{i,j}^{(m)}(\log\hat{\omega}_j^{(m)} + \log f_j(x_i|\hat{\theta}_j^{(m)})), \tag{2.10}$$

with respect to all unknown parameters in $\Theta = (\omega_1, ..., \omega_K, \theta_1, ..., \theta_K)$, leading to a new estimate $\hat{\Theta}^{(m)}$. It is easy to verify that for an arbitrary mixture

$$\hat{\omega}_k^{(m)} = \frac{\sum_{i=1}^{N} \hat{z}_{i,k}^{(m)}}{N}, \tag{2.11}$$

However, the estimation of the component parameters θ_j, of course, depends on the distribution family underlying the mixture.

Figure 2.3 illustrates an example of the EM algorithm applied to a binary classification based on Gaussian mixture model where blue points come from class $\mathcal{C}_1$, red points from class $\mathcal{C}_2$ and ambiguous points appear purple.

In the following section, we present the detailed definition and parameter estimation for the Mixture of generalized Gaussians (MoGG) model as given by Allili *et al.* [1–3].

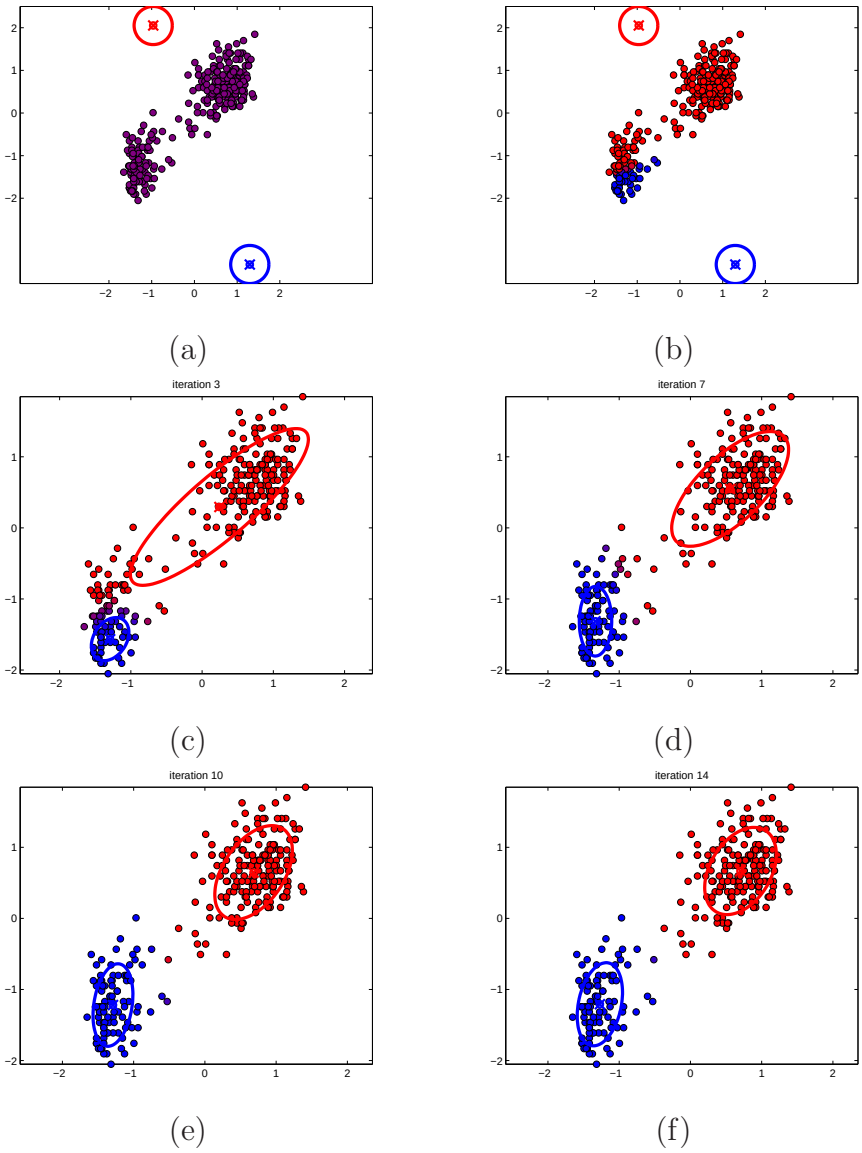
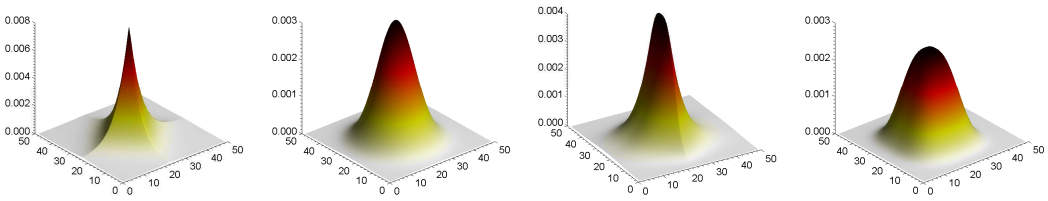


FIGURE 2.3: Illustration of the EM algorithm for a Gaussian mixture model applied to the Old Faithful data. (a) Initial model. (b) E-step: The posterior probability is indicated by the color of each point. (c) The updated parameters after the first M step. (d) After 7 iterations. (f) After 10 iterations. (f) After 14 iterations iterations. Based on [91].

2.2.3 Mixture of generalized Gaussian distributions

The generalized Gaussian distribution (GGD) which is an extension of the Gaussian and the Laplacian distributions [20], has the following formulation:



$\lambda = (1.1, 1.1)$ $\lambda = (2, 2)$ $\lambda = (1.1, 2.8)$ $\lambda = (2.8, 2.8)$

FIGURE 2.4: Different illustrations of the GGD distribution as a function of the shape parameter λ. The location and scale parameters μ and σ are fixed at $(23, 24)$ and $(7, 7)$ respectively. Based on [3].

$$p(x_i|\mu, \sigma, \lambda) = A(\lambda, \sigma) \exp\big(- B(\lambda) \, |(x_i - \mu)/\sigma|^{\lambda} \big) \tag{2.12}$$

where $A(\lambda, \sigma) = \lambda\sqrt{\Gamma(3/\lambda)/\Gamma(1/\lambda)}/(2\sigma\Gamma(1/\lambda))$ and $B(\lambda) = [\Gamma(3/\lambda)/\Gamma(1/\lambda)]^{\lambda/2}$; $\Gamma(.)$ being the gamma function. The parameters μ and σ are the GGD location and dispersion parameters. The parameter λ controls the kurtosis of the pdf and determines whether it is peaked or flat: the larger the value of λ, the flatter the pdf; and the smaller λ is, the more peaked the pdf. This gives the pdf a flexibility to fit the shape of heavy-tailed data [20]. Two well-known special cases of the GGD model are the Laplacian, as $\lambda = 1$, and the Gaussian distribution, as $\lambda = 2$ (see Figure 2.4) [3].

With a mixture of K GGDs, the probability of random variable X_i is given by

$$p(x_i|\Theta) = \sum_{j=1}^{K} \omega_j p(x_i|\mu_j, \sigma_j, \lambda_j), \tag{2.13}$$

Where $0 \leq \omega_j \leq 1$ and $\sum_{j=1}^{K} \omega_j = 1$. The parameters of the mixture with K components are $\Theta = (\omega_1, ..., \omega_K, \theta_1, ..., \theta_K)$ where $\theta_j = (\mu_j, \sigma_j, \lambda_j)/j = 1, .., K$ is a vector that contains the parameters set for the j-th distribution component of the mixture and $(\omega_1, ..., \omega_K)$ are the mixing parameters.

The maximum likelihood method consists of getting the mixture parameters that maximize the log-likelihood function given by

$$max_{\Theta}\{\log(P(X|\Theta))\} = max_{\Theta}\{\log(\prod_{i=1}^{N} \sum_{j=1}^{K} \omega_j p(x_i|\theta_j))\}, \tag{2.14}$$

with the constraint $\sum_{j=1}^{K} \omega_j = 1$. To take into account this constraint, a Lagrange multiplier is used and the following function is maximized [3]:

$$\Delta(X, \Theta, \Lambda) = log(P(X|\Theta)) + \Lambda(1 - \sum_{j=1}^{K} \omega_j), \tag{2.15}$$

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

$$\frac{\partial \Delta(X, \Theta, \Lambda)}{\partial \Theta} = 0, \tag{2.16}$$

$$\frac{\partial \Delta(X, \Theta, \Lambda)}{\partial \Lambda} = 0, \tag{2.17}$$

Straightforward manipulations yield the iterative equations:

$$\hat{\omega}_j = \frac{\sum_{i=1}^{N} p(j|x_i)}{N}, \tag{2.18}$$

$$\hat{\mu}_j = \frac{\sum_{i=1}^{N} p(j|x_i)|x_i - \mu_j|^{\lambda_j - 2} x_i}{\sum_{i=1}^{N} p(j|x_i)|x_i - \mu_j|^{\lambda_j - 2}}, \tag{2.19}$$

$$\hat{\sigma}_j = \left[\frac{\lambda_j A(\lambda_j) \sum_{i=1}^{N} p(j|x_i)|x_i - \mu_j|^{\lambda_j}}{\sum_{i=1}^{N} p(j|x_i)} \right]^{1/\lambda_j}, \tag{2.20}$$

Where

$$p(j|x_i) = \frac{\omega_j p(x_i|j)}{\sum_{k=1}^{K} \omega_k p(x_i|k)}. \tag{2.21}$$

For the parameter λ_j, The Newton-Raphson method is used. The following updating equation is obtained:

Data: Data sample $x = (x_1, ..., x_N)$, the component number (K).
Result: The mixture parameters vector (Θ).
initialization;
repeat
  `// E-step:`
  Compute the posterior probabilities $p(j|x_i)$;
  `// M-step:`
  for *For each component* $j = 1, ..., K$ **do**
    update $(\omega_j, \mu_j, \sigma_j, \lambda_j)$ *using Equations. (2.18) - (2.20) and (2.22)*;
  end
until *convergence*;

Algorithm 1: Expectation Maximization (EM) algorithm.

$$\hat{\lambda_j} \simeq \lambda_j - \left\{ \frac{\partial^2 \log[p(X|\Theta)]}{\partial \lambda^2} \right\}^{-1} \frac{\partial \log[p(X|\Theta)]}{\partial \lambda}. \tag{2.22}$$

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

2.3 Image Foreground Segmentation (IFS)

Image segmentation is a crucial task in computer vision, pattern recognition, and visual information retrieval fields by providing a compact and summary representation of the image [19]. Figure 1.2 illustrates several image segmentation applications such as satellite remote sensing, medical imaging, pedestrian detection, object detection, defect detection and document analysis. Image segmentation consists of partitioning a digital image into several *homogeneous regions* which may correspond to the objects in this image. *Homogeneous regions* refers to a connected set of pixels that share common features such as intensity, color, or texture, motion [19].

2.3.1 Definitions

Let $\Omega = \{(x, y) : 1 \leq x \leq W \text{ and } 1 \leq y \leq H\}$ stand for a $W \times H$ matrix of pixels (x,y). An observed image I is a function defined on the Ω domain, and for a given pixel (x, y) the observation $I(x, y)$ at that pixel takes a value from a set L. Two main examples for the set L are: $L = \{l : 0 \leq l \leq 255\}$ for gray level images, and $L = \{(l_1, l_2, l_3) : 0 \leq l_1 \leq 255, 0 \leq l_2 \leq 255, \text{ and } 0 \leq l_3 \leq 255\}$ for RGB (red, green and blue channel) color images. Image segmentation is also a function $\mathcal{S}$ on the same domain Ω, but for any given pixel (x, y) the segmentation $\mathcal{S}(x, y)$ at that pixel takes a value from a different set C. Two common examples for the set C are: $C = \{c : c = 0 \text{ or } 1\}$ denoting the two labels in a binary segmentation (in this case, the image segmentation process is usually called thresholding), and $C = \{c : c = 1, 2, 3, ..., K\}$ that corresponds to K different labels in the case of multi-class segmentation. Of course, $\mathcal{S}$, could also denote a boundary map. For any given point (x, y), $\mathcal{S}(x, y) = 1$ could denote the presence of a boundary at that pixel and $\mathcal{S}(x, y) = 0$ the absence of such a boundary [19].

Consider the image partition into a set of homogeneous and non-overlapping regions whose union is the entire image. Haralick and Shapiro [50] reported that the resulting parts of the image segmentation process should respect the four following rules:

1. Regions should be uniform and homogeneous with respect to some characteristics such as gray level or texture;
2. Region interiors should be simple and without many small holes;
3. Adjacent regions of segmented image should have significantly different values with respect to the characteristic on which they are uniform; and
4. Boundaries of each partition should be simple, not ragged, and must be spatially accurate.

A formal definition of image segmentation can be given as follows [41]. Let I be the observed image and let $\{R_i, i = 1, 2, ..., K\}$ are disjoint non-empty regions of I, a formal definition of image segmentation $\mathcal{S}$ consists of the following conditions [41]:

1. $\bigcup_{i=1}^{K} R_i = I$;
2. for all i and j, $i \neq j$, there exists $R_i \cap R_j = \varnothing$;
3. for $i = 1, 2, ..., K$, it must have $\mathbf{P}(R_i) = \text{TRUE}$;
4. for all $i \neq j$, there exits $\mathbf{P}(R_i \cup R_j) = \text{FALSE}$;

where $\mathbf{P}(R_i)$ is a uniformity predicate for all elements in the set R_i and $\varnothing$ represents the empty set.

In addition to these conditions, some researchers consider that the following condition is also important:

5. For all $i = 1, 2, ..., K$, R_i, is a connected component.

In the above, condition (1) indicates that the union of the resulting regions should contain all the entire image pixels. Condition (2) indicates that different resulting regions could not overlap each other. Condition (3) indicates that the pixels in the same segmented regions should have some similar characteristics. Condition (4) indicates that the pixels belonging to different segmented regions should have some different characteristics. Finally, condition (5) indicates that the pixels in the same segmented region are connected [157].

2.3.2 Image (histogram) thresholding

In some applications such as change detection [90], object recognition [11, 156] and document image analysis [116], the segmentation process consists to separate foreground objects from image background. In this case, the segmentation process can be estimated simply by thresholding the image at a particular intensity level. The basic idea of several thresholding methods is to exploit the shape of image histogram to establish an adaptive threshold. The results of this operation is a threshold $\mathbf{t}$ that separates the histogram into two parts that correspond to the foreground and background regions, respectively (i.e. $K = 2$). A generalization of this concept can be made when considering K regions ($R_1, ..., R_K$, with $K > 2$) which are separated by $K - 1$ thresholds $\mathbf{t}_1, ..., \mathbf{t}_{K-1}$, the image is segmented into K distinct regions. In most of existing industrial applications, the parameter K is generally known. Thresholding techniques have been extensively studied in the literature, surveys and comparative studies about existing thresholding-based segmentation methods can be found in [43, 112, 118, 150]. For example, Sezgin *et al.* [118] categorize the thresholding methods in six groups according to the information they are exploiting. These categories are histogram shape-based methods, clustering-based methods, entropy-based methods, object attribute-based methods, the spatial methods, and local methods.

2.3.2.1 Definitions

Let $X = \{x_1, x_2, ..., x_N\}$ be the gray levels of the pixels of an image I of size $N = H \times W$; H and W being the height and the width of the image. If the gray level range is not explicitly indicated as $[L_{min}, L_{max}]$, it will be assumed as:

$\{x_i \in [0, L],\ \text{for } i = 1, ..., N\}$, where L is the maximum gray level value, typically 255 if 8-bit quantization is assumed.

Let $\mathbf{t} = (\mathbf{t}_1, ..., \mathbf{t}_{K-1})$ be a set of thresholds that partitions an image into K classes. We consider the simple case of $K = 2$. One threshold t yields two classes: the foreground class $\mathcal{C}_1(t) = \{x : 0 \leq x \leq t\}$ and background class $\mathcal{C}_2(t) = \{x : t + 1 \leq x \leq T\}$, where T is the maximum gray level value in the image.

We denote by $h(x)$ the histogram frequency of the gray level x, where $\sum_{x=0}^{T} h(x) = 1$. The resulting histogram in this case ($K = 2$) is bimodal. The foreground (object) and background histograms are expressed as $h_1(x), 0 \leq x \leq t$, and $h_2(x), t + 1 \leq x \leq T$, respectively, where t is the threshold value. The foreground class probability (ω_1) and background class probability (ω_2) are calculated as:

$$\omega_1 = \sum_{i=0}^{t} h(i), \omega_2 = \sum_{i=t+1}^{T} h(i) = 1 - \omega_1 \tag{2.23}$$

The mean $\mu(t)$ and variance $\sigma^2(t)$ of the foreground class ($\mathcal{C}_1$) and background class ($\mathcal{C}_2$) as functions of the thresholding level t can be similarly defined as:

$$\mu_1(t) = \sum_{x=0}^{t} x.h(x), \sigma_1^2(t) = \sum_{x=0}^{t} (x - \mu_1(t))^2.h(x) \tag{2.24}$$

$$\mu_2(t) = \sum_{x=t+1}^{T} x.h(x), \sigma_2^2(t) = \sum_{x=t+1}^{T} (x - \mu_2(t))^2.h(x) \tag{2.25}$$

2.3.3 Popular approaches of image thresholding

Among the most used approaches for image foreground segmentation, statistical thresholding methods are popular for their simplicity and efficiency. Based on analysis of the image histogram, the statistical approaches search for an optimal threshold $\mathbf{t}$ that divides the histogram into two parts, the foreground with gray values lower than $\mathbf{t}$ and background for the remainder. For example, Otsu [97] described a method that uses inter-class separability to calculate optimal global thresholds between classes. Kittler and Illingworth (KI) [66] proposed the Minimum Error Thresholding (MET), a method based on the minimization of Bayes classification error, where each class is modeled by a Gaussian distribution. Both standard Otsu's and MET methods assume a unimodal shape for classes and use

sample mean and standard deviation to approximate their distributions. The relationship between the two methods is that the parameters can be obtained by either method using maximum likelihood estimation of a Gaussian model for each class [59, 150]. *Entropy* and *relative entropy* can also be used to derive suitable thresholds for image segmentation when the distribution of classes is Gaussian [21, 25, 113]. For example, Jiulun *et al.* [59] gave a relative-entropy interpretation for the minimum error thresholding (MET) [66, 89]. In that work, the Kullback-Leiber divergence [67] is used to measure the discrepancy between histograms of an observed image and a mixture of two Gaussians.

Recently, Xue *et al.* [148] proposed a thresholding method where a mixture of Laplacian distributions is used to model class data. They showed that the obtained thresholds offered better separation of classes when their distributions are skewed, heavy-tailed or contaminated by outliers. Indeed, the location and dispersion parameters of the Laplacian distribution are the median and the absolute deviation from the median, which are more robust to outliers compared to the sample mean and standard deviation, respectively [56]. In the following sections, we present in detail some of the above mentioned histogram-based image thresholding methods.

2.3.3.1 Otsu's thresholding method

The Otsu's method [97] determines the optimal threshold $\mathbf{t}$ using discriminant analysis i.e. by maximizing inter-class variation, or equivalently minimizing intra-class variation, which will lead to solve an optimization problem to find the threshold that maximizes one of the following objective functions:

$$\rho(t) = \frac{\sigma_B^2(t)}{\sigma_W^2(t)}, \kappa(t) = \frac{\sigma_G^2}{\sigma_W^2(t)}, \phi(t) = \frac{\sigma_B^2(t)}{\sigma_G^2}, \tag{2.26}$$

Where $\sigma_G^2(t)$ is the global variance and where $\sigma_W^2(t)$ and $\sigma_B^2(t)$ are the intra-class and inter-class variance respectively, defined as

$$\sigma_W^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t) \tag{2.27}$$

$$\sigma_B^2(t) = \omega_1(t)(\mu_1(t) - \mu(t))^2 + \omega_2(t)(\mu_2(t) - \mu(t))^2 = \omega_1(t)\omega_2(t)(\mu_1(t) - \mu_2(t))^2 \tag{2.28}$$

Criteria $\rho(t)$, $\kappa(t)$, and $\phi(t)$ are equivalent each other in term of maximization for a threshold t, because they are linked by the formula: for example $\kappa(t) = \rho(t) + 1$, $\phi(t) = \rho(t)/(\rho(t) + 1)$ in terms of $\rho(t)$, which arise from the following

basic equation:

$$\sigma_G^2 = \sigma_B^2(t) + \sigma_W^2(t) \tag{2.29}$$

We note that $\sigma_B^2(t)$ and $\sigma_W^2(t)$ are functions of the threshold t, however σ_G^2 is independent of t. We note also that $\sigma_W^2(t)$ is expressed in terms of second order statistics (variance), while $\sigma_B^2(t)$ is expressed in terms of first order statistics (means).

The threshold $\mathbf{t}$ that maximize $\kappa(t)$, or equivalently minimize the intra-class variation $\sigma_W^2(t)$, is given by:

$$\mathbf{t} = \arg\min_{\mathbf{t}} \{\omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)\}, \tag{2.30}$$

For multi-level thresholding, the Otsu's rule for selecting optimal thresholds $\mathbf{t} = (\mathbf{t}_1, ..., \mathbf{t}_{K-1})$ can be written as

$$\mathbf{t} = \arg\min_{\mathbf{t}} \sum_{k=1}^{K} \{\omega_k(t)\sigma_k^2(t)\}, \tag{2.31}$$

where $\omega_k(t)$ and $\sigma_k^2(t)$ present the probability and variance respectively, defined for the class $\mathcal{C}_k(t)$.

The Otsu's method remains one of the most referenced thresholding methods. This method gives satisfactory results when the pixel data in each class follows a uni-modal Gaussian distribution. However, This may lead to inaccurate approximation of the distribution mean μ_k and variance σ_k^2 of each class $\mathcal{C}_k$, especially in applications where the data is contaminated with noise and outliers.

2.3.3.2 Minimum Error Thresholding (MET)

Kittler and Illingworth's minimum error thresholding (MET) method [66] selects the threshold $\mathbf{t}$ under the assumption of gray level values of each class $\mathcal{C}_k$ following a Gaussian distribution. The proposed rule in the case of two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ that gives the threshold $\mathbf{t}$ is

$$\mathbf{t} = \arg\min_{t} \left\{ \omega_1(t)\log\frac{\sigma_1^2(t)}{\omega_1(t)} + \omega_2(t)\log\frac{\sigma_2^2(t)}{\omega_2(t)} \right\}, \tag{2.32}$$

where $\omega_1(t)$, $\omega_2(t)$, $\sigma_1(t)$ and $\sigma_2(t)$, defined in Equations. (2.23), (2.24) and (2.25), are positive here.

The multi-level-thresholding version of the MET method is given by

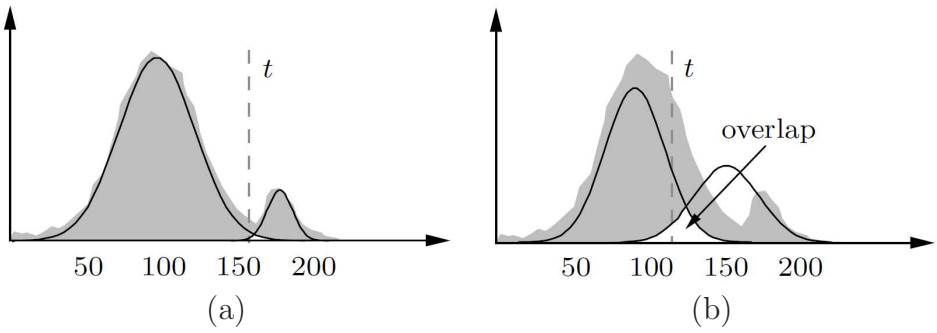


FIGURE 2.5: Illustration of the overlapping area between two classes. (a) good threshold, (b) bad threshold. Based on [66].

$$\mathbf{t} = \arg\min_{t} \sum_{k=1}^{K} \left\{ \omega_k(t) \log \frac{\sigma_k^2(t)}{\omega_k(t)} \right\}, \tag{2.33}$$

where $\omega_k(t)$ and $\sigma_k^2(t)$ present the probability and variance respectively, obtained for the class $\mathcal{C}_k(t)$.

The criterion given in (2.33) reflects indirectly the overlapping area between the Gaussian distributions. Each class $\mathcal{C}_k$ is represented as a Gaussian distribution with the mean μ_k and the variance σ_k^2. The optimal thresholds $(\mathbf{t}_1, ..., \mathbf{t}_K)$ are the ones which minimize the overlapping areas between these distributions and therefore minimizing the classification error. This behaviour is illustrated in Figure 2.5 for two different values of threshold $\mathbf{t}$. The value of threshold $\mathbf{t}$ yielding the minimum value of criterion Equation (2.33) will give the best fit and therefore the minimum error threshold [66].

2.3.3.3 Median-based image thresholding

Xue *et al.* [148] propose a median based extension to the Otsu's and MET methods to select an optimal threshold that is relatively robust to the presence of skew and heavy-tailed class-conditional distributions. The proposed approach is based on the mixtures of Laplace distributions. The basic idea of this extension is the use of the median instead of the mean to provide a threshold that is more robust to the presence of skew and heavy-tailed distributions than those selected by the Otsu's method and the MET method. The median-based version of the rule for selecting the optimal threshold $\mathbf{t}$ is defined as follows:

$$\mathbf{t} = \arg\min_{t} \{\omega_1(t) MAD_1(t) + \omega_2(t) MAD_2(t)\}, \tag{2.34}$$

where $MAD_1(t)$ and $MAD_2(t)$, the mean absolute deviation from the median for classes $\mathcal{C}_1(t)$ and $\mathcal{C}_2(t)$ respectively, are given by

$$MAD_1(t) = \sum_{x=0}^{t} \frac{h(x)}{\omega_1(t)} |(x - m_1(t)|, \tag{2.35}$$

$$MAD_2(t) = \sum_{x=t+1}^{T} \frac{h(x)}{\omega_2(t)} |(x - m_2(t)|, \tag{2.36}$$

in which $m_1(t) = med\{x_i : i \in \mathcal{C}_1(t)\}$ and $m_2(t) = med\{x_i : i \in \mathcal{C}_2(t)\}$ are the sample medians for the two classes $\mathcal{C}_1(t)$ and $\mathcal{C}_2(t)$, respectively.

Therefore, for multi-level thresholding, the rule underlying Otsu's median-based extension becomes

$$\mathbf{t} = \arg\min_{t} \sum_{k=1}^{K} \{\omega_k(t) MAD_k(t)\}, \tag{2.37}$$

where $\omega_k(t)$ and $MAD_k(t)$ present the probability and the mean absolute deviation from the median respectively, defined for the class $\mathcal{C}_k(t)$.

By analogy with the median extension for the Otsu's method, the rule underlying a median-based extension of the MET method can be derived by substituting the MAD for σ as

$$\mathbf{t} = \arg\min_{t} \left\{ \omega_1(t) \log \frac{MAD_1(t)}{\omega_1(t)} + \omega_2(t) \log \frac{MAD_2(t)}{\omega_2(t)} \right\}, \tag{2.38}$$

The median extension rule for the multi-level thresholding case of the MET method is given as

$$\mathbf{t} = \arg\min_{t} \sum_{k=1}^{K} \left\{ \omega_k(t) \log \frac{MAD_k(t)}{\omega_k(t)} \right\}, \tag{2.39}$$

The median-based extension of the Otsu's method is derived in a natural way by substituting the median for the mean and the MAD for the variance. But the assumption that the data have a Laplacien distribution is not always realistic.

Previous methods for image foreground segmentation were basically implemented to separate unimodal classes. Therefore, they are not adapted to deal with multi-modal class segmentation. For example, in many segmentation applications such as medical images or document analysis, the image foreground and/or background region may have a multi-modal distribution. In addition, the standard

methods are limited to the assumption that each class data is Gaussian like in the case of Otsu's method and the MET method or Laplacien such as the median-based one. In several image examples, one can find histogram modes with different other shapes such as heavy tailed ones, making this assumption not always realistic. To overcome this limitation, different mixture-based approaches have been proposed. We present some of them in the following section.

2.3.3.4 Non-Gaussian mixture models

Recently several researchers showed that the use of the non-Gaussian distributions (instead of the Gaussian) could explicitly improve the efficiency and robustness of modeling the data and hence improve the performance of the image thresholding algorithms based on these models. The use of such statistical models is justified by the fact that the Gaussian assumption can not handle a broad range of industrial signals such as non-gaussian noise and illumination effects to name a few.

Among the existing non-Gaussian models, is the generalized Gaussian distribution (GGD) which is an extension of the Gaussian and the Laplacian distributions [20] (see Section 2.2.3). Bazi *et al.* [9] assume that the 'foreground' and 'background' classes follow a unimodal generalized Gaussian distribution (GGD) and use the expectation-maximization (EM) algorithm to estimate the statistical parameters of each class. An initialization procedure based on genetic algorithms (GAs) is proposed to provide good initial conditions of the EM algorithm. In the same vein, Fan *et al.* [37] develop a global thresholding algorithm based on the generalized Gaussian mixture modeling. The proposed solution combines the particle swarm optimization (PSO) method with the EM algorithm to estimate the GGD parameters accurately.

Nacereddine *et al.* [92] consider the image histogram to be a mixture of asymmetric generalized Gaussian distributions (AGGD). In addition to the fact that the AGDD can fit a large class of statistical distributions (e. g. Laplacian and Gaussian), the AGGD includes many of symmetric as well as asymmetric distributions. The gray level histogram will be modeled by a mixture of univariate AGGDs. The EM algorithm is used to estimate the mixture model parameters.

Moser *et al.* [90] propose an automatic process to change detection for synthetic aperture radar (SAR) images. The proposed approach is based on a generalization of the Kittler and Illingworth Minimum-Error Thresholding (MET) algorithm by taking into account the non-Gaussian distribution of SAR images. In particular, the three non-Gaussian models "Nakagami-ratio", the "Weibull-ratio" and the log-normal are used to model the change and no-change hypothesis. Each resultant version is endowed with suitable parameter estimation algorithms based on the method of LogCumulants (MoLC).

Xue *et al.* [147] show that Rayleigh distributions can best approximate histograms of different patterns in SAR image. In fact, both Gaussian distribution and Poisson distribution have some weakness. The Gaussian distribution dispersed from negative to positive coordinates, which does not satisfy real situation as no negative gray level in the image. The Poisson distribution has equal mean and deviation, which does not fit to image property [147]. Thus, a minimum error thresholding (MET) algorithm is developed under the assumption that gray level histogram of SAR image fits a finite mixture of shifted Rayleigh distributions. The new MET criterion based on shifted Rayleigh distribution is given as

$$\mathbf{t} = \arg\min_{t} \Big\{ \sum_{i=1}^{2} \omega_i(t) \big[\log R_i(t) - \log\omega_i(t)\big] - \sum_{x=0}^{t} h(x)\log(x-g_1) - \sum_{x=t+1}^{T} h(x)\log(x-g_2) \Big\}, \tag{2.40}$$

where g_1, g_2 are Rayleigh distribution parameters and $R_1(t)$ and $R_2(t)$ can be estimated as $R_1(t) = \frac{1}{\omega_1(t)} \sum_{x=1}^{t} h(x)(x-g_1)$ and $R_2(t) = \frac{1}{\omega_2(t)} \sum_{x=t+1}^{T} h(x)(x-g_2)$. Pal *et al.* [35, 98] propose a minimum error thresholding (MET) algorithm based on the Poisson distribution instead of the Gaussian distribution to model the histogram of gray level images. To obtain the optimal threshold $\mathbf{t}$, Pal *et al.* propose to minimize the following minimum-error criterion function:

$$\mathbf{t} = \arg\min_{t} \big\{ \mu - \omega_1(t)\big(\log\omega_1(t) + \mu_1(t)\log\mu_1(t)\big) - \omega_2(t)\big(\log\omega_2(t) + \mu_2(t)\log\mu_2(t)\big) \big\}, \tag{2.41}$$

where μ corresponds to the image mean, $\mu_1(t)$, $\mu_2(t)$ correspond to the class means at threshold t for $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively and $\omega_1(t)$ and $\omega_2(t)$ define the class probabilities at threshold t for the two classes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively.

The aforementioned works use a single non-Gaussian distribution to represent each data class $\mathcal{C}_k$ i.e. they assume that each data class must be fitted by only one component. As mentioned above, in several image foreground segmentation applications such as medical imaging or satellite remote sensing, the foreground and/or background data may be composed of many components. In addition, the data in each class $\mathcal{C}_k$ can be contaminated by several kinds of noise and outliers which means that the different shapes of the data histogram such as skewed, sharply peaked or heavy tailed should be supported by the probability distribution. Therefore, a robust multi-component modeling of each class data such as mixture modeling is recommended.

2.3.4 Evaluation metrics for image thresholding

Performance evaluation of thresholding algorithms is an important task to assess the performance of the proposed algorithms as well as the comparison to the state-of-the-art ones. The dissimilarity between the segmented image and the ground-truth image (i.e. the hand segmented image) can be used to evaluate the performance of one or several algorithms. Many evaluation metrics has been proposed, for example, the Misclassification Error (ME) and the Dice Coefficient (DC).

The following quantities are involved:

- True positives (TP): the number of foreground pixels correctly detected;
- False positives (FP): the number of background pixels incorrectly detected as foreground (also known as false alarms);
- True negatives (TN): the number of background pixels correctly detected; and
- False negatives (FN): the number of foreground pixels incorrectly detected as background (also known as misses).

Based on the quantities mentioned above, segmentation evaluation performance can be described using the following coefficients [110]:

2.3.4.1 The Misclassification Error (ME)

$$ME = 1 - \frac{TP + TN}{TP + TN + FP + FN}, \tag{2.42}$$

2.3.4.2 The Jaccard Coefficient (JC)

$$JC = \frac{TP}{TP + FP + FN}, \tag{2.43}$$

2.3.4.3 The Dice Coefficient (DC)

$$DC = 1 - 2\frac{TP}{2TP + FP + FN}, \tag{2.44}$$

Where TP, FN, FP and FN stand for true positive, false negative, false positive and false negative, respectively.

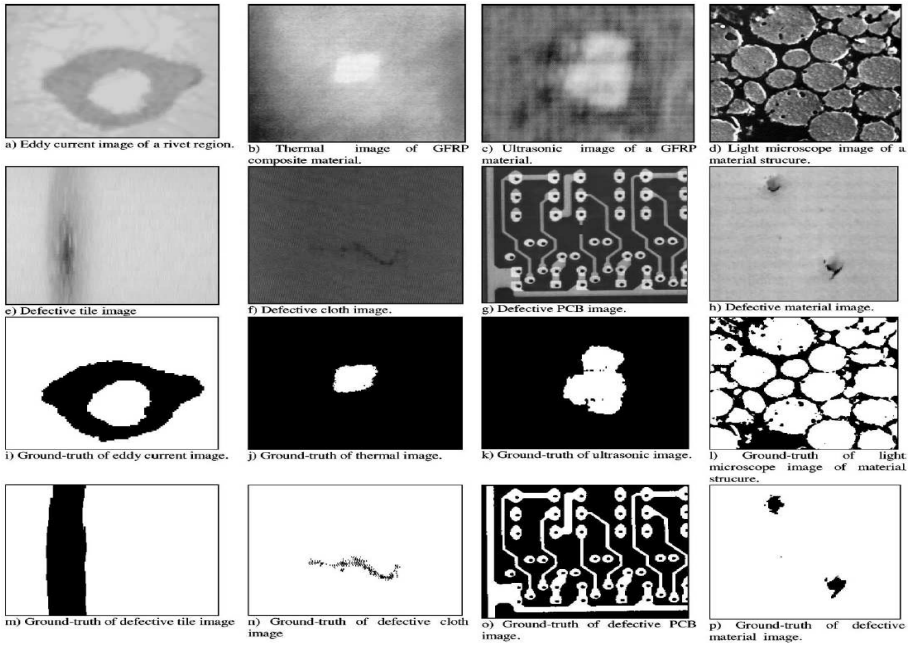


FIGURE 2.6: Sample NDT images and their ground truths. Based on [118].

2.3.4.4 Image foreground segmentation datasets

Image foreground segmentation evaluation is an important issue since it makes possible the quantitative evaluation and comparison between the proposed algorithm and the state-of-the-art algorithms. Several datasets have been proposed over the last decades to evaluate the performance of image foreground segmentation algorithms. Usually, the ground truth of each image is suitably defined and publicly available. We give here a selected list of 5 datasets for the evaluation of image foreground segmentation.

2.3.4.4.1 NDT dataset This dataset is proposed by Sezgin *et al.* [118] in a survey about image thresholding techniques. It consists of a set of 40 NDT greyscale and 40 document greyscale images. The NDT images consisted of 8 eddy current, 4 thermal, 2 ultrasonic, 6 light microscope, 4 ceramic, 6 material, 2 PCB, and 8 cloth images (see Fig. 2.6). Documents containing ground-truth character images were created with different fonts (times new roman, arial, comics, etc.), sizes (10,12,14), and typefaces (normal, bold, italic, etc.). In addition, more realistic effects such as the effects of the poor quality of paper, photocopied and faxed documents, etc. have been generated using degradation models [118].

2.3.4.4.2 BrainWeb: Simulated Brain Database The BrainWeb dataset provides a set of realistic MRI data volumes generated by an MRI simulator [69]. These brain MRI data can be used to evaluate the performance of the proposed image segmentation methods in a setting where the ground truth is known. Currently, the BrainWeb dataset contains simulated brain MRI data based on two anatomical models: normal and multiple sclerosis (MS). For both of these, full 3-dimensional data volumes have been simulated using three sequences (T1-, T2-, and proton-density- (PD-) weighted) and a variety of slice thicknesses, noise levels, and levels of intensity non-uniformity. These data are available at the website of BrainWeb dataset [1]. Fig. 2.7 shows some examples of brain MRI slices from this dataset.

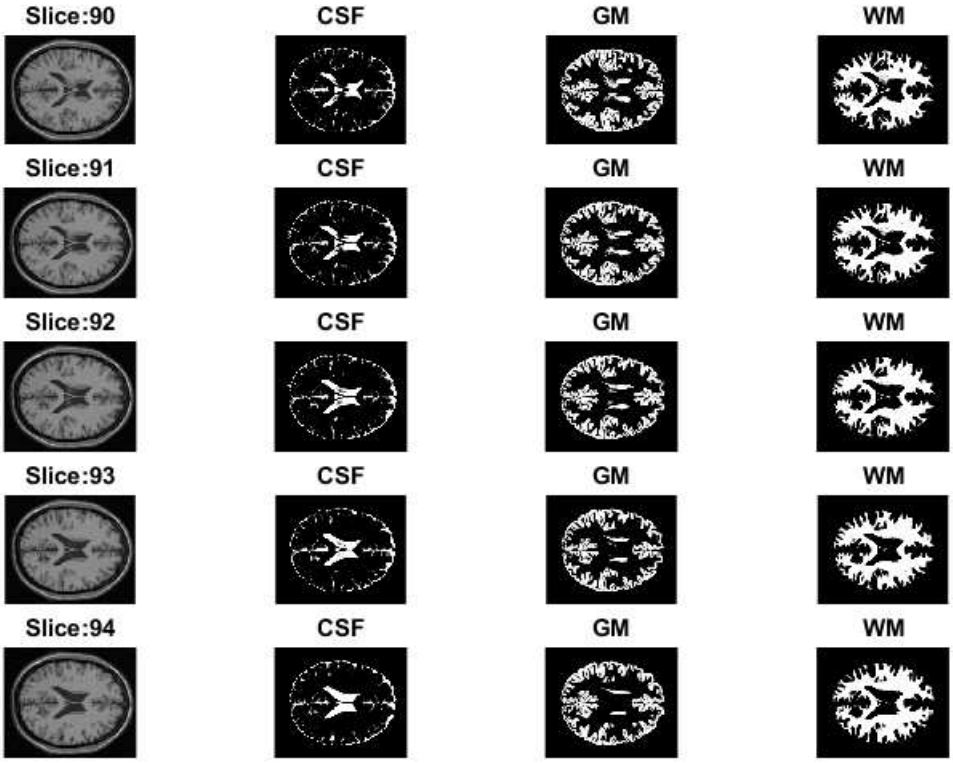


FIGURE 2.7: Examples of Brain MRI slice images from the BrainWeb dataset.

2.3.4.4.3 Berkeley Segmentation Database The goal of this dataset is to provide a tool for evaluating the performance of image segmentation algorithms

[1] http://brainweb.bic.mni.mcgill.ca/brainweb/

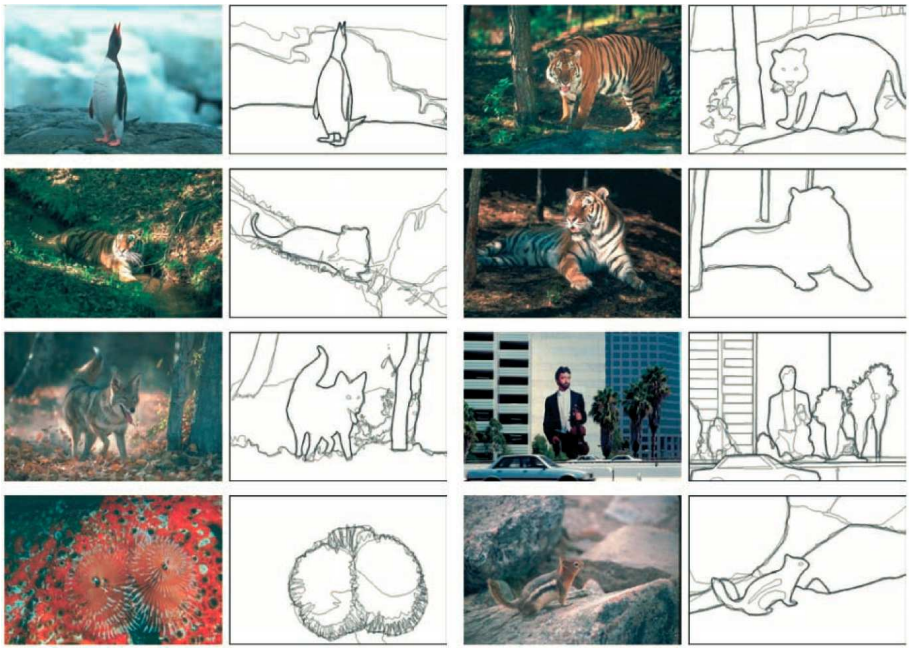


FIGURE 2.8: Example of images and ground truth from the BSD dataset. Each image shows multiple (4-8) human segmentations. The pixels are darker where more humans marked a boundary. Based on [83].

[83]. The BSD dataset contains 300 color images of size 481×321 pixels. For each of these images, the database provides between several ground truth maps. The image segmentations are provided considering the color and grayscale versions of each image. The dataset based on this data consists of all of the grayscale and color segmentations for 300 images. The complete benchmark is divided into two sets: a training set of 200 images, and a test set consisting of the remaining 100 images (see Fig. 2.8 for an illustration). The dataset images along with their ground truth and MATLAB code are available publicly at the BDS website [2].

2.3.4.4.4 GrabCut dataset To evaluate the GrabCut method [111], the authors designed a new ground truth database of 50 images. The ground truth information is given in three types of information:(1) Segmentation: A tri-map which specifies background (0), foreground (255) and mixed area (128). The mixed area contains pixels which are a combination of foreground and background texture. Note, in low contrast regions the true boundary is not observed and the ground

[2]https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/

(a) (b) (c) (d)

FIGURE 2.9: Example of images and ground truth from the GrabCut dataset [111]. (a) Original image. (b) Ground truth. (c) Labelling-Lasso. (d) Labelling-Rectangle.

truth is in this case a "good guess". (2) Labelling-Lasso: Imitates a tri-map obtained by a lasso or pen tool. The colour coding is: background (0); background - used for colour model training (64); inference (unknown) region (128); foreground - used for colour model training (255). Note, a lasso tool can be imitated by specifying the foreground region (255) as unknown (128). (3) Labelling-Rectangle: Imitates a tri-map obtained by two mouse clicks (rectangle). Same colour coding as in Labelling-Lasso (see Fig. 2.9).

2.3.4.4.5 MS-COCO dataset The Microsoft Common Objects in COntext (MS-COCO) dataset [77] contains more than $200K$ images and 80 foreground object categories. The training set has about $80K$ images, and $40K$ images for validation. Fig. 2.10 shows some annotated images from the MS-COCO dataset. All object instance are annotated with a detailed segmentation mask. Annotations on the training and validation sets (with over $500K$ object instances segmented) are publicly available at the dataset website [3].

2.4 Video Foreground Segmentation (VFS)

2.4.1 Definitions and challenges

Background Subtraction (BS) is a fundamental and crucial task for several video processing applications such as smart video surveillance [32], human activity recognition [132] and interactive gaming [39]. The background subtraction purpose is to separate the foreground moving objects from the static background in video sequences. As shown in Figure 2.11, the background subtraction process can be composed of three steps: (1) Background initialization using first frames to obtain the free background (background image without objects inside). (2) Background classification: the foreground detection is made by classifying pixels as foreground

[3] http://mscoco.org/home/

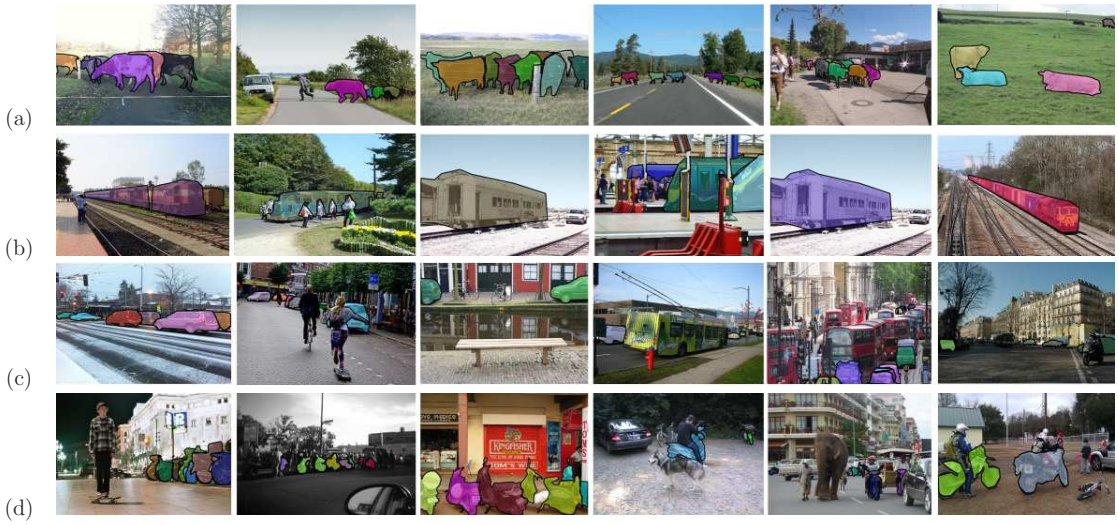


FIGURE 2.10: Samples of annotated images in the MS-COCO dataset. (a) Cow category. (b) Train category. (c) Car category. (d) Motorbike category. Based on [77].

or background according to the comparison between the current frame and the background model frame. (3) Background updating: to update the background model over time. The steps (2) and (3) are repeated for all the sequence frames [18].

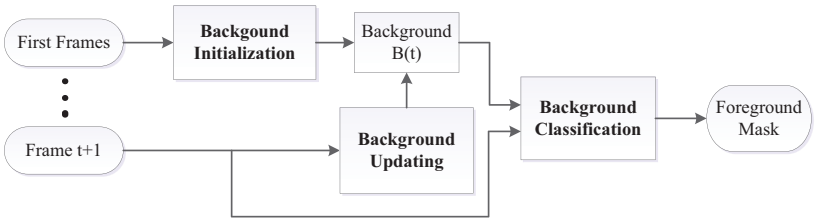


FIGURE 2.11: Background subtraction process [22].

Generally, to simplify the background subtraction problem and ensure success, the proposed algorithms assume mainly three conditions: stationary cameras, constant illumination conditions and static background (i.e., no dynamics or noise occur in the background). Several challenges result from the violation of these three assumptions. Different enumerations of these challenges exist in the literature. Toyama et *et al.* [131] defined 10 challenging situations in the field of video

surveillance. Recently, Bouwmans et *et al.* [18] proposed an extended list of 13 background subtraction challenges. The following sentences describe some of the most important ones:

- *Cast shadows:* tend to be classified as parts of the foreground. Indeed, they may generate patterns of movement which have the same magnitude and direction as the objects that generated them [104, 115].
- *Dynamic backgrounds:* can be caused by background objects, such as swaying tree leafs, water flowing, etc. They can cause a significant number of false positives for foreground detection.
- *Noisy videos:* noise can be generated by several sources such as sensor noise, low-quality cameras or compression artifacts. Indeed, noisy videos tend to produce numerous false detections.
- *Camera jitter:* can be caused by camera instability (e.g., by wind or vibrations). If the resulting camera motions are not accounted for by a robust background subtraction method, they can lead to many false detections.
- *Illumination changes:* can be gradual such as light variation in outdoor scenes or sudden such as a light switch in a room. For example, some sudden illumination changes can affect all pixels, which may produce a foreground mask containing a big amount of false detections.
- *Stopped objects:* foreground objects that stop or slow momentarily their motion can be confused with the background and ultimately added to it. For instance, a car may stop at a traffic junction waiting for the green signal. If the standing time is too long, the car can be merged rapidly with the background.
- *Camouflage effects:* occur when a moving object or some of its parts are made of colors similar to the background. They may cause false negatives for foreground detection.
- *Bootstrapping:* Bootstrapping approaches are used to initialize the background model when the free background is not available in the beginning stage of the background subtraction process.

Over the past years, several foreground detection techniques have proposed background models that address some of these challenges [17, 120]. Some methods have used parametric models such as the Gaussian mixture models (GMMs) [61, 120, 125] to cope with complex backgrounds. Other methods are rather data-driven and use non-parametric models [8, 33, 52, 96], or propose models that

Methodology	References
Single Gaussian	Wren *et al.* [144]
Mixture of Gaussian model (MoG)	Stauffer and Grimson [125], Friedman and Russell [40]
Mixture of Generalized Gaussian model (MoGG)	Allili *et al.* [3]
GMM with dynamic number of models	Zivkovic and van der Heijden [160]
Fast GMM	Haque *et al.* [49], KaewTraKulPong and Bowden [61], Lee [70]
Self-adaptive GMM	Greggio *et al.* [45], Chen *et al.* [29]
Illumination-based GMM	Pilet *et al.* [100], Shah *et al.* [119]
Gradient-based GMM	Zhao and Lee [158], Izadi and Saeedi [58]
GMM with shadow removal	Martel-Brisson and Zaccarin [82]
GMM using eigenbackground	Vosters *et al.* [136]
Self-organizing with artificial neural networks	Maddalena and Petrosino [79], Singh *et al.* [123]
Adaptive patch-based background modeling	Reddy *et al.* [105], Zhao and Lee [158]
Scale invariant local pattern	Liao *et al.* [75]
Non-parametric density estimations	Elgammal *et al.* [33], Han *et al.* [48]
Low Rank Minimization	Guyon *et al.* [47]
Support Vector	SVM [76], SVR [139], SVDD [129]
Clusters	K-means [23], Codebook [63]
Subspace Learning	PCA [96], ICA [151]
Transform Domain	FFT [145], DCT [102], Wavelet [42], Hadamard [7]
Advanced Statistical Non Parametric Models	ViBe [8], PBAS [52], SuBSENCE [26]

Table 2.2: Methodologies and references for video foreground segmentation [29].

cope with specific challenges, such as shadows [115] or illumination changes [3, 117]. However, no method is guaranteed to yield high performance in all the challenges [17]. Several surveys in the literature dedicated to reviewing traditional and recent techniques [17, 18, 99, 104]. Table 2.2 summarizes some approaches used to perform video foreground segmentation along with their corresponding references [29].

2.4.2 Popular approaches of background subtraction

2.4.2.1 Parametric methods for BS

An efficient and adaptive approach to video foreground segmentation is to construct a parametric model which represents the probabilistic distribution of the pixel's intensity or color. Wren *et al.* [144] adopted a single Gaussian to represent the background model. A more efficient statistical approach for background modeling is using the finite Gaussian mixture model (GMM) [61, 125], for example, Stauffer and Grimson [125] use the GMMs to model the history of each pixel intensity along with an online version of the EM algorithm to estimate the model

parameters.

2.4.2.1.1 Single Gaussian Model (SGM) Firstly, a unimodal mixture Gaussian distribution was used in [144], where a real-time system for tracking people is proposed. The history of pixel YUV color data is modeled using one Gaussian distribution $\eta(x_t|\mu_t, \Sigma_t)$ where x_t represents the YUV color vector at pixel (x, y) and time t and where μ_t and Σ_t correspond to the mean and covariance matrix, respectively. The distance metric can be defined as the following Mahalanobis distance:

$$d_M(t) = |x_t - \mu_t|\Sigma_t^{-1}|x_t - \mu_t|^T, \tag{2.45}$$

Considering the background changes in time, the background model B_t is updated using the running (or on-line cumulative) average approach described as follows:

$$\mu_{t+1} = (1 - \alpha).\mu_t + \alpha.x_{t+1}, \tag{2.46}$$

The covariance of each pixel can also be iteratively updated following this equation:

$$\Sigma_{t+1} = (1 - \alpha).\Sigma_t + \alpha.(x_{t+1} - \mu_{t+1})(x_{t+1} - \mu_{t+1})^T, \tag{2.47}$$

Where α is a learning parameter. Simple thresholding is then applied to decide whether a pixel is a foreground or background pixel, and the background subtraction results are represented using a binary foreground mask FG, which is computed for each pixel (x, y) at time t as follows:

$$FG(t) = \begin{cases} 1 & \text{if } d_M(t) > \tau \\ 0 & \text{otherwise} \end{cases} \tag{2.48}$$

where τ is a threshold that is set empirically and $d_M(t)$ is the Mahalanobis distance obtained from Equation (2.45). Note that the Σ_t matrix can be assumed to be diagonal to reduce memory and processing costs.

Low memory requirement and speed are two advantages of the single Gaussian model. In fact, for each pixel (x, y), only two parameters (μ, Σ) are required instead of the buffer with N history pixel values [99]. In turn, the unimodality assumption is relatively true when the scene consists of a static situation such as

an office and a moving person. However, this method cannot perform well with complex scenarios that require multi-modal density functions such as the presence of dynamic backgrounds or sudden illumination changes.

2.4.2.1.2 Gaussian Mixture Model (GMM) A mixture of three Gaussian distributions model is proposed in [40] to deal with the traffic surveillance problem. The three components are associated with the road, shadows, and vehicles. The RGB color data for each pixel over the time is employed to learn a probabilistic model using the Expectation-Maximization (EM) algorithm.

More realistic Gaussian mixture models (GMMs) are given [61, 125]. The basic GMM idea consists to model the recent history of the pixel (x, y), which are $\{x_1, ..., x_N\}$ by a mixture of K Gaussian distributions. For example the pixel (x, y) at time t is defined by its RGB intensity vector $x_t = (x_{t,1}, x_{t,2}, x_{t,3})$. The probability of appearance of a color to a given pixel is given by:

$$p(x_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(x_t|\theta_{i,t}), \tag{2.49}$$

where $\theta_{i,t} = (\mu_{i,t}, \Sigma_{i,t})$ are the mean and variance parameters describing the ith Gaussian mixture component at time t, respectively, $\omega_{1,t}$,..., $\omega_{K,t}$ are the component weights such that $\sum_{i=1}^{K} \omega_{i,t} = 1$, K is a fixed parameter that represents the maximum number of foreground/background components, and η is the Gaussian *pdf* defined as:

$$\eta(x_t|\mu_t, \Sigma_t) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma_t|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_t-\mu_t)\Sigma_t^{-1}(x_t-\mu_t)^T}. \tag{2.50}$$

Note that the covariance matrix Σ_t can be simplified to be diagonal $\Sigma_t = \sigma_t^2 Id$ due to computational reasons.

At frame $t + 1$, each pixel is compared with the current background model, if a component is matched (i.e. the pixel intensity is within 2.5 standard deviation of its mean) then the mixture parameters are updated recursively using an online K-means approximation as follows:

$$\omega_{t+1} = (1 - \alpha)\omega_t + \alpha. \tag{2.51}$$

$$\mu_{t+1} = (1 - \rho)\mu_t + \rho x_{t+1}. \tag{2.52}$$

$$\sigma_{t+1}^2 = (1 - \rho)\sigma_t^2 + \rho(x_{t+1} - \mu_{t+1})(x_{t+1} - \mu_{t+1})^T. \tag{2.53}$$

Where ρ represents the learning rate and α is named the learning factor. If there is no match, the location and scale parameters (μ and σ respectively) are unchanged, and the weight parameter is reduced using the formula:

$$\omega_{t+1} = (1 - \alpha)\omega_t. \tag{2.54}$$

If no component among the K components matches the pixel intensity vector x_{t+1}, then the component with small weight is replaced with a new Gaussian with a mean value x_{t+1}, a large variance σ_0^2 and a small weight ω_0. After updating the K weights, they are normalized, so they sum up to 1. Then, the K components are ordered based on the values of ω/ρ and the background model is composed of only B first components, such that:

$$B = arg \min_b \left(\sum_{i=1}^{b} \omega_{i,t+1} > \tau \right). \tag{2.55}$$

where τ is a threshold (usually $\tau = 0.8$), that represents the minimum portion of data considered to belong to the background model.

2.4.2.1.3 Limitations of the GMM model The original GMM method developed by Stauffer and Grimson [125] is widely used for background modeling of static camera sequences. The GMMs are able to cope with different challenges such as gradual illumination changes and background dynamics with small repetitive motions (e.g., moving vegetation, monitor flickering, etc.) [70, 85, 125, 143, 159]. However, major limitations exist for this model.

- First, GMMs represent well foreground objects appearing and disappearing faster than the background. Slow objects (e.g., stopping cars, a person waiting in a queue) tend to be rapidly absorbed by the background.
- The second limitation lies in the difficulty to cope with shadows which increase false foreground detections since the shadows are usually segmented to the foreground.
- The third limitation lies in the hand-tuning of the GMM parameters which, in general, are manually set in advance (e.g., the learning rate). For example, too high learning rates will accelerate object absorption by the background, which may cause false negatives; whereas, too low learning rates will prevent the model from absorbing background dynamics and cause false positives.

- Finally, GMMs incur a significant computational time and memory, especially for high-resolution videos.

2.4.2.1.4 Extensions and improvements to the GMM model Since the first use of GMMs for BS [40, 125], several improvements have been proposed to mitigate the aforementioned limitations:

- **Parameters updating**: The concept of hysteresis thresholding is used by Power and Schoonees [103] to fill the holes and gaps that can be generated by the original version proposed by Stauffer and Grimson [125]. Zivkov *et al.* [159] propose a scheme to update the GMM parameters automatically according to the video data. However, the algorithm does not handle shadows or complex scene backgrounds. In [61], new equations have been proposed for updating the learning rate of the GMM. The approach ensures fast background recovering when the sequence starts with a foreground object at a given location. In [85] a method has been proposed for adapting detection thresholds to varying video statistics. Moreover, a foreground model based on small spatial neighborhood and Markov models has been proposed to improve background subtraction. In [70], the static learning factor has been replaced by an adaptive one, which is calculated for each Gaussian at every frame. In the same vein, a particle swarm optimization has been used to estimate the GMMs parameters [143].

- **The choice of the feature space**: The selection of the adequate feature space is a crucial task for the background subtraction or modeling process. Stauffer and Grimson [125] use the RGB color space to represent the pixel value in the GMM model. Other space colors are proposed such as Normalized RGB, YUV, HSI and Luv spaces. In addition, the gradient is used as a feature to deal with local sudden illumination changes [58, 158]. Shah et *et al.* [119] combine the GMM model with SURF features to propose a framework that allows the GMM algorithm to adapt local parameters quickly and remove local illuminations changes. Yoshinaga *et al.* [154] applied a GMM to a local difference pattern to cope with background subtraction challenges such as illumination changes and dynamic backgrounds.

- **Non-Gaussian probability distributions**: Instead of using the GMMs, Allili *et al.* [3] use the mixture of generalized Gaussian (MoGG) distribution, which is a generalization of the GMM, to model the background along with an Expectation-Maximization (EM) based procedure to update the mixture parameters. However, the procedure uses Fisher scoring which incurs a huge computation time to calculate the likelihood derivatives. Similarly, the

mixture of asymmetric generalized Gaussians is used in [34] to enhance the robustness and flexibility of the temporal model.

Most of these algorithms achieve some automation in updating and adapting the GMMs parameters to video data. However, their performance drastically decreases with challenges such as complex background dynamics, thick shadows and camera jitter.

2.4.2.2 Non-parametric methods for BS

2.4.2.2.1 Kernel Density Estimation (KDE) Several works have addressed dynamic background modeling using non-parametric approaches. Elgammal *et al.* [33] have introduced the kernel density estimation (KDE) for background modeling. Effects of small motions are reduced by enforcing spatial coherence for background subtraction.

Let $\{x_1, x_2, ..., x_N\}$ be a recent history of intensity values for a pixel (x, y) and let x_t be a d-dimensional vector that represents the pixel (x, y) at time t. The probability that the pixel (x, y) at time t will have the value x_t can be estimated non-parametrically using the kernel estimator $\mathbf{E}$ as:

$$p(x_t) = \frac{1}{N}\sum_{i=1}^{N}\mathbf{E}(x_t - x_i) \tag{2.56}$$

If we choose the Gaussian distribution *pdf* $\eta(0, \Sigma)$ as the kernel estimator $\mathbf{E}$, then the density function can be estimated as:

$$p(x_t) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}}e^{-\frac{1}{2}(x_t - x_i)^T\Sigma^{-1}(x_t - x_i)} \tag{2.57}$$

Where Σ represents the kernel function bandwidth. Considering the simple case of independence between color channels, then Σ can be rewritten as

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & .. & 0 \\ 0 & \sigma_2^2 & .. & 0 \\ 0 & 0 & .. & 0 \\ 0 & 0 & .. & \sigma_d^2 \end{pmatrix} \tag{2.58}$$

where σ_j^2 is the bandwidth for the j^{th} channel and the kernel density can be reduced to

$$p(x_t) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d}\left\{\frac{1}{\sqrt{2\pi\sigma_j^2}}e^{-\frac{(x_{t,j}-x_{i,j})^2}{2\sigma_j^2}}\right\} \tag{2.59}$$

The pixel (x, y) is considered foreground at time t if $p(x_t) < \tau$, where τ is a global threshold.

KDE ensures a smooth and continuous version of background distributions. However, it is very demanding in terms of computational time and memory storage. Besides, shadows and illumination changes are not handled using such an approach.

2.4.2.2.2 Pixel-Based Adaptive Segmenter (PBAS) Recently, the pixel-based adaptive segmenter (PBAS) method has been proposed [52]. PBAS is a non-parametric approach that uses the history of pixel values to build a background model at each pixel and uses a dynamic threshold to detect the foreground. Considering the current frame and the background model, the background/foreground decision is based on a per-pixel threshold. In order to detect changes in the background such as illuminations changes or dynamic background objects, the background model is generated according to a dynamic updating rate. This method performs well with some dynamic backgrounds. However, foreground objects tend to be absorbed by the background from outside after a certain time. Besides, the approach does not explicitly deal with shadows and illumination changes.

2.4.2.2.3 Self-Balanced SENsitivity SEgmenter (SuBSENSE) Inspired by PBAS, the pixel-level based algorithm called Self-Balanced SENsitivity SEgmenter (SuBSENSE) has been proposed in [26]. This algorithm combines RGB color intensities and *local binary similarity patterns* (LBSP) string features [12, 51]. The resulting samples carry both local RGB color information and spatiotemporal neighborhood similarity information. The comparison of colors is done using L_1 distance and on binary strings using Hamming distance. The algorithm uses a pixel-level feedback loop for updating the decision threshold and learning controllers. This enables the algorithm for effective modeling of several types of background dynamics (e.g. water dropping from a fountain or camera movement caused by jitter effect) and regions with intermittent dynamic changes. However, it does not deal efficiently with illumination changes, shadows, and camouflage problems.

2.4.2.2.4 Other non-parametric approaches Eigenvalue decomposition at the image level has been proposed for background modeling [96] to introduce spatial information for background subtraction. This approach allows implicit encoding of spatial correlation between pixels and avoids the tiling effect of block partitioning. However, the quality of results significantly depends on the training images of the eigenspace. For instance, when the current frame contains a moving object in the same position as in a training frame, the object will be wrongly classified as background.

2.4.2.3 Shadow detection and removal

Several methods have been proposed to cope explicitly with shadows for background subtraction. The moving shadows can be performed considering many types of information [5]. Basically, the moving shadows can be modeled using properties such as color spaces, photometric models, texture patterns, size, shape, and direction. Furthermore, the methodology of moving cast shadows can include geometrical-information or spatial-cues as well as a training stage, or a combination of two or more of them. The shadow detection also can be based on the level of shadow detection processing, for instance considering only pixel level, region level, or even considering the whole frame.

Many taxonomies are proposed in the literature to classify these methods. Prati *et al.* [104] classify these methods in an algorithm-based taxonomy. As a secondary classification, they categorized these methods into three feature-based categories: *spectral*, *spatial* and *temporal*. Sanin *et al.* [115] have proposed another feature-based taxonomy by dividing spectral features into *intensity*, *chromaticity* and *physical* ones. They compared methods from each category as follows: 1) *color-based methods*, 2) *geometry-based methods*, 3) *physical-based methods*, 4) *texture-based methods*. Since the choice of feature has an important impact on the shadow detection results, the feature-based taxonomy [115] is described in the following paragraphs.

2.4.2.3.1 Color-based methods The color-based methods assume chromaticity preservation in the presence of shadows and, therefore, separate intensity from chromaticity for BS. However, they are sensitive to noise and strong illumination changes.

Cucchiara *et al.* [30] propose a chromatic-based approach that uses the Hue-Saturation-Value (HSV) color space. The main reason for choosing the HSV color space is that it provides a natural separation between chromaticity and luminosity and has proven best than the RGB color space to set a mathematical formulation for shadow detection. Each pixel (x, y) is classified as part of a shadow considering the following three conditions:

$$\beta_1 \leq (F_p^V / B_p^V) \leq \beta_2 \tag{2.60}$$

$$(F_p^S - B_p^S) \leq \tau_S \tag{2.61}$$

$$|F_p^H - B_p^H| \leq \tau_H \tag{2.62}$$

where F and B represent the observed and reference frame respectively, and where x_p^C represent the C-component values of HSV for the pixel (x, y) in the frame X. β_1, β_2, τ_S and τ_H are empirically fixed thresholds.

Horprasert *et al.* [53] introduced a computational color model that can detect shadow regions from the real background or moving foreground objects. This model is based on brightness distortion (BD) and chromaticity distortion (CD), which are defined as follows:

$$BD = \frac{\frac{F_R.\mu_R}{\sigma_R^2} + \frac{F_G.\mu_G}{\sigma_G^2} + \frac{F_B.\mu_B}{\sigma_B^2}}{(\frac{\mu_R}{\sigma_R})^2 + (\frac{\mu_G}{\sigma_G})^2 + (\frac{\mu_B}{\sigma_B})^2}; \tag{2.63}$$

$$CD = \sqrt{\big(\frac{F_R - BD.\mu_R}{\sigma_R}\big)^2 + \big(\frac{F_G - BD.\mu_G}{\sigma_G}\big)^2 + \big(\frac{F_B - BD.\mu_B}{\sigma_B}\big)^2}; \tag{2.64}$$

Where (μ_R, μ_G, μ_B) and $(\sigma_R, \sigma_G, \sigma_B)$ represent the arithmetic means and variance of the red, green, and blue channel values over N background frames. By forcing thresholds on the normalized color distortion (NCD) and normalized brightness distortion (NBD), a pixel is assigned into one of the four categories original background, shaded background, highlight background, and moving foreground objects, by the following decision procedure:

$$\begin{cases} Foreground : NCD > \tau_{CD}\ OR\ NBD < \tau_{\alpha lo},\ else \\ Background : NBD < \tau_{a1}\ AND\ NBD < \tau_{a2},\ else \\ Shadow : NBD < 0,\ else \\ Highlight : otherwise. \end{cases} \tag{2.65}$$

where τ_{CD}, $\tau_{\alpha lo}$, τ_{a1}, and τ_{a2} are selected threshold values used to compute the similarities of the chromaticity and brightness between the background image and the current observed image.

2.4.2.3.2 Physical-based methods The physical-based methods estimate shadow shape and size for objects with specific geometry (e.g., standing people, vehicles, etc.). They also assume a unique and known light source in the scene. In the absence of these assumptions, their success is not guaranteed.

Huang and Chen [55] proposed an unsupervised method that can adapt to illumination conditions or environment changes. For a pixel (x, y), given the vector from shadow to background value denoted as v, the color change is modelled using the 3D color feature $[\alpha, \theta, \phi]$. Where α represents the illumination attenuation while θ and ϕ indicate the direction of v in spherical coordinates. Based on the bi-illuminant (i.e. direct illumination sources and ambient light) dichromatic reflection model, a physics-based color features is developed. The model is learned using the video sequences data in an unsupervised way. Firstly, a weak shadow detector identifies pixels in the foreground that have decreased luminance from that of the background. Then, these candidate shadow pixels are used to update a Gaussian mixture model of the color features, penalizing the learning rate parameter of pixels with higher gradient values than the background, which are more likely to be foreground regions. Finally, each pixel in the foreground is labeled as object or shadow according to the posterior probabilities of the model.

2.4.2.3.3 Geometry-based methods The geometry-based methods learn models for illumination attenuation induced by shadows. However, since they are based on local spectral properties, they can be sensitive to noise.

Hsieh *et al.* [54] proposed a Gaussian shadow model for detecting and eliminating pedestrian shadows from a static background scene. Several features (including the mean intensity, the orientation, and the center position of a shadow region) are used to parameterize the model. Yoneyama *et al.* [153] developed a monitoring system for vehicle and traffic tracking. The joint 2D vehicle-shadow model is employed to eliminate the shadow cast by moving vehicles. The proposed 2D vehicle-shadow models are classified into six types, and the estimation of the parameters of these models can be obtained by fitting the segmented vehicles based on these models.

2.4.2.3.4 Texture-based methods The texture-based methods exploit the fact that shadowed regions can be correlated with the corresponding original image surfaces. These methods try to obtain such a correlation using for example normalized cross correlation (NCC), local binary patterns (LBP), color cross covariant (CCC), Markov random field (MRF) ..., etc. After detecting potential shadows using color/geometric methods, texture can help to validate it. These methods are generally powerful but computationally expensive.

For example, Leone and Distante [71] proposed a shadow detection methodology based on the assumption that shadows are half-transparent regions which maintain the pattern of corresponding reference background. In fact, they use the Gabor filters to characterize the textured patches. A foreground binary mask is generated to evaluate the photometric information for all pixels marked as foreground. For all potential shadow pixels, texture analysis is conducted by projecting the neighborhood of pixels onto a set of Gabor functions, obtained by using the Matching Pursuit strategy. The best matching between the current frame and the reference model can be computed using the Euclidean distance.

Sanin *et al.* [114] performed the shadow removal by using the chromaticity and gradient features. Firstly, shadow pixels are pre-selected based on chromaticity invariance. Then, these pixels are grouped into candidate shadow regions. After that, the shadow regions are determined by the correlation of the gradient direction between the current frame and the reference one.

2.4.2.4 Combined features modeling

When observing the performance of the above methods, one of many promising avenues for improving BS is the combination of several features [22, 107].

For example, in [122] a method has been proposed to recover the contour structure of objects based on image gradients. The approach works well in uniform backgrounds since object contours can be reliably detected. However, in textured or cluttered backgrounds, the accuracy of the method decreases.

In [128], a probabilistic model integrating color, texture and shape cues has been proposed. The algorithm gives accurate results for static backgrounds, but its accuracy decreases in complex background dynamics.

In [107], the authors have used the Gaussian distribution to model spatial and temporal information of videos. This method has the ability to detect mild shadows and deal with background noise. However, its efficiency decreases with non-stationary backgrounds where the Gaussian assumption can be violated.

In [60], the authors have proposed a BS method where local temporal and spatial data are assumed to follow the same distribution. This method is more robust to noise and can guarantee some spatial coherence for foreground detection. However, it is not efficient in dealing with shadows, illumination changes, and complex background dynamics.

2.4.3 BS Evaluation and datasets

2.4.3.1 Evaluation metrics

Many evaluation metrics has been proposed to measure accurately the performance of background subtraction methods to detect foreground objects. For instance, the recall measure favors methods with a low False Negative Rate. On the contrary, specificity measure favors methods with a low False Positive Rate. Finding a compromise is not trivial.

Let TN = number of true negatives, TP = number of true positives, FN = number of false negatives, and FP = number of false positives.

The principle metrics used for background subtraction evaluation are:

- Recall (Re): $TP/(TP + FN)$
- Specificity (Sp): $TN/(TN + FP)$
- False Positive Rate (FPR): $FP/(FP + TN)$
- False Negative Rate (FNR): $FN/(TN + FP)$
- Percentage of Wrong Classifications (PWC): $100(FN + FP)/(TP + FN + FP + TN)$
- Precision (Pr): $TP/(TP + FP)$
- F-measure: $2.Pr.Re/(Pr + Re)$

2.4.3.2 Video foreground segmentation evaluation datasets

The quantitative evaluation, as well as the comparison between background subtraction algorithms, are made by means of datasets. Several datasets have been proposed in the literature [22, 44, 73, 131]. They can be classified in traditional and recent datasets [18]. The traditional datasets covered some challenges, but there is no one of them that addressed all background subtraction challenges. In addition, the available ground truth for these datasets is not provided for the entire video frames. On the other hand, the recent datasets are characterized by large-scale sequences along with accurate pixel-level ground truth for the foreground, background and shadow regions. This enables objective and quantitative evaluation and comparison of the video background subtraction algorithms. The dataset sequences, there ground truth masks, and the algorithm rankings are publicly available and updated on a specified dataset website. A list 20 of those datasets is presented in Table 2.3 and illustrated with a sample frame in Figure 2.12. We describe below a few of the key existing datasets along with their main characteristics.

2.4.3.2.1 Wallflower Dataset The Wallflower dataset was provided by Toyama *et al.* [131] and contains seven sequences (Time of Day, Light Switch, Moved Object, Camouflage, Waving Trees, Bootstrapping, and Foreground Aperture) that represent real-life videos typical of scenes susceptible to meet in video surveillance. Each sequence represents a distinct challenge encountered in background subtraction. The size of the sequences is 160×120 pixels. Each sequence has only one ground-truth foreground mask. Thus, the performance is evaluated against hand-segmented ground truth. This dataset is the most used because it was the first in the field. However, as it provided only one ground-truth image by sequence, its use tends to disappear for the profit of the recent datasets.

2.4.3.2.2 PETS The PETS (Performance Evaluation of Tracking and Surveillance) dataset [155] was introduced to evaluate visual tracking and surveillance approaches. The PETS dataset contains videos for the scientific community since the year 2000 and now includes several dozen videos such as PETS 2001, PETS 2003 and PETS 2006. Since the ground-truth is provided as bounding boxes, this dataset is more adapted for tracking evaluation than for background subtraction.

2.4.3.2.3 I2R Dataset The I2R dataset was provided by Lin and Huang [73], it consists of 9 video sequences, each sequence presents dynamic backgrounds, illumination changes, and bootstrapping challenges. The size of the images is 176×144 pixels. This dataset consists of the following sequences: Curtain, Campus, Lobby, Shopping Mall, Airport, Restaurant, Water Surface, and Fountain. The sequences Curtain, Campus, Water Surface, and Fountain present dynamic backgrounds whereas the sequence Lobby presents sudden illumination changes and the sequences Shopping Mall, Airport and Restaurant show bootstrapping issues. The ground truth is provided for twenty images for each sequence. This dataset is frequently used due to the different kinds of dynamic backgrounds.

2.4.3.2.4 SABS Dataset The SABS (Stuttgart Artificial Background Subtraction) dataset [22] is an artificial dataset for pixel-wise evaluation of background models. The sequences were generated by modern ray tracing which make available high-quality ground-truth data. The sequences have a resolution of 800×600 pixels and are captured from a fixed viewpoint. The dataset includes nine video sequences for different challenges of background subtraction for video surveillance: Basic, Dynamic Background, Bootstrapping, Darkening, Light Switch, Noisy Night, Shadow, Camouflage, and Video Compression. These sequences are further split into training and test data. For every frame of each test sequence, the ground-truth annotation is provided as color-coded foreground masks. Additional shadow masks are provided to indicate the luminance change introduced by foreground objects. Each

challenge includes a sequence of 800 frames for the training phase (with exception of the bootstrapping scenario) and a sequence of 600 frames for the test-phase (except for the Darkening and Bootstrap scenarios that both contain 1400 frames) with full ground truth masks.

2.4.3.2.5 Change Detection Dataset The Change Detection (CD) dataset [44, 142] is a realistic and large-scale video dataset for background subtraction evaluation. The first version of this dataset was introduced in 2012 [44]. It consists of 31 video sequences ($\sim$ 90,000 frames) that represent 6 categories selected to depict various challenges that are common in background subtraction. The following categories are covered by this dataset: Baseline, Dynamic Background, Camera Jitter, Shadows, Intermittent Object Motion, and Thermal. The latest release of the CD dataset (2014) [142] includes 22 additional videos ($\sim$ 70 000 pixel-wise annotated frames) including 5 new categories to deal with additional challenges encountered in real-world surveillance applications. The new categories provided by the CD 2014 dataset are Challenging Weather, Low-Frame-Rate, Night, PTZ, and Air Turbulence. In addition, an online evaluation methodology is provided to help researchers compare their algorithms with the state-of-the-art methods including ranking software, ground-truth masks for all the dataset videos, and online ranking results. This enables objective and accurate comparison and ranking of background subtraction algorithms.

2.5 Conclusion

In this chapter, we give background theory and related topics to finite mixture models (FMM) and foreground segmentation in images and videos. We choose finite mixture models (FMM) as a statistical framework to develop algorithms in this thesis due to the good compromise regarding the processing time, the simplicity of use, and the quality of results. The finite mixture model (FMM) has been defined and illustrated with two concrete examples: namely, the Gaussian mixture model (GMM) and the Mixture of Generalized Gaussians (MoGG) along with the Expectation-Maximization (EM) algorithm which allows to compute the mixture parameters. Moreover, popular algorithms and approaches for foreground segmentation (FS) in still images and video sequences have been presented. The following chapters include our contributions to deal with image and video FS.

	Dataset	Description	Size	Ground truth
1.	Wallflower [131]	Each video represents a specific challenge such as illumination change, background motion, etc.	7 short videos.	Pixel-based labeling of one frame per video.
2.	CAVIAR [24]	Different scenarios of interest. These include people walking, meeting with others, window shopping, entering and exiting shops, but not least, leaving a package in a public place	60 videos.	Bounding boxes.
3.	ATON [104]	Sequences present shadows in indoor and outdoor environments.	5 videos.	The ground-truth is provided for each sequence.
4.	I2R [73]	Videos with illumination changes and dynamic background.	10 short videos.	10 frames/video Pixel-wise labels.
5.	PETS 2001 [155]	Outdoor people and vehicle tracking.	5 videos.	Bounding boxes.
6.	PETS 2002	Indoor people tracking (and counting).	6 sequences.	Bounding boxes.
7.	PETS 2004	People tracking and activity recognition.	28 sequences, 6 scenarios.	Bounding boxes.
8.	PETS 2006	Surveillance of public spaces, detection of left luggage.	7 datasets (4 camera views each one).	Bounding boxes.
9.	PETS 2007	Multisensor sequences containing loitering, attended luggage removal (theft), and unattended luggage.	8 datasets (4 camera views each one).	Bounding boxes.
10.	VSSN 2006 [137]	Semi-synthetic videos composed of a real background and artificially-moving objects. The videos contain animated background, illumination changes and shadows (now included in ViSOR).	9 videos.	Pixel-based labeling of each frame.
11.	i-Lids [57]	Long videos meant for action recognition. Shows parked vehicles, abandoned objects, walking people, doorways.	14 sequences.	Not fully labeled.
12.	ETISEO [95]	Videos meant to evaluate tracking and event detection methods.	More than 80 videos.	High-level labels such as bounding box, object class, event type.
13.	BEHAVE [10]	Videos shot by the same camera showing human interactions such as walking in group, meeting, splitting, etc.	7 short videos.	Bounding boxes.
14.	cVSG [130]	Semi-synthetic videos with various levels of textural complexity, background motion, moving object speed, size and interaction.	15 videos.	Pixel-wise labeling.
15.	UCSD [80]	Videos with background motion and/or camera motion.	18 short videos.	Pixel-wise labeling.
16.	ViSOR [134]	Indoor and outdoor surveillance sequences; annotation data for object detection, tracking, events, etc.	500 short videos.	Bounding boxes.
17.	BMC [133]	Outdoor videos, most being synthetic.	29 sequences.	Pixel-wise labeling for 10 synthetic and 9 real world videos. Ground truth is given for a small subset of frames.
18.	SABS [22]	Computer-generated videos showing a street corner. Includes bootstrapping, illumination changes, dynamic background, shadows, noise, and video compression	9 sequences.	Pixel-wise labeling.
19.	CDnet 2012 [44]	Realistic videos include the following background subtraction challenges: dynamic Background, camera Jitter, shadows, intermittent object motion, and thermal videos.	6 categories, 33 sequences.	Pixel-wise labeling.
20.	CDnet 2014 [142]	In addition to the CDnet 2012 sequences, the 2014 version of this dataset contains videos with the following challenges: weather, low frame-rate, night, PTZ, and air turbulence.	11 categories, 53 sequences.	Pixel-wise labeling.

TABLE 2.3: Existing datasets used to evaluate background subtraction methods.

1. Wallflower 2. CAVIAR 3. ATON 4. I2R

5. PETS 2001 6. PETS 2002 7. PETS 2004 8. PETS 2006

9. PETS 2007 10. VSSN 2006 11. i-Lids 12. ETISEO

13. BEHAVE 14. cVSG 15. UCSD 16. ViSOR

17. BMC 18. SABS 19. CDnet 2012 20. CDnet 2014

FIGURE 2.12: Sample frames for the datasets of Table 2.3.

3

Image Foreground Segmentation Based on Mixture Modelling

3.1 Introduction

In this chapter, we propose a new thresholding approach that performs segmentation for multi-modal classes with arbitrarily-shaped modes. We generalize the aforementioned state-of-art techniques, based on using single probability density functions (pdf's), to mixtures of generalized Gaussian distributions (MoGG's). The Generalized Gaussian Distributions (GGD) is a generalization of the Laplacian and the normal distributions in that it has an additional degree of freedom that controls its kurtosis. Therefore, histogram modes, ranging from sharply peaked to flat ones, can be accurately represented using this model. Furthermore, skewed and multi-modal classes are accurately represented using mixtures of GGDs. We propose an objective function that finds optimal thresholds for multi-modal classes of data. It also extends easily to arbitrary numbers of classes ($K > 2$) with reasonable computational time. Experiments on synthetic data, as well as real-world image segmentation, show the performance of the proposed approach. The proposed approach has been published in the *Int'l Conference on Pattern Recognition (ICPR 2012)* [14] and in the *Pattern Recognition (PR)* journal [15].

This chapter is organized as follows: Section 3.2 presents the Otsu's method and their median extension. In Section 3.3, we outline our proposed approach for image foreground segmentation. Experimental results are given in Section 3.4. We

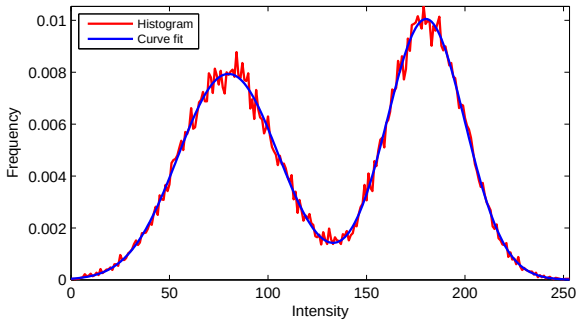


FIGURE 3.1: Bimodal histogram ($K = 2$).

end this chapter with a conclusion.

3.2 General formulation of the Otsu's method (case K=2)

Let $\mathcal{X} = \{x_1, x_2, ..., x_N\}$ be the gray levels of the pixels of an image I of size $N = H \times W$; H and W being the height and the width of the image. Let $\mathbf{t} = (t_1, t_2, ..., t_{K-1})$ be a set of thresholds that partitions an image into K classes. Firstly we consider the simple case of $K = 2$. The most general case of $K > 2$ will be elaborated later in this thesis. In the case of $K = 2$, one threshold t yields two classes $\mathcal{C}_1(t) = \{x : 0 \leq x \leq t\}$ and $\mathcal{C}_2(t) = \{x : t + 1 \leq x \leq T\}$, where T is the maximum gray level. Finally, we denote by $h(x)$ the histogram frequency of the gray level x, where $\sum_{x=0}^{T} h(x) = 1$. The resulting histogram in this case ($K = 2$) is bimodal, as shown in Figure 3.1. Otsu's method [97] determines the optimal threshold t using discriminant analysis, by maximizing inter-class variation, or equivalently minimizing intra-class variation. A generalized formula of the Otsu's method for $K = 2$ can be defined as follows (see refs. [43, 66, 97, 118, 148, 150]):

$$\sigma_B^2(t) = \arg\min_t \{\omega_1(t)V_1(t) + \omega_2(t)V_2(t)\}, \tag{3.1}$$

where, we have:

$$\begin{cases} \omega_1(t) &= \sum_{x=0}^{t} h(x) \\ \omega_2(t) &= \sum_{x=t+1}^{T} h(x) = 1 - \omega_1(t) \end{cases}, \tag{3.2}$$

and

$$\begin{cases} V_1(t) &= \frac{1}{\omega_1(t)} \sum_{x=0}^{t} h(x) \|(x - m_1(t)\|_\lambda \\ V_2(t) &= \frac{1}{\omega_2(t)} \sum_{x=t+1}^{T} h(x) \|(x - m_2(t)\|_\lambda \end{cases}, \tag{3.3}$$

where $\|\cdot\|$ is the norm symbol. When $\lambda = 2$, the model corresponds to the standard Otsu's method and the minimum error thresholding proposed in [66] and [97], respectively. When $\lambda = 1$, the model corresponds to the method proposed in [148]. In the case $\lambda = 2$, the estimated location parameters $m_1(t)$ and $m_2(t)$ correspond to the sample means of the classes $\mathcal{C}_1$ and $\mathcal{C}_2$; whereas, in the second case $\lambda = 1$, these parameters correspond to the sample medians of the classes, as proposed in [148, 150]. For multi-thresholding, since the classical Ostu's method, extensions were proposed to arbitrary number of classes (see the next section). However, all these works assume the classes follow unimodal distributions with their data generally represented by their mean or median parameters.

3.2.1 Standard Otsu's method (case of $\lambda = 2$)

This method, proposed in [97], consists of calculating an optimal threshold t that segments an image into two distinct regions using the following minimization:

$$t = \arg\min_t \{\omega_1(t)s_1^2(t) + \omega_2(t)s_2^2(t)\}, \tag{3.4}$$

where $\omega_1(t)$ and $\omega_2(t)$, defined in Eq. (3.2), correspond to proportions of pixels representing classes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The parameters $s_1(t)$ and $s_2(t)$ represent sample standard deviations for the classes $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively, and are defined as follows:

$$\begin{cases} s_1^2(t) &= \sum_{x=0}^{t} \left[\frac{h(x)}{\omega_1(t)} (x - \overline{x}_1(t))^2 \right] \\ s_2^2(t) &= \sum_{x=t+1}^{T} \left[\frac{h(x)}{\omega_2(t)} (x - \overline{x}_2(t))^2 \right] \end{cases}, \tag{3.5}$$

where $\overline{x}_1(t)$ and $\overline{x}_2(t)$ correspond to the sample means for $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, which are defined as: $\overline{x}_1(t) = \sum_{x=0}^{t} \frac{h(x)}{\omega_1(t)} x$ and $\overline{x}_2(t) = \sum_{x=t+1}^{T} \frac{h(x)}{\omega_2(t)} x$.

It can be shown that the Otsu's method formulation can be derived from maximization of the log-likelihood of a mixture of two Gaussian distributions [68]. Finally, this method can be easily extended to multi-level thresholding. Given that the image histogram contains K unimodal classes (see Figure 3.2), then $K-1$ thresholds are required to separate the classes $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_K$ that correspond to gray level intervals $[0, t_1]$, $[t_{1+1}, t_2]$, ..., $[t_{K-1}, T]$, respectively. More formally, multi-level thresholds $\mathbf{t} = (t_1, ..., t_{K-1})$ are obtained using the following minimization

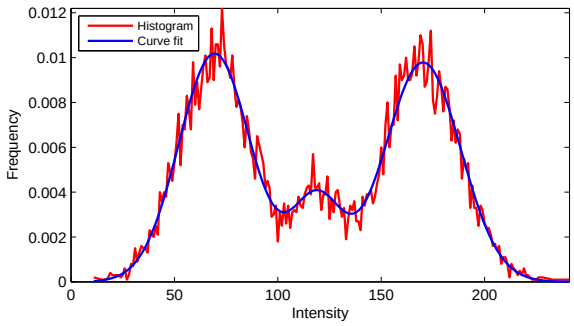


FIGURE 3.2: Multimodal histogram ($K = 3$).

[66, 97, 118]

$$\mathbf{t} = \arg\min_{\mathbf{t}} \sum_{k=1}^{K} \{\omega_k(\mathbf{t}) s_k^2(\mathbf{t})\}, \tag{3.6}$$

where $s_k^2(\mathbf{t}) = \sum_{x=t_{k-1}}^{t_k} \left[\frac{h(x)}{\omega_k(t)}(x - \overline{x}_k(\mathbf{t}))^2\right]$ and $\overline{x}_k(\mathbf{t}) = \sum_{x=t_{k-1}}^{t_k} \frac{h(x)}{\omega_k(\mathbf{t})} x$. Note that Kittler and Illingworth's method [66] uses a slightly modified formula, where an optimal threshold vector $\mathbf{t}$ is selected for arbitrary K, $K \geq 2$, as follows:

$$\mathbf{t} = \arg\min_{\mathbf{t}} \sum_{k=1}^{K} \omega_k(\mathbf{t}) \log \frac{s_k^2(\mathbf{t})}{\omega_k(\mathbf{t})}, \tag{3.7}$$

3.2.2 Median-based extension of Otsu's method (case of $\lambda = 1$)

Recently, Xue *et al.* [148] proposed to use the median parameter instead of the mean in the Otsu's method formulation. For the general case of multi-level thresholding, the optimal threshold $\mathbf{t} = (t_1, ..., t_{K-1})$ can be reached through the following rule:

$$\mathbf{t} = \arg\min_{\mathbf{t}} \sum_{k=1}^{K} \{\omega_k(\mathbf{t}) MAD_k(\mathbf{t})\}, \tag{3.8}$$

where $MAD_k(t)$ represents the absolute deviation from the median for class $C_k(\mathbf{t})$ $(k = 1, .., K)$, which is given by: $MAD_k(\mathbf{t}) = \sum_{x=t_{k-1}}^{t_k} \frac{h(x)}{\omega_k(\mathbf{t})} |(x - m_k(\mathbf{t})|$, where $m_k(\mathbf{t}) = \mathsf{median}\{x \in C_k(\mathbf{t})\}$. Using this formulation, the authors proved that the median-based approach can be derived by modeling each class using the Laplacian distribution. They also showed the robustness of using the median for obtaining

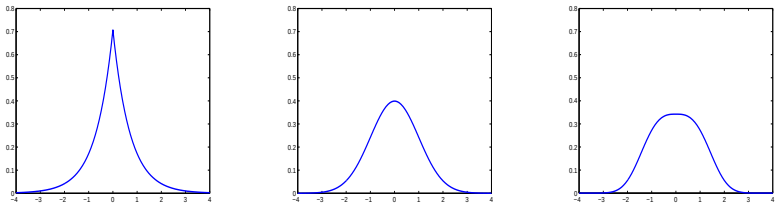
optimal thresholds when class-conditional distributions are skewed or contaminated by outliers. However, when one of the classes contains multiple modes, the models underlying Eq. (3.1), and therefore Eq. (3.8), will not be applicable since it is based on intra-class variation which is built on the assumption that classes are unimodal.

3.3 Multi-modal class thresholding

We propose to extend the general model formulated by Eqs. (3.1) to (3.3) to represent multi-modal class-conditional distributions using finite mixture models (FMM). Finite mixtures are a flexible and powerful probabilistic tool for modeling univariate and multivariate data [38, 86]. They allow for modeling randomly-generated data from multiple sources in an unsupervised fashion. Recently, Allili *et al.* [3] proposed to use finite mixtures of generalized Gaussian distributions (MoGG) to model non-Gaussian data containing multiple classes. Indeed, the generalized Gaussian density (GGD) is an extension of the Gaussian and the Laplacian distributions [20] and has the following formulation:

$$p(x|\mu,\sigma,\lambda) = A(\lambda,\sigma)\exp\big(-B(\lambda)\,|(x-\mu)/\sigma|^{\lambda}\big) \tag{3.9}$$

where $A(\lambda,\sigma) = \lambda\sqrt{\Gamma(3/\lambda)/\Gamma(1/\lambda)}/(2\sigma\Gamma(1/\lambda))$ and $B(\lambda) = [\Gamma(3/\lambda)/\Gamma(1/\lambda)]^{\lambda/2}$; $\Gamma(.)$ being the gamma function. The parameters μ and σ are the GGD location and dispersion parameters. The parameter λ controls the kurtosis of the pdf and determines whether it is peaked or flat: the larger the value of λ, the flatter the pdf; and the smaller λ is, the more peaked the pdf. This gives the pdf a flexibility to fit the shape of heavy-tailed data [20]. Two well-known special cases of the GGD model are the Laplacian, as $\lambda = 1$, and the Gaussian distribution, as $\lambda = 2$ (see Figure 3.3).



$\lambda = 1$ $\lambda = 2$ $\lambda = 3$

FIGURE 3.3: Different shapes of the GGD distribution as a function of the parameter λ.

3.3.1 Multimodal class thresholding: Case of $K = 2$

Here, we suppose that the data consist of two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ separated by a candidate threshold t, each of which is multi-modal and modeled using a MoGG, as follows:

$$\begin{cases} p(x|\mathcal{C}_1) = \sum_{k=1}^{K_1} \alpha_k(t) p(x|\vec{\theta}_k(t)), & (3.10) \\ p(x|\mathcal{C}_2) = \sum_{j=1}^{K_2} \pi_j(t) p(x|\vec{\varphi}_j(t)), & (3.11) \end{cases}$$

where $\alpha_k(t)$, $k \in \{1, ..., K_1\}$ and $\pi_j(t)$, $j \in \{1, ..., K_2\}$, are the mixing parameters of the two mixtures, such that $\sum_{k=1}^{K_1} \alpha_k(t) = 1$ and $\sum_{j=1}^{K_2} \pi_j(t) = 1$. The number of components K_1 and K_2 are application specific (e.g., segmentation of the brain cortex in medical images, foregournd/background segmentation, etc.). Each component in Eqs. (3.10) and (3.11), $p(x|\vec{\theta}_k(t))$ and $p(x|\vec{\varphi}_j(t))$ is a GGD. Using our mixture modelling on the image data $\mathcal{X}$, Eq. (3.1) can be expressed as a maximization of class-likelihoods, as follows:

$$t = \arg\max_t \left\{ \omega_1(t) L_1(t) + \omega_2(t) L_2(t) \right\}, \quad (3.12)$$

where:

$$\begin{cases} L_1(t) = \prod_{0 \le x_i \le t} \left\{ \sum_{k=1}^{K_1} \alpha_k(t) p(x_i|\vec{\theta}_k(t)) \right\}, & (3.13) \\ L_2(t) = \prod_{t+1 \le x_i \le T} \left\{ \sum_{j=1}^{K_2} \pi_j(t) p(x_i|\vec{\varphi}_j(t)) \right\}. & (3.14) \end{cases}$$

By taking the logarithm of each term in Eq. (3.12) and writing the results in term of gray level frequencies, we obtain:

$$\begin{cases} \log(\omega_1(t) L_1(t)) = N \sum_{x=0}^{t} h(x) \log\left\{ \omega_1(t) \sum_{k=1}^{K_1} \alpha_k(t) p(x|\vec{\theta}_k(t)) \right\}, & (3.15) \\ \log(\omega_2(t) L_2(t)) = N \sum_{x=t+1}^{T} h(x) \log\left\{ \omega_2(t) \sum_{j=1}^{K_2} \pi_j(t) p(x|\vec{\varphi}_j(t)) \right\}. & (3.16) \end{cases}$$

Finally, the parameters $(\alpha_k(t), \vec{\theta}_k(t))$, $k \in \{1, ..., K_1\}$, and $(\pi_j(t), \vec{\varphi}_j(t))$, $j \in \{1, ..., K_2\}$, are estimated using the maximum likelihood method. We use the

Expectation-Maximization (EM) algorithm to obtain the parameters of the mixture of generalized Gaussian distributions, as proposed in [3] (we refer the reader to that reference for the derivation of the E-M steps).

3.3.2 Multimodal class thresholding: Case of $K > 2$

When there are more than two classes in the image, one can readily generalize the two-class case model of Eq. (3.12) using a vector of thresholds $\mathbf{t} = \{t_1, ..., t_{K-1}\}$. Therefore, the maximization in Eq. (3.12) will be reformulated as follows:

$$\mathbf{t} = \arg\max_{\mathbf{t}} \left\{ \sum_{r=1}^{K} \omega_r(\mathbf{t}) L_r(\mathbf{t}) \right\}, \tag{3.17}$$

where, we have:

$$L_r(\mathbf{t}) = \prod_{a_r \leq x_i \leq b_r} \left\{ \sum_{k=1}^{K_r} \alpha_{r,k}(\mathbf{t}) p(x_i | \vec{\theta}_{r,k}(\mathbf{t})) \right\}, \tag{3.18}$$

and

$$a_r = \begin{cases} 0 & \text{if } r = 1 \\ t_{r-1} & \text{if } r > 1 \end{cases} \tag{3.19}$$

$$b_r = \begin{cases} T & \text{if } r = K-1 \\ t_r & \text{if } r < K-1 \end{cases} \tag{3.20}$$

After taking the logarithm of each class-likelihood and writing the result in terms of gray level frequencies, we obtain:

$$\log(\omega_r(\mathbf{t}) L_r(\mathbf{t})) = N \sum_{x=a_r}^{b_r} h(x) \log \left\{ \omega_r(\mathbf{t}) \sum_{k=1}^{K_r} \alpha_{r,k}(\mathbf{t}) p(x | \vec{\theta}_{r,k}(\mathbf{t})) \right\}. \tag{3.21}$$

The advantage of using Eq. (3.17) is that it can segment classes with multi-modal distributions, which is impossible to achieve using the thresholding formulation given by Eq. (3.1). In fact, Eq. (3.1) can be obtained as a special case of Eq. (3.17) where each class is medeled as a mixture of a single GGD component with unit variance. Note that this advantage comes with additional time cost to estimating a mixture model parameters for each class using the maximum likelihood method (MLE). Nonetheless, this additional time is not a limitation for our approach given the high thresholding accuracy and flexibility it offers.

The procedure for estimating the optimal threshold vector $\mathbf{t}$ calculates iteratively for each threshold t_r ($r = 1, ..., K-1$). For each threshold t_r, we consider a

left class $\mathcal{C}^r_{left}$ and a right class $\mathcal{C}^r_{right}$, represented each by a mixture of K^r_{left} and K^r_{right} GGDs, respectively. The maximum likelihood estimation is then performed for the parameters of $\mathcal{C}^r_{left}$ and $\mathcal{C}^r_{right}$ using the Expectation-Maximization (EM) algorithm [3]. Note that to speed up the EM estimation for a candidate threshold t, we use the estimated parameters obtained for $t-1$ as initialization. Thus, a small number of iterations is required to adjust the MoGG parameters for each tested threshold. The different steps of the method for estimating the elements of $\mathbf{t}$ are explained in the script of Algorithm 2.

Algorithm 2: Compute the threshold $\mathbf{t} = \{t_1, ..., t_{K-1}\}$ using Eq.(3.17).

Data: Image histogram, Number of classes K, Number of components of each class $\{K_i, i = 1, ...K\}$.
Result: Threshold vector $\mathbf{t} = \{t_1, ..., t_{K-1}\}$.

```
// the first threshold
```
$r \leftarrow 1$;
while $r \leq K-1$ **do**
  $K^r_{left} \leftarrow \sum_{i=1}^{r} K_i$;
  $K^r_{right} \leftarrow \sum_{j=r+1}^{K} K_j$;
  if $(r = 1)$ **then**
    `// minimum gray level +1`
    $s_1 \leftarrow 1$;
  else
    `// maximum gray level −1`
    $s_1 \leftarrow t_{r-1}$;
  end
  `// maximum gray level −1`
  $s_2 \leftarrow T - 1$;
  for $t = s_1 \rightarrow s_2$ **do**
    Calculate $\omega_1(t)$ and $\omega_2(t)$;
    Estimate the parameters of a mixture of K_{left} GGDs (class $\mathcal{C}_{left}$);
    Estimate the parameters of a mixture of K_{right} GGDs (class $\mathcal{C}_{right}$);
    Calculate $J_r(t) = \log(\omega_1(t)L_1(t)) + \log(\omega_2(t)L_2(t))$;
  end
  $t_r \leftarrow \arg\min_t J_r(t)$;
  `// the next threshold`
  $r \leftarrow r + 1$;
end

3.4 Experimental results

We conducted several experiments to measure the performance of the proposed approach by comparing it to recent stat-of-the-art thresholding methods. For this purpose, we used synthetic histograms as well as real images from known datasets [118] (all used datasets can be downloaded at [1]). Quantitative results are presented showing how well the proposed model finds optimal thresholds in terms of segmentation accuracy.

We objectively measure thresholding performance by using Misclassification Error (ME) criterion. For foreground/background image segmentation, the ME reflects the percentage of misclassified pixels, expressed as follows:

$$ME = 1 - \frac{|B_O \cap B_T| + |F_O \cap F_T|}{|B_O| + |F_O|}, \tag{3.22}$$

where B_O and F_O denote, respectively, the background and foreground of the original ground-truth image. B_T and F_T denote, respectively, the background and foreground pixels in the segmented image, where $|.|$ denotes set cardinality. The ME varies from 0, for a perfectly segmented image, to 1, for a totally wrongly segmented image.

The compared methods include: *i*) The standard Otsu's method [97], *ii*) The median-based Otsu's extension method [148], *iii*) Thresholding based on MoGs (Mixture of Gaussians) and *iv*) Our thresholding method based on MoGGs. Note that the difference between MoG and MoGG methods is that for the MoGG method, the shape parameter λ is estimated using the EM algorithm (see algorithm 2), while for the MoG method, the shape parameter is fixed to the Gaussian distribution (i.e., $\lambda = 2$). We compute the ME for all compared methods and rewrite for each method Eqs. (3.6), (3.8) and (3.17) as follows:

$$\mathbf{t}_O^* = \arg\min_{\mathbf{t}} J_O(\mathbf{t}), \tag{3.23}$$

$$\mathbf{t}_M^* = \arg\min_{\mathbf{t}} J_M(\mathbf{t}), \tag{3.24}$$

$$\mathbf{t}_G^* = \arg\min_{\mathbf{t}} J_G(\mathbf{t}), \tag{3.25}$$

$$\mathbf{t}_{GG}^* = \arg\min_{\mathbf{t}} J_{GG}(\mathbf{t}), \tag{3.26}$$

where $J_O(\mathbf{t})$ and $J_M(\mathbf{t})$ (standard Otsus's and median-based methods) are the terms to be minimized on the right-hand sides of Eqs. (3.6) and (3.8), $J_G(\mathbf{t})$ and $J_{GG}(\mathbf{t})$ (MoG and MoGG methods) are the terms to be maximized on the right-hand side of Eq. (3.17). To have better visualization for performance comparison,

[1] http://w3.uqo.ca/allimo01/doc/ThreshDataTest.rar

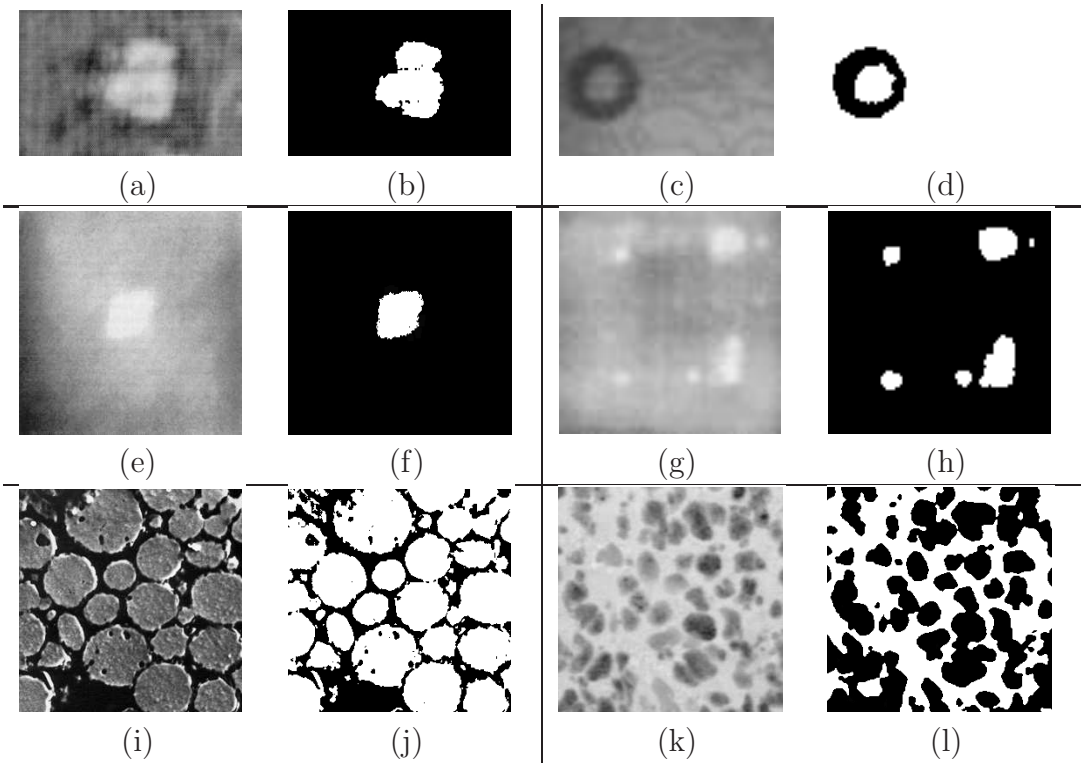


FIGURE 3.4: A sample of NDT-images and their ground truth segmentation: (a,b): ultrasonic GFRP material, (c,d): eddy current image, (e,f): thermal image of GFRP composite material, (g,h): defective eddy current image, (i,j): light microscopic image of a material structure and (k,l): material structure image.

all values of the functions $J_O(\mathbf{t})$, $J_M(\mathbf{t})$, $J_G(\mathbf{t})$ and $J_{GG}(\mathbf{t})$ were re-scaled to the range $[0, 1]$.

3.4.1 Real-world image segmentation

We used the NDT-image dataset (NDT: Non Destructive Testing images) [118], which has also been used to evaluate some popular thresholding methods (e.g., [113, 141, 148]). The main properties of this dataset is that, in the one hand, it includes a variety of real-life thresholding applications like document image analysis, quality inspection of materials, defect detection, cell images, eddy current images, etc. In the other hand, it contains for each NDT-image the corresponding ground truth that considerably facilitates the evaluation task. Figure 3.4 shows a sample of 6 images from this set that contains altogether 25 images.

Tests conducted on the NDT image dataset show that both MoGG and MoG methods give better results in terms of ME values against the standard Otsu's

and the Median extension methods. Table 3.1 shows results obtained using the compared methods. We can note that on average, both MoG and MoGG have a lesser ME than the standard Otsu's and the Median extension methods. This performance of mixture-based methods is justified by their flexibility to represent multi-modal classes better than using class mean or median parameters used in the classical methods. Finally, we can note that thresholds given by the MoGG method are better than those obtained by the MoG method. This can be seen especially in histograms containing non-Gaussian modes, as will be illustrated in the following examples.

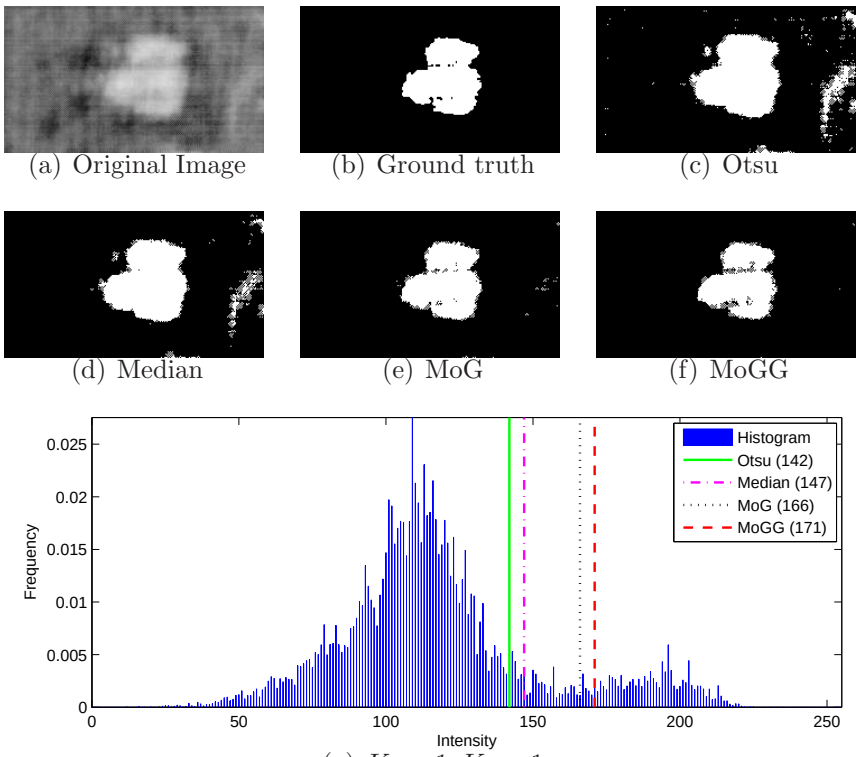
Figures 3.5 and 3.6 show two examples of image segmentation (NDT-1 and NDT-8) using the compared methods. For each example, we show the segmentation ground truth, the image histogram with thresholds $\mathbf{t}_O^*, \mathbf{t}_M^*, \mathbf{t}_G^*$ and $\mathbf{t}_{GG}^*$, the plots for $J_O(\mathbf{t}), J_M(\mathbf{t}), J_G(\mathbf{t})$ and $J_{GG}(\mathbf{t})$, and the binarized image. For each example, the histogram should be separated into two classes (forground and background). In the first example, the classes are approximately Gaussian, and all compared methods gave reasonable thresholds. However, using mixture models gave slightly better segmentations than using standard Otsu's and median extension methods. Finally, using MoGG gave slightly better segmentation than using MoG. In the second example, the classes are multi-modal and the mode shapes range from sharply peaked to flat ones. The standard Otsu's and median extension methods diverged and gave erroneous segmentations. For the mixture-based methods, the MoGG gave a better segmentation than the MoG.

	Classical Modeling		Mixture Modeling	
	Standard Otsu's	Median Extension	MoG	MoGG
AVG	0.1767	0.2315	0.0149	0.0115
STD	0.2103	0.2389	0.0137	0.0106
MIN	0.0003	0.0003	0.0001	0.0001
MAX	0.6261	0.6085	0.0486	0.0381

TABLE 3.1: Misclassification Errors obtained for the NDT-images. Columns from left to right show: the standard Otsu's, the median extension, the MoG and the MoGG methods, respectively. Rows from top to bottom show, respectively, the average, the standard deviation, the minimum and the maximum values of ME obtained by each method.

3.4.2 Simulated datasets

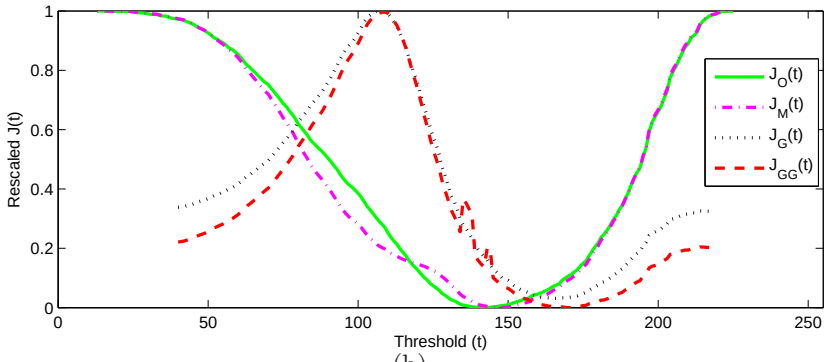
In this experiment, we used 5 benchmarks of randomly generated data. Each benchmark contains 100 datasets, and each dataset contains 10,000 data samples



(a) Original Image (b) Ground truth (c) Otsu

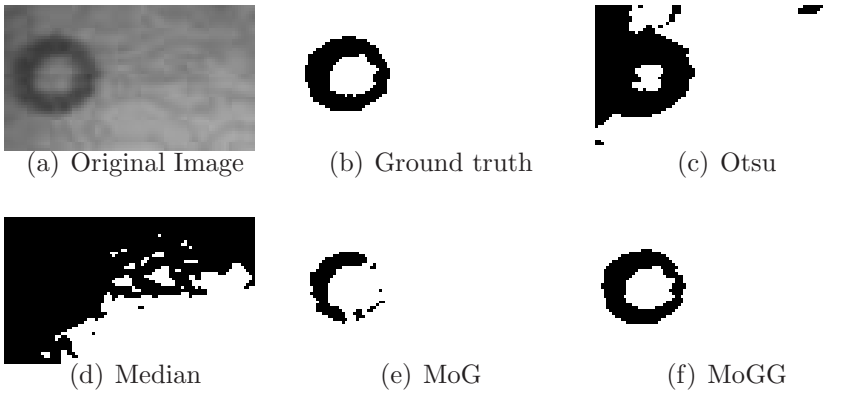
(d) Median (e) MoG (f) MoGG

(g) $K_1 = 1, K_2 = 1$



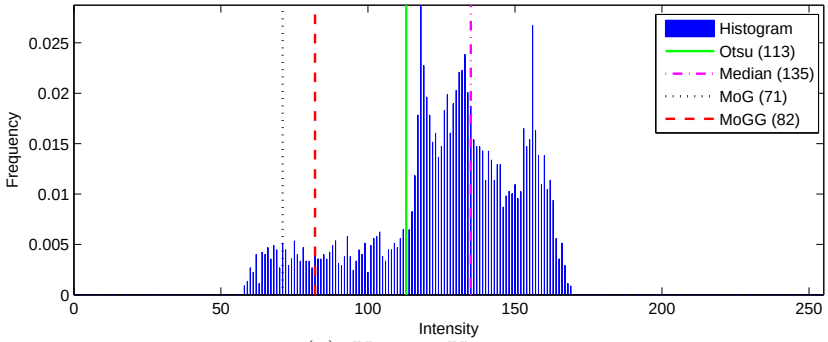
(h)

FIGURE 3.5: For 'NDT-image1' we present: (a) the original image, (b) the ground truth, (c)-(f) segmentation results obtained by the standard Otsu's, the median extension, MoG and MoGG methods, respectively, (g) image histogram superimposed with optimal thresholds ($\mathbf{t}_O^*$, $\mathbf{t}_M^*$, $\mathbf{t}_G^*$ and $\mathbf{t}_{GG}^*$) and (h) plots of functions $J_O^*(\mathbf{t})$, $J_M^*(\mathbf{t})$, $J_G^*(\mathbf{t})$ and $J_{GG}^*(\mathbf{t})$ corresponding to optimal thresholds in the above histogram.

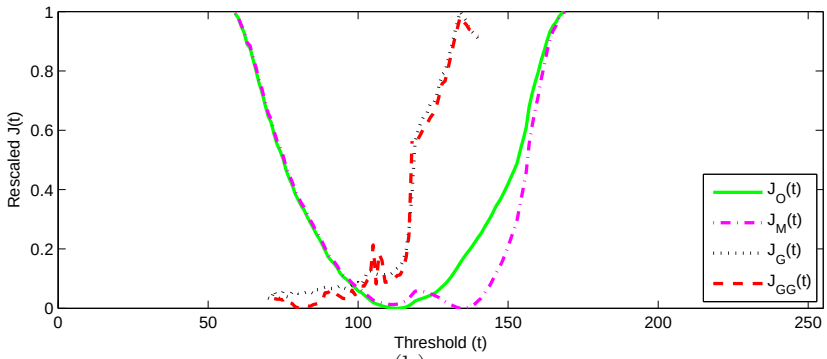


(a) Original Image (b) Ground truth (c) Otsu

(d) Median (e) MoG (f) MoGG



(g) $K_1 = 1, K_2 = 5$



(h)

FIGURE 3.6: For 'NDT-image8'. Caption is as for Figure 3.5.

which correspond to gray levels. Each dataset is constituted of two classes ($\mathcal{C}_1$ for $r = 1$ and $\mathcal{C}_2$ for $r = 2$), where each class is modeled using a mixture of GGDs. The parameter setting of our generated benchmarks is presented in the Table 3.2. Our tests on the simulated datasets are conducted in two phases. In the first phase, the standard Otsu's and the median extension methods are compared with mixture-based methods (MoG and MoGG methods). In the second phase, we compared between the MoG and MoGG methods.

3.4.2.1 Otsu's based methods versus mixture methods

Conducted tests on our simulated datasets show that mixture-based models are very efficient against the standard Otsu's and the median-based extension methods. The key point of the mixture methods (MoG and MoGG) is that they consider the two classes (class $\mathcal{C}_{left}$ and class $\mathcal{C}_{right}$) follow a mixture of K_{left} and K_{right} components, respectively. In the case of the MoG model, distribution components are Gaussians and for the MoGG model, components are considered as generalized Gaussians. The standard Otsu's and Median-based extension methods consider that each class $\mathcal{C}_k$ follows a unimodal distribution, and therefore, fail to get the optimal threshold when the classes are multi-modal. To illustrate these facts, Figure 3.7 shows two examples of generated datasets and thresholds obtained using the standard Otsu's, median-based extension, MoG and MoGG methods, respectively. Finally, Table 3.3 summarizes results obtained by the application of the four methods. It contains for each benchmark the average misclassification error (ME) of each method and, between brackets, the percentage of data sets where MoGG gave the least error. For the majority of the data sets, results show that MoGG method has a lesser ME than the other methods. This clearly demonstrates the performance of our approach.

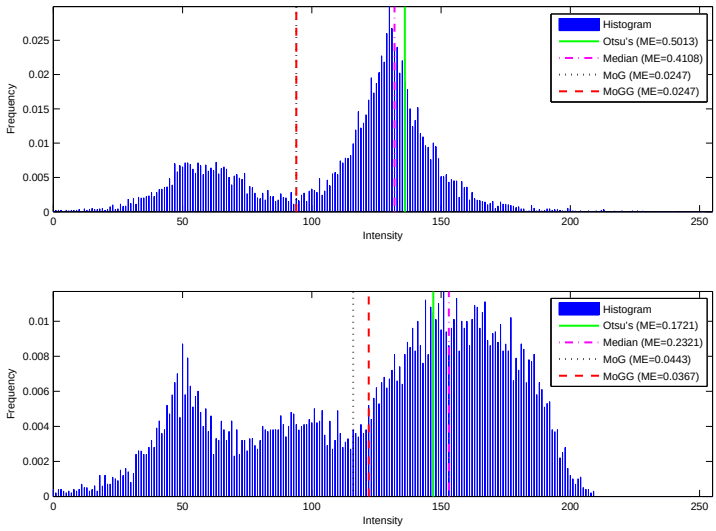


FIGURE 3.7: Examples of generated data sets histograms with optimal thresholding results and misclassification errors (ME) given by different methods (original Otsu's method, median based extension method, MoG method and MoGG method.)

Bench.	Class	K_r	μ	σ	λ	π
Bench.1	$r=1$	2	$\mu_{1,1}=50$	$\sigma_{1,1}=20$	$\lambda_{1,1}=1.00$	$\pi_{1,1}=0.40$
			$\mu_{1,2}=100$	$\sigma_{1,2}=20$	$\lambda_{1,2}=2.00$	$\pi_{1,2}=0.40$
	$r=2$	2	$\mu_{2,1}=150$	$\sigma_{2,1}=20$	$\lambda_{2,1}=4.00$	$\pi_{2,1}=0.10$
			$\mu_{2,2}=150$	$\sigma_{2,2}=20$	$\lambda_{2,2}=4.00$	$\pi_{2,2}=0.10$
Bench.2	$r=1$	2	$\mu_{1,1}=70$	$\sigma_{1,1}=20$	$\lambda_{1,1}=1.00$	$\pi_{1,1}=0.10$
			$\mu_{1,2}=70$	$\sigma_{1,2}=20$	$\lambda_{1,2}=2.00$	$\pi_{1,2}=0.10$
	$r=2$	2	$\mu_{2,1}=120$	$\sigma_{2,1}=20$	$\lambda_{2,1}=1.00$	$\pi_{2,1}=0.20$
			$\mu_{2,2}=170$	$\sigma_{2,2}=20$	$\lambda_{2,2}=4.00$	$\pi_{2,2}=0.60$
Bench.3	$r=1$	2	$\mu_{1,1}=50$	$\sigma_{1,1}=20$	$\lambda_{1,1}=1.00$	$\pi_{1,1}=0.10$
			$\mu_{1,2}=50$	$\sigma_{1,2}=20$	$\lambda_{1,2}=2.00$	$\pi_{1,2}=0.10$
	$r=2$	2	$\mu_{2,1}=100$	$\sigma_{2,1}=20$	$\lambda_{2,1}=2.00$	$\pi_{2,1}=0.20$
			$\mu_{2,2}=160$	$\sigma_{2,2}=20$	$\lambda_{2,2}=4.00$	$\pi_{2,2}=0.60$
Bench.4	$r=1$	2	$\mu_{1,1}=100$	$\sigma_{1,1}=20$	$\lambda_{1,1}=1.00$	$\pi_{1,1}=0.20$
			$\mu_{1,2}=100$	$\sigma_{1,2}=20$	$\lambda_{1,2}=4.00$	$\pi_{1,2}=0.10$
	$r=2$	2	$\mu_{2,1}=150$	$\sigma_{2,1}=30$	$\lambda_{2,1}=4.00$	$\pi_{2,1}=0.30$
			$\mu_{2,2}=180$	$\sigma_{2,2}=20$	$\lambda_{2,2}=1.00$	$\pi_{2,2}=0.40$
Bench.5	$r=1$	2	$\mu_{1,1}=60$	$\sigma_{1,1}=20$	$\lambda_{1,1}=2.00$	$\pi_{1,1}=0.05$
			$\mu_{1,2}=70$	$\sigma_{1,2}=10$	$\lambda_{1,2}=1.00$	$\pi_{1,2}=0.05$
	$r=2$	3	$\mu_{2,1}=100$	$\sigma_{2,1}=30$	$\lambda_{2,1}=4.00$	$\pi_{2,1}=0.20$
			$\mu_{2,2}=150$	$\sigma_{2,2}=20$	$\lambda_{2,2}=2.00$	$\pi_{2,2}=0.35$
			$\mu_{2,3}=170$	$\sigma_{2,3}=20$	$\lambda_{2,3}=1.00$	$\pi_{2,3}=0.35$

TABLE 3.2: Parameters setting for bimodal simulated dataset benchmarks.

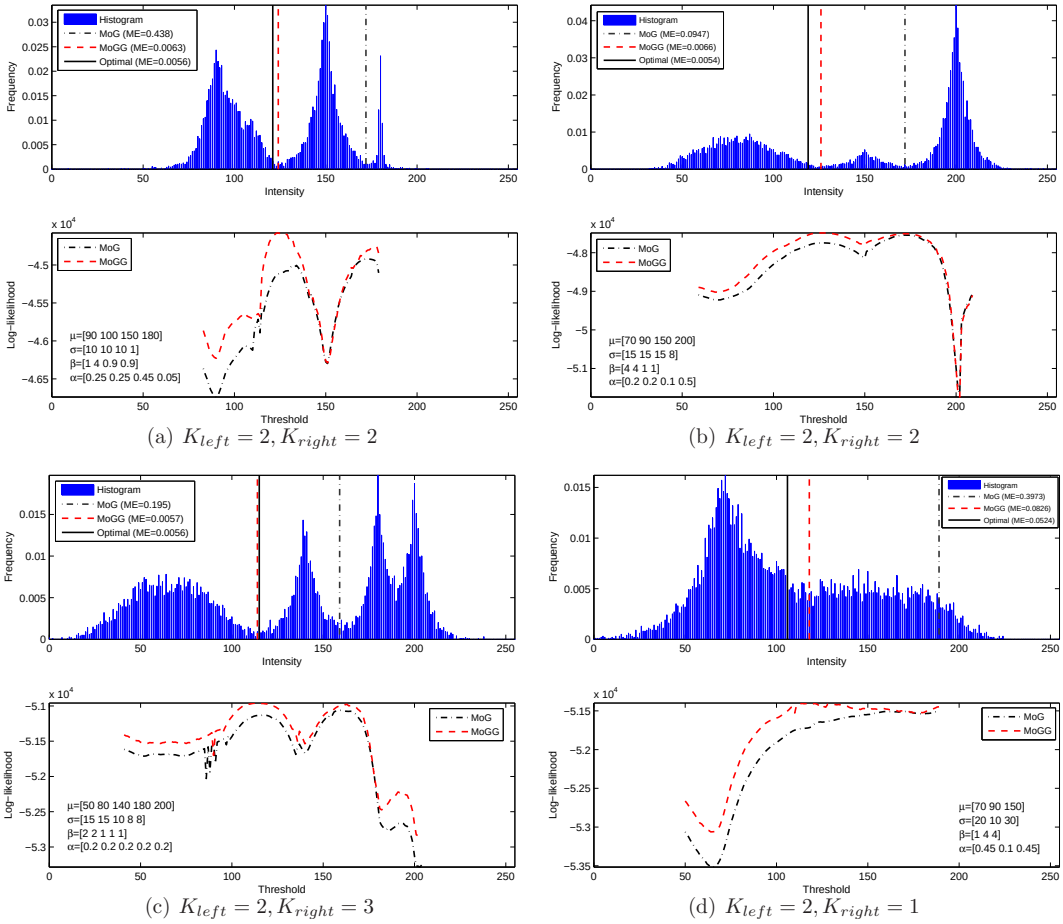
Bench.	Error Otsu	Error Median	Error MoG	Error MoGG
1	0.188 (100%)	0.182 (100%)	0.126 (97%)	0.093
2	0.187 (100%)	0.199 (100%)	0.086 (89%)	0.035
3	0.168 (99%)	0.173 (100%)	0.152 (85%)	0.062
4	0.168 (67%)	0.159 (63%)	0.190 (80%)	0.146
5	0.181 (100%)	0.232 (100%)	0.124 (92%)	0.089

TABLE 3.3: Classification error in the simulated data sets (Between brackets is the percentage of data sets where MoGG gives the least error).

3.4.2.2 MoG versus MoGG thresholding

This experiment aims to compare MoGG against MoG models for thresholding. Similarly to the experiment performed in [86] (see Chapter 1, page 12), three types of mixtures densities we generated, namely: *bimodal*, *trimodal* and *multimodal* distributions, using GGD components instead of Gaussian distributions. Figure 4.2 shows 4 examples of these histograms and the applied thresholding using MoG and MoGG models. These examples were chosen particularly because they contain modes with non-Gaussian shapes. The number of components K_{left} and K_{right} used for thresholding are those used in the sampling step. We can clearly note the improvement in terms of log-likelihood fitting by using MoGG method against the MoG method. In terms of thresholding, using MoG gave lesser performance than using MoGG in all the chosen examples. For instance, in the second example (top right), the right class (modeled by 2 Gaussians) was fitted completely to the rightmost mode which is very sharp, whereas the MoGG used only one component for this mode, yielding finally to a nearly-optimal threshold.

To emphasize our comparison, we generated 3 benchmarks of 100 datasets. Each dataset contains 10,000 data samples and is constituted of 2 classes generated using two MoGGs. Parameter setting of our generated datasets is presented in the Table 3.4. Table 3.5 shows classification errors obtained using the MoG and MoGG models, respectively. We can note that in the 3 benchmarks, using MoGG gives better thresholds than using MoG. Figures 3.9 to 3.11 show different examples where MoGG modeling was more efficient than using MoG's for thresholding. The third and fourth rows of each figure depict the histogram of each class ($\mathcal{C}_1$ and $\mathcal{C}_2$), the thresholds obtained by the application of MoG and MoGG methods and the plots of estimated mixture distribution fit of each class, respectively.



(a) $K_{left} = 2, K_{right} = 2$ (b) $K_{left} = 2, K_{right} = 2$

(c) $K_{left} = 2, K_{right} = 3$ (d) $K_{left} = 2, K_{right} = 1$

FIGURE 3.8: Optimal thresholds, misclassification error and log-likelihoods obtained using MoG and MoGG methods for 4 synthetic generalized Gaussian mixture densities inspired from Gaussian mixture densities given by [86] (Chapter 1, page 12).

Bench.	Error MoG	Error MoGG
1	0.231 (85%)	0.081
2	0.344 (100%)	0.069
3	0.240 (100%)	0.010

TABLE 3.5: Classification error in the simulated data sets of Table 3.4 benchmarks (Between brackets is the percentage of data sets where MoGG gives the least error).

In Figure 3.9, a two-class histogram is generated using the parameter setting of the benchmark 1 of Table 3.4 ($K_{left} = 2$, $K_{right} = 2$). We can observe that the MoGG method gave a better threshold $\simeq 101$ ($ME = 0.0616$) against the MoG

Bench.	Class	K_r	μ	σ	λ	π
Bench.1	$r=1$	2	$\mu_{1,1}=70$	$\sigma_{1,1}=15$	$\lambda_{1,1}=4.00$	$\pi_{1,1}=0.15$
			$\mu_{1,2}=90$	$\sigma_{1,2}=15$	$\lambda_{1,2}=4.00$	$\pi_{1,2}=0.15$
	$r=2$	2	$\mu_{2,1}=120$	$\sigma_{2,1}=15$	$\lambda_{2,1}=1.00$	$\pi_{2,1}=0.30$
			$\mu_{2,2}=140$	$\sigma_{2,2}=8$	$\lambda_{2,2}=1.00$	$\pi_{2,2}=0.40$
Bench.2	$r=1$	2	$\mu_{1,1}=70$	$\sigma_{1,1}=25$	$\lambda_{1,1}=4.00$	$\pi_{1,1}=0.30$
			$\mu_{1,2}=90$	$\sigma_{1,2}=15$	$\lambda_{1,2}=4.00$	$\pi_{1,2}=0.10$
	$r=2$	2	$\mu_{2,1}=120$	$\sigma_{2,1}=15$	$\lambda_{2,1}=1.00$	$\pi_{2,1}=0.20$
			$\mu_{2,2}=130$	$\sigma_{2,2}=8$	$\lambda_{2,2}=1.00$	$\pi_{2,2}=0.40$
Bench.3	$r=1$	2	$\mu_{1,1}=80$	$\sigma_{1,1}=10$	$\lambda_{1,1}=4.00$	$\pi_{1,1}=0.25$
			$\mu_{1,2}=100$	$\sigma_{1,2}=10$	$\lambda_{1,2}=4.00$	$\pi_{1,2}=0.25$
	$r=2$	2	$\mu_{2,1}=150$	$\sigma_{2,1}=15$	$\lambda_{2,1}=1.00$	$\pi_{2,1}=0.25$
			$\mu_{2,2}=200$	$\sigma_{2,2}=8$	$\lambda_{2,2}=1.00$	$\pi_{2,2}=0.25$

TABLE 3.4: Parameters setting for bimodal simulated dataset benchmarks (MoG Versus MoGG experimentation).

method which gives a threshold $\simeq 128$ ($ME = 0.2457$). This result is due to the non-Gaussian type of component distributions (see for instance the modes of the class in the right side). The right MoG model assigned its two components to the right mode. Consequently, it was not able to find the correct threshold between the classes $\mathcal{C}_1$ and $\mathcal{C}_2$ (see the graph on the bottom of the figure). The MoGG gave a nearly-optimal threshold where the right MoGG model assigned its two components to the two rightmost modes of the histogram.

Another important fact that we have observed in the results, especially the benchmark 2, is the divergence of the MoG method for some examples. In Figure 3.10, we show an example in benchmark 2 illustrating this fact. We can note that for the $J_{GG}(t)$ function, the global minimum is close to the optimal threshold $\simeq 100$ ($ME = 0.0797$), whereas for the $J_G(t)$ function, the global minimum is far from the optimal threshold $\simeq 43$ ($ME = 0.3458$). This problem of MoG-based thresholding is due to its inefficiency to fit accurately histograms which are constituted of non-Gaussian components (see the graph on the bottom of the figure). The MoGG model adequately fitted the non-Gaussian histogram modes in the left and right classes, and, consequently, gave a better threshold.

Finally, in Figure 3.11, the left and right classes contain three and two modes, respectively. After computing the optimal thresholds using the two mixture models, the MoG method failed to place the optimal threshold in the true position $t_1 \simeq 177$ ($ME = 0.2421$). In fact, two Gaussian components were necessary to fit one non-Gaussian histogram mode in the right class. The MoGG method successfully fitted each mode adequately with one mixture GGD component, giving,

therefore, a better threshold $t_1 \simeq 123$ ($ME = 0.012$). These examples demonstrate the performance of using MoGG models instead of MoG models for thresholding.

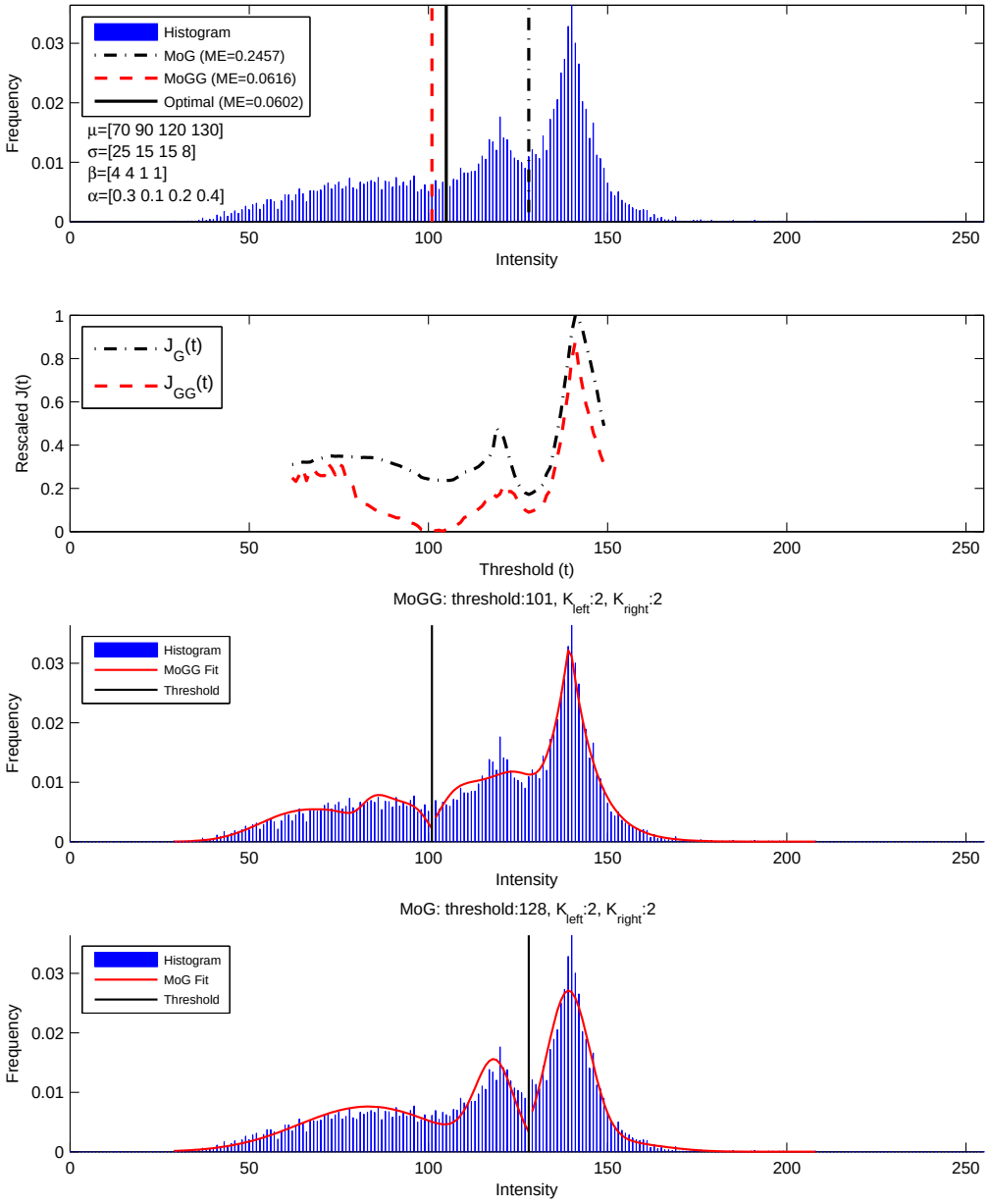


FIGURE 3.9: Example of a generated data set histogram from the benchmark 1 of Table 3.5 with optimal thresholding results and misclassification errors (ME) given by MoG and MoGG methods.

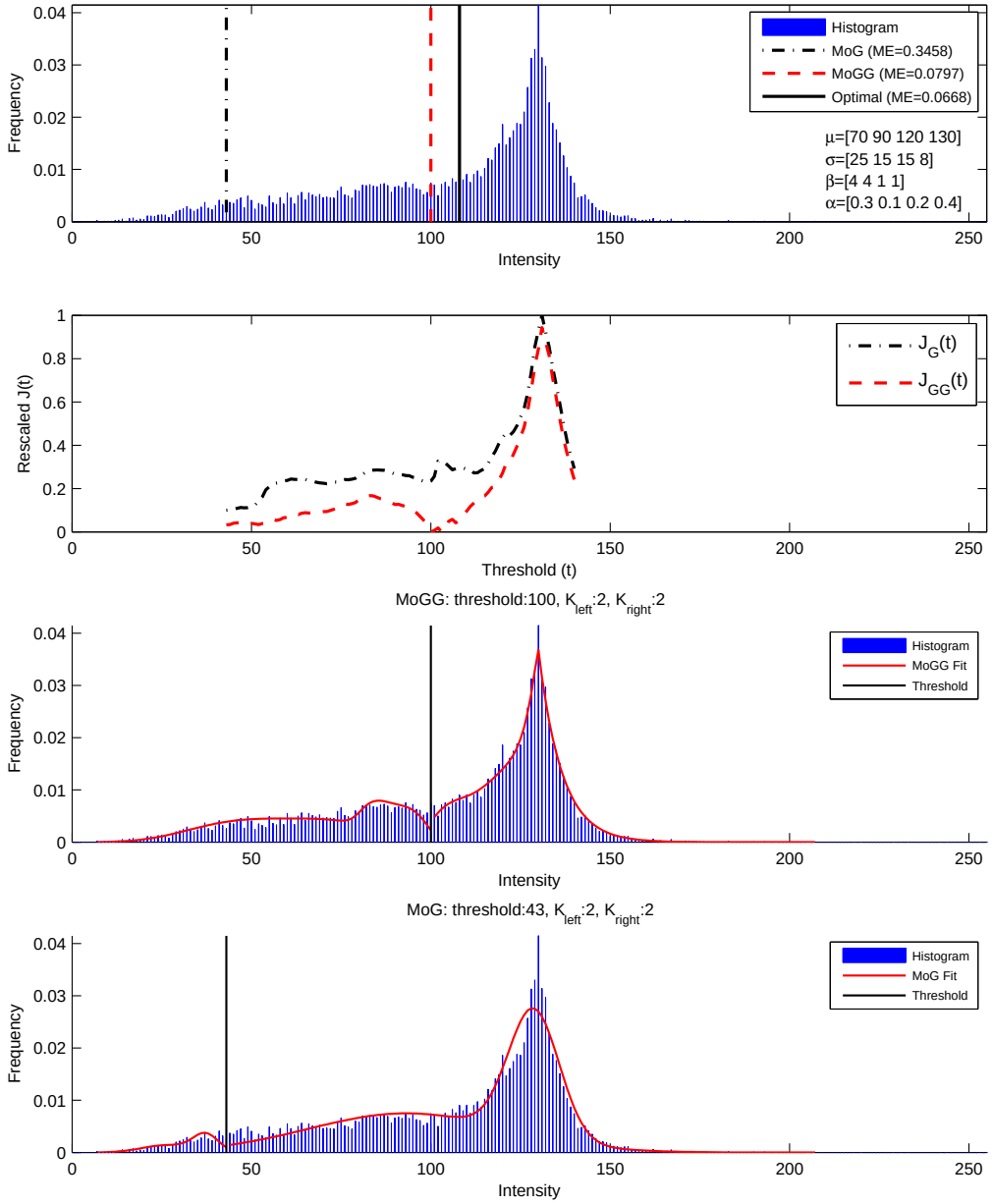


FIGURE 3.10: Example of a generated data set histogram from the benchmark 2 of Table 3.5 with optimal thresholding results and misclassification errors (ME) given by MoG and MoGG methods.

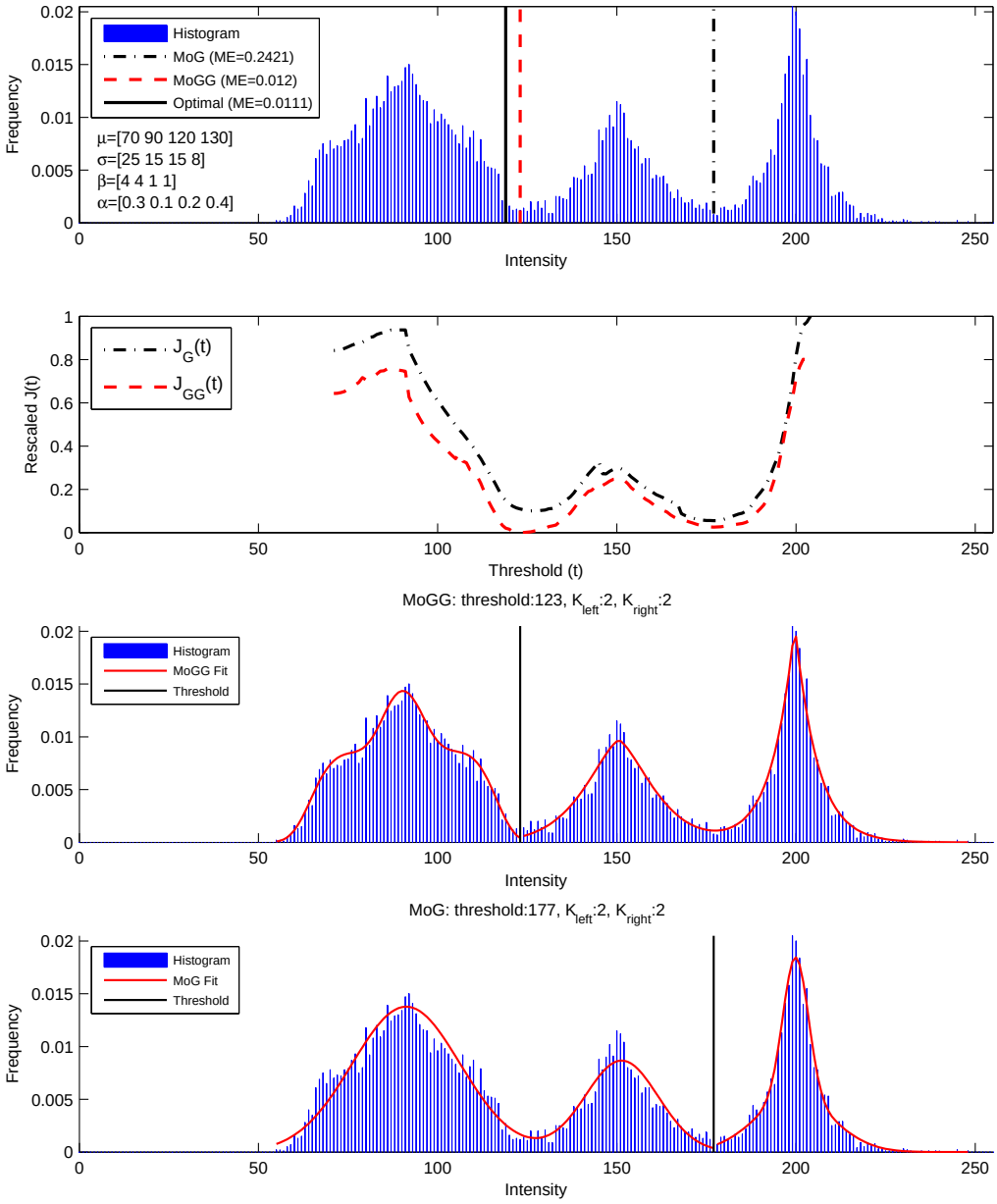


FIGURE 3.11: Example of a generated data set histogram from the benchmark 3 of Table 3.5 with optimal thresholding results and misclassification errors (ME) given by MoG and MoGG methods.

3.4.3 Multi-thresholding for simulated datasets ($K > 2$)

Multi-thresholding has a variety of application fields (e.g., remote sensing, medical image segmentation, etc.). Similarly to the simple case of binarization ($K = 2$), five benchmarks of randomly-generated data were considered. Each benchmark contains 50 datasets with parameter settings presented in Table 3.6. For the case where the histogram is multimodal ($K > 3$) and with different types of component shapes, results show that mixture-based methods (MoG and MoGG) are much more efficient compared to standard Otsu's and median extension methods.

Figure 3.12 shows an example of tri-class histogram thresholding (i.e., $\mathbf{t} = (t_1^*, t_2^*)$). The minimized functions for the MoG and MoGG models are denoted by $J_G(\mathbf{t}) = (J_{G1}(\mathbf{t}), J_{G2}(\mathbf{t}))$ and $J_{GG}(\mathbf{t}) = (J_{GG1}(\mathbf{t}), J_{GG2}(\mathbf{t}))$, respectively. We can observe that the standard Otsu's and median based methods failed to find good thresholds because of the difference between shape and size parameters of the three classes $(\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3)$: *i*) for the scale parameter σ, $\mathcal{C}_1$ and $\mathcal{C}_2$ have $\sigma_1 = 10$ and $\sigma_2 = 10$, respectively, while the only component that constitute $\mathcal{C}_3$ class have $\sigma_3 = 2$, *ii*) the proportion parameter π: $\mathcal{C}_1$ have each one $\pi_{11} = \pi_{12} = 0.35$ for the class $\mathcal{C}_2$ have $\pi_2 = 0.20$, whereas $\mathcal{C}_3$ has a value of $\pi_3 = 0.10$. The MoG method failed to find the first threshold and succeeded in finding the second one, whereas the MoGG method succeeded in finding both thresholds.

Table 3.7 summarizes results obtained by all studied methods for all generated data sets. It shows for each benchmark average ME values obtained by each method and between brackets the percentage of data sets where MoGG method gave the best threshold. We note that for the case of three-levels thresholding, the misclassification error (ME) is computed as the average of the two thresholds t_1^* and t_2^* computed ME (i.e., $ME = \frac{ME_1 + ME_2}{2}$), where ME_1 and ME_2 are values of ME corresponding to thresholds t_1^* and t_2^*, respectively.

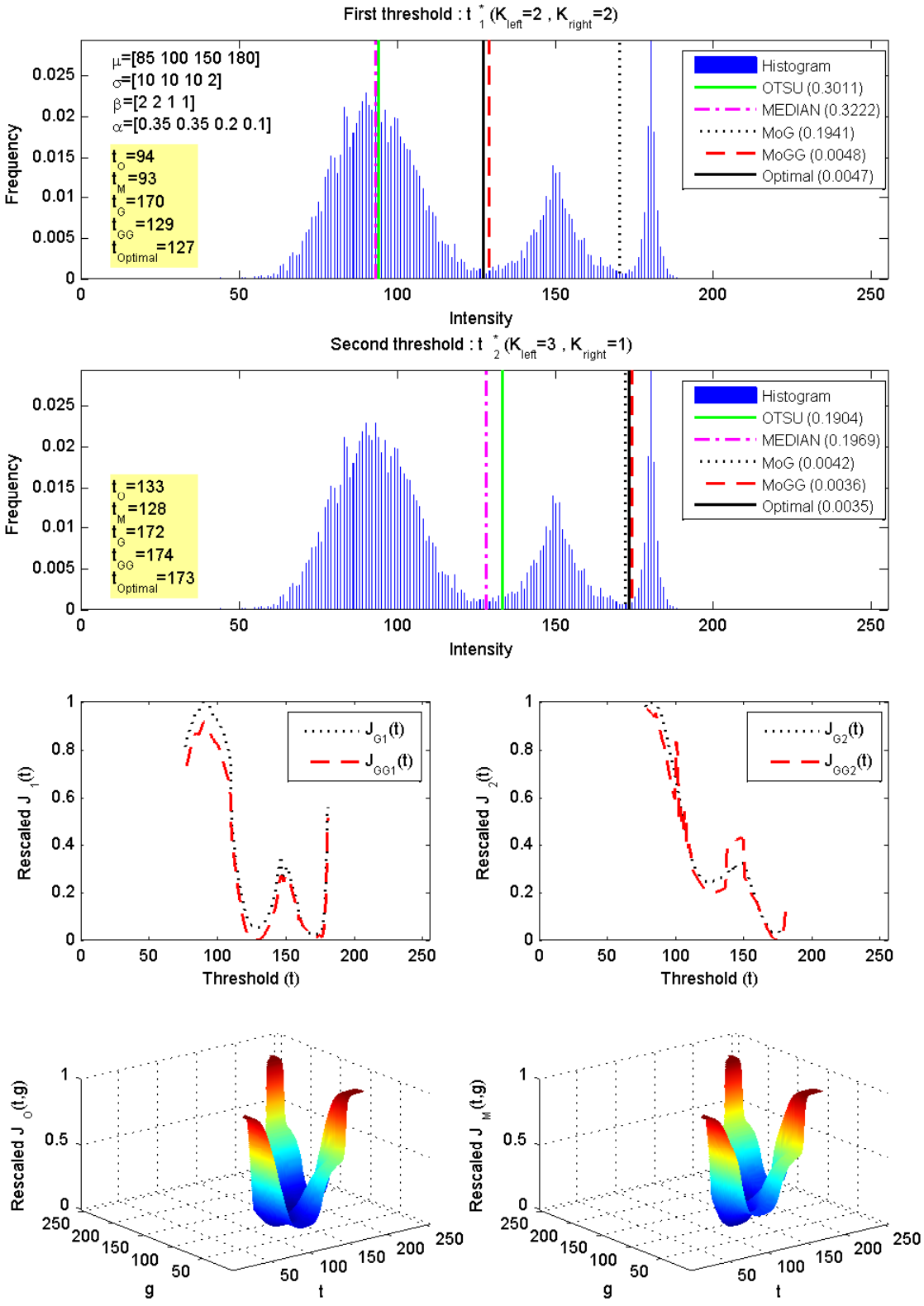


FIGURE 3.12: Multi-thresholding results for a trimodal dataset histogram. From top to bottom and left to right, we show: the first and second optimal thresholds ($\mathbf{t}_1^*$ and $\mathbf{t}_2^*$) given by all methods (first and second row), plots of MoG and MoGG criterion functions $J_1(\mathbf{t})$ and $J_2(\mathbf{t})$ respectively (third row) and plots of standard Otsu's method function $J_O(\mathbf{t}, \mathbf{g})$ and median extension method function $J_M(\mathbf{t}, \mathbf{g})$ (fourth row).

Bench.	Class	K_r	μ	σ	λ	π
Bench.1	$r=1$	2	$\mu_{1,1}=50$	$\sigma_{1,1}=20$	$\lambda_{1,1}=2.00$	$\pi_{1,1}=0.40$
			$\mu_{1,2}=65$	$\sigma_{1,2}=20$	$\lambda_{1,2}=2.00$	$\pi_{1,2}=0.20$
	$r=2$	2	$\mu_{2,1}=120$	$\sigma_{2,1}=20$	$\lambda_{2,1}=3.00$	$\pi_{2,1}=0.15$
			$\mu_{2,2}=135$	$\sigma_{2,2}=20$	$\lambda_{2,2}=1.50$	$\pi_{2,2}=0.15$
	$r=3$	1	$\mu_{3,1}=180$	$\sigma_{3,1}=15$	$\lambda_{3,1}=1.00$	$\pi_{3,1}=0.10$
Bench.2	$r=1$	1	$\mu_{1,1}=4$	$\sigma_{1,1}=3$	$\lambda_{1,1}=2.10$	$\pi_{1,1}=0.15$
	$r=2$	1	$\mu_{2,1}=22$	$\sigma_{2,1}=9$	$\lambda_{2,1}=2.70$	$\pi_{2,1}=0.15$
	$r=3$	3	$\mu_{3,1}=77$	$\sigma_{3,1}=19$	$\lambda_{3,1}=2.40$	$\pi_{3,1}=0.25$
			$\mu_{3,2}=107$	$\sigma_{3,2}=20$	$\lambda_{3,2}=2.10$	$\pi_{3,2}=0.35$
			$\mu_{3,3}=133$	$\sigma_{3,3}=33$	$\lambda_{3,3}=1.50$	$\pi_{3,3}=0.15$
Bench.3	$r=1$	2	$\mu_{1,1}=50$	$\sigma_{1,1}=10$	$\lambda_{1,1}=2.00$	$\pi_{1,1}=0.30$
			$\mu_{1,2}=75$	$\sigma_{1,2}=15$	$\lambda_{1,2}=2.00$	$\pi_{1,2}=0.15$
	$r=2$	2	$\mu_{2,1}=130$	$\sigma_{2,1}=25$	$\lambda_{2,1}=3.10$	$\pi_{2,1}=0.15$
			$\mu_{2,2}=180$	$\sigma_{2,2}=15$	$\lambda_{2,2}=1.30$	$\pi_{2,2}=0.30$
	$r=3$	1	$\mu_{3,1}=220$	$\sigma_{3,1}=5$	$\lambda_{3,1}=2.30$	$\pi_{3,1}=0.10$
Bench.4	$r=1$	2	$\mu_{1,1}=85$	$\sigma_{1,1}=10$	$\lambda_{1,1}=2.00$	$\pi_{1,1}=0.35$
			$\mu_{1,2}=100$	$\sigma_{1,2}=10$	$\lambda_{1,2}=2.00$	$\pi_{1,2}=0.35$
	$r=2$	1	$\mu_{2,1}=150$	$\sigma_{2,1}=10$	$\lambda_{2,1}=1.00$	$\pi_{2,1}=0.20$
	$r=3$	1	$\mu_{3,1}=180$	$\sigma_{3,1}=2$	$\lambda_{3,1}=1.00$	$\pi_{3,1}=0.10$
Bench.5	$r=1$	2	$\mu_{1,1}=60$	$\sigma_{1,1}=15$	$\lambda_{1,1}=4.00$	$\pi_{1,1}=0.20$
			$\mu_{1,2}=80$	$\sigma_{1,2}=15$	$\lambda_{1,2}=4.00$	$\pi_{1,2}=0.20$
	$r=2$	1	$\mu_{2,1}=140$	$\sigma_{2,1}=15$	$\lambda_{2,1}=1.00$	$\pi_{2,1}=0.30$
	$r=3$	1	$\mu_{3,1}=190$	$\sigma_{3,1}=8$	$\lambda_{3,1}=1.00$	$\pi_{3,1}=0.30$

TABLE 3.6: Parameters setting for multimodal simulated dataset benchmarks.

Bench.	Error Otsu	Error Median	Error MoG	Error MoGG
1	0.050 (100%)	0.059 (100%)	0.034 (82%)	0.032
2	0.296 (100%)	0.269 (100%)	0.015 (60%)	0.015
3	0.159 (100%)	0.099 (100%)	0.026 (94%)	0.023
4	0.252 (100%)	0.263 (100%)	0.099 (94%)	0.023
5	0.015 (100%)	0.014 (98%)	0.151 (100%)	0.012

TABLE 3.7: ME for multi-level thresholding of the multimodal simulated datasets benchmarks (Between brackets is the percentage of data sets where MoGG gives the least error).

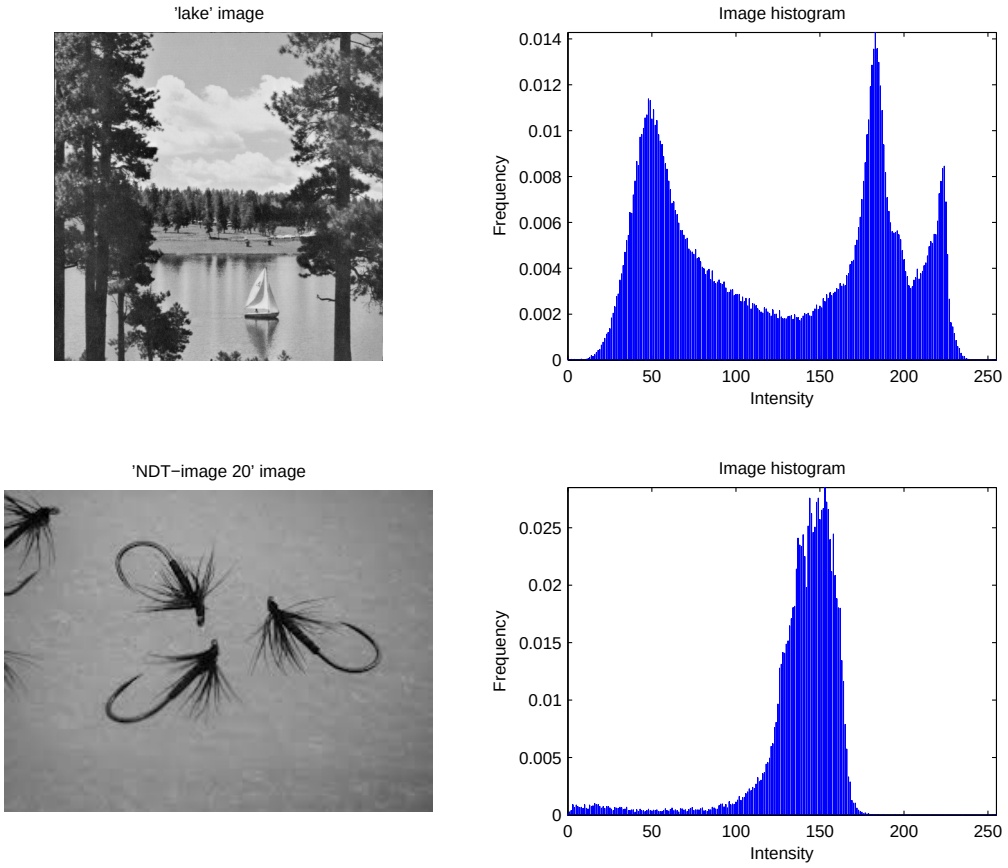


FIGURE 3.13: Real images used for the multi-thresholding tests. From top to down, left to right, we show: the 'Lake' original image, its histogram (first row) and an image from the web and its histogram (second row).

3.4.4 Multi-thresholding for real images ($K > 2$)

Here, two sample images were used, the 'Lake' (512×512) classic benchmark test image that was also employed in [148], and another image image from the web. The second image is characterized by a non-uniform background. For both images, the parameters K_1, K_2 and K_3 represent the number of components in each class $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$, respectively (see Figure 3.13).

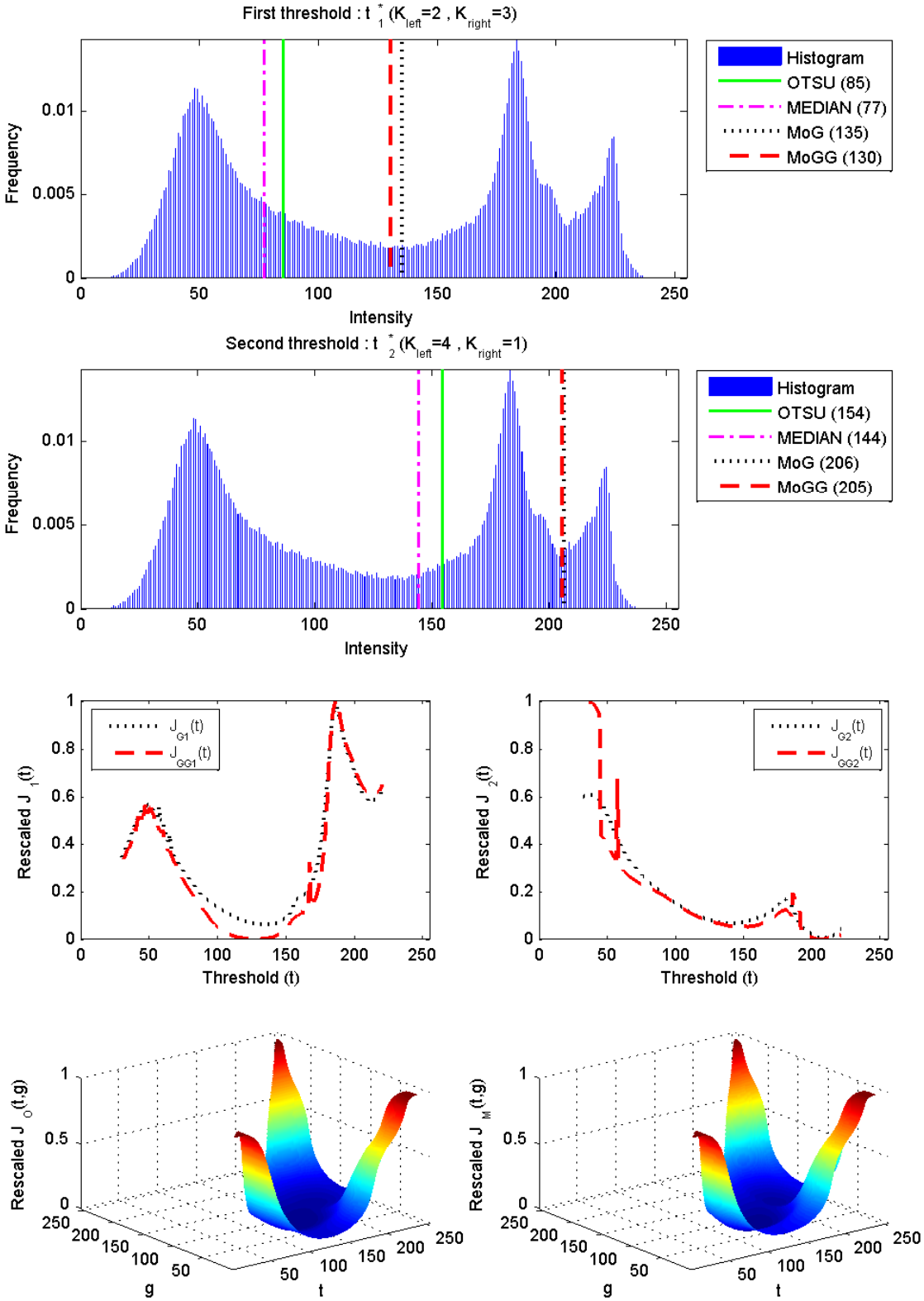
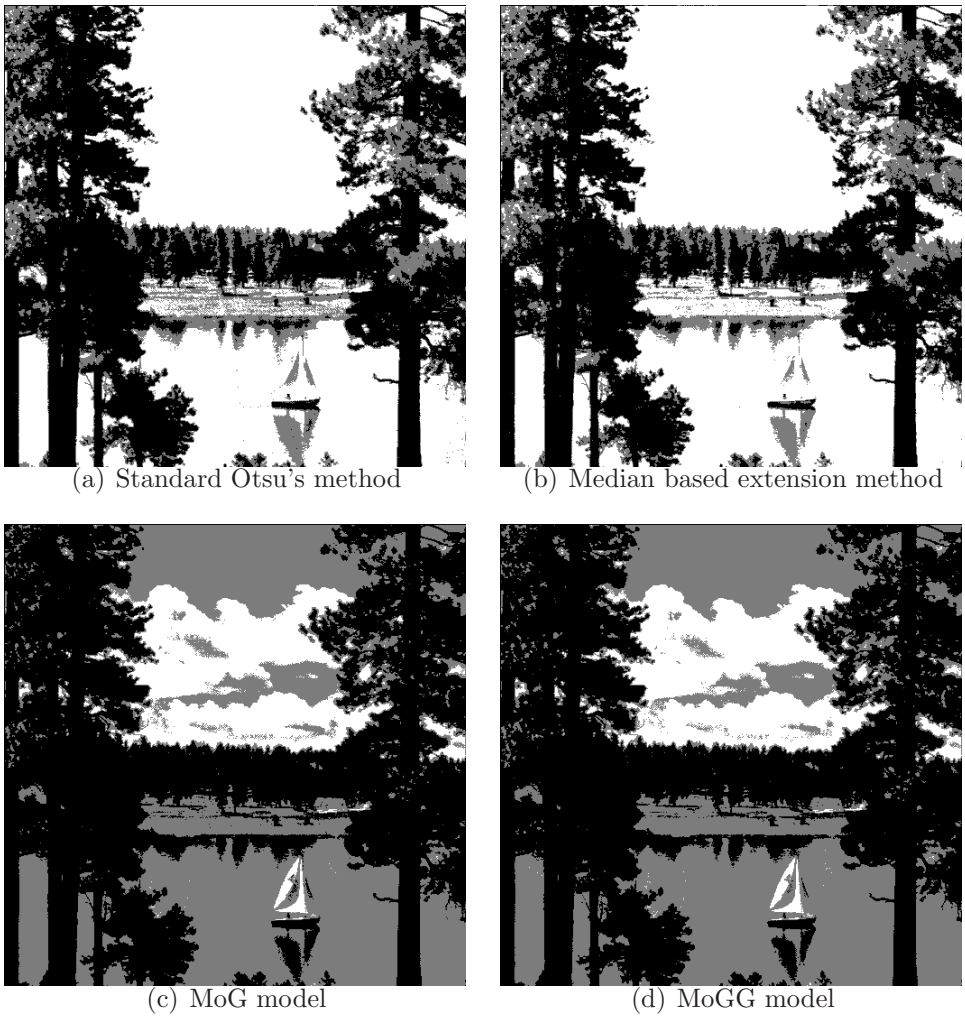


FIGURE 3.14: Multi-thresholding results for 'Lake' image. From top to down, left to right, we show: the first and second optimal thresholds ($\mathbf{t}_1^*$ and $\mathbf{t}_2^*$) given by all methods (first and second row), plots of MoG and MoGG criterion functions $J_1(\mathbf{t})$ and $J_2(\mathbf{t})$ respectively (third row) and plots of standard Otsu's method function $J_O(\mathbf{t}, \mathbf{g})$ and median extension method function $J_M(\mathbf{t}, \mathbf{g})$ (fourth row).



(a) Standard Otsu's method (b) Median based extension method

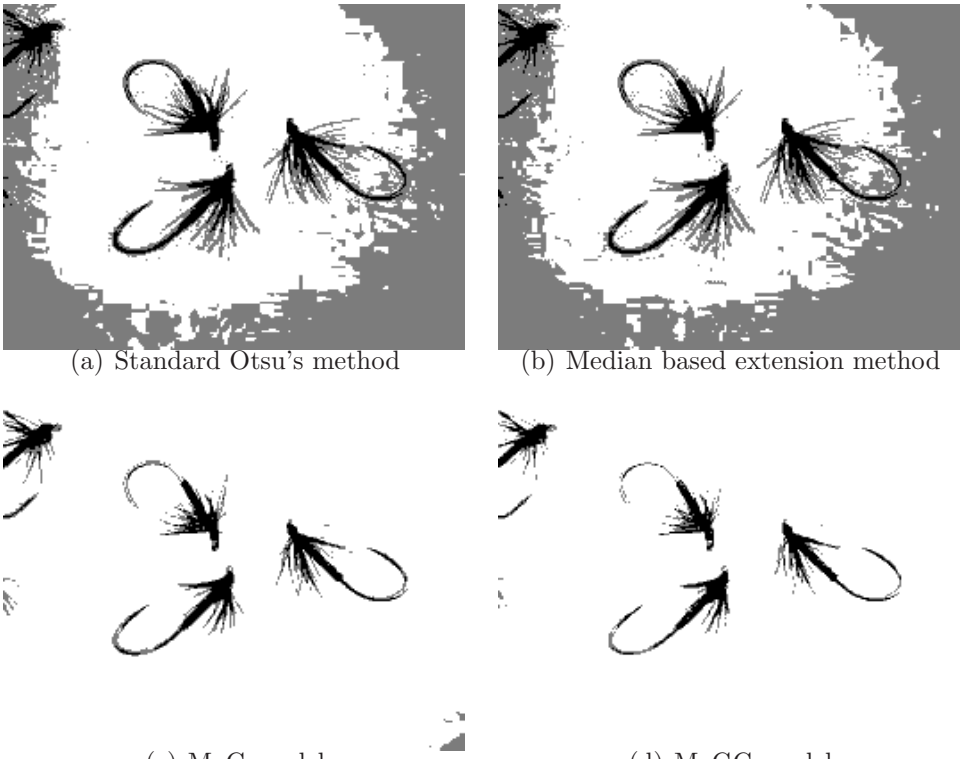
(c) MoG model (d) MoGG model

FIGURE 3.15: Segmentation results for the 'lake' image obtained by application of standard Otsu's, median based extension , MoG and MoGG methods, respectively.

Figure 3.14 illustrates the multi-thresholding results for the 'Lake' image. In the shown histogram, there are three clearly perceptible modes and one or more classes with skewed distributions. We can observe that only mixture based methods successfully identified the three classes represented by the three modes of intensity (see Figs. 3.14 and 3.15). Although the first optimal threshold t_1 can be determined successfully by all methods, both Otsu's based methods failed to place the second optimal threshold t_2 in the right position of the histogram.

This is mainly due to: *i*) the difference between the parameters π_1, π_2 and π_3 that represents the size of classes $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$, respectively. It can be observed that there is a great difference between π_1 and π_2 against π_3, *ii*) the skewness of

components distributions. The same comments can be made about Figure 3.17 for the other image, where the background of the image was over-segmented using the standard Otsu's and median extension methods, as well as the MoG-based method. The MoGG gave a better segmentation compared to the other methods since it was able to model successfully the non-uniform background in one single GGD (see Figs. 3.17 and 3.16).



(a) Standard Otsu's method (b) Median based extension method

(c) MoG model (d) MoGG model

FIGURE 3.16: Segmentation results of the second image by application of standard Otsu's, median based extension , MoG and MoGG methods, respectively.

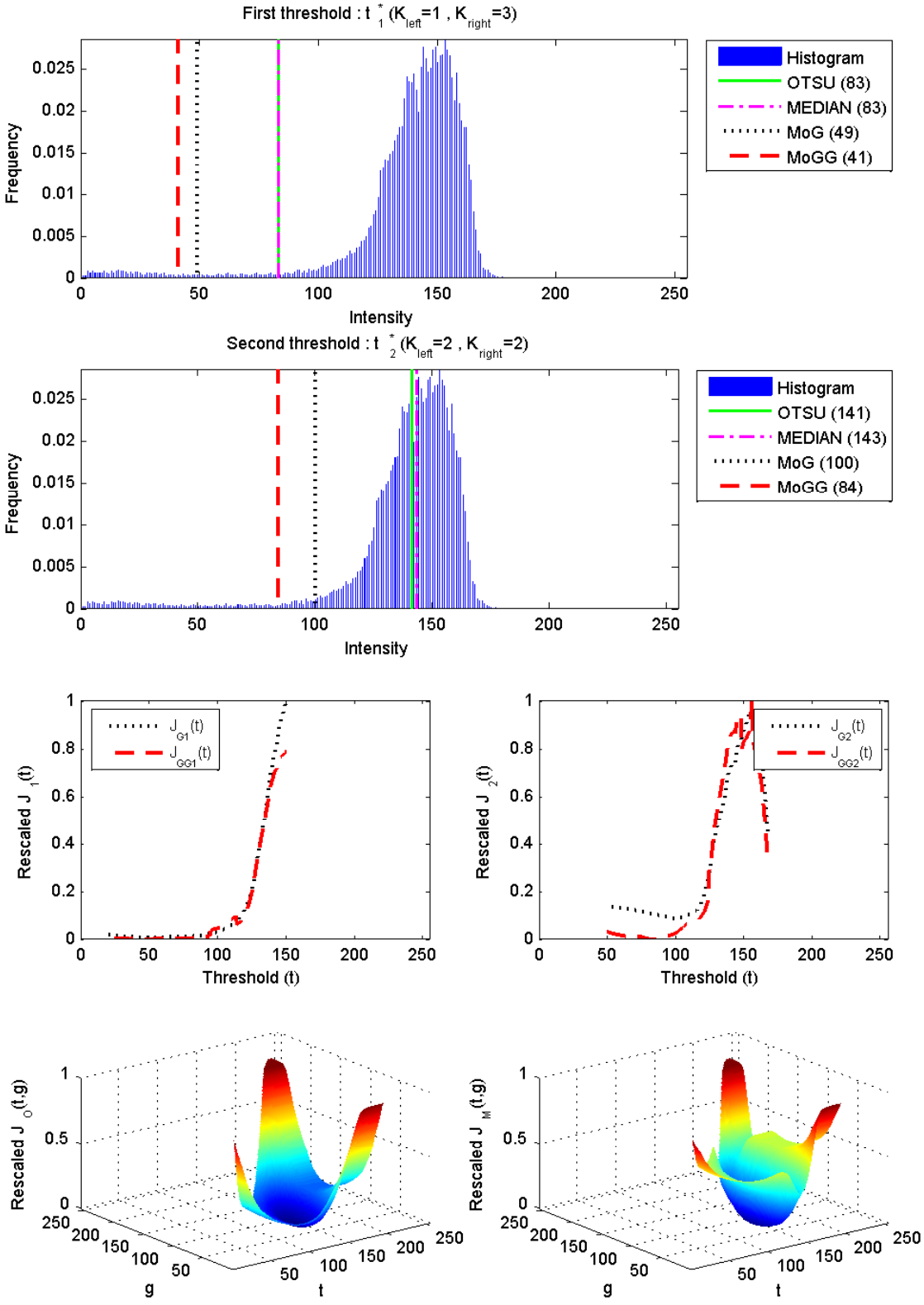


FIGURE 3.17: Multi-thresholding results for the second image. Caption is as for Figure 3.14.

3.5 Computational time complexity

The computational complexity of the standard Otsu's and its median extension approaches grows exponentially with the number of thresholds and gray levels. The two methods can compute the optimal threshold in $\mathcal{O}(N^L)$ operations, where N is the number of gray levels, $L = K - 1$ is the number of thresholds and K is the number of mixture components. This complexity is due to the minimization of criterion functions $J_O(\mathbf{t})$ and $J_M(\mathbf{t})$ which have L-dimensions (i.e., a function of L variables). This property considerably increases the computation time, especially for great values of K (eg. $K > 5$). Several works in the past (e.g., see [74, 152]) proposed faster versions for multi-level thresholding using the Otsu's method.

In our approach, the thresholding is based on fitting a mixture of generalised Gaussian distributions to multi-modal classes, where the EM algorithm is used to estimate the mixture parameters. Consequently, the MoGG method adds some computation time for determining the thresholds, compared to the standard Otsu's and its median extension variant. However, given the improvement of accuracy in the thresholds determination, this additional time is not a limitation. To speed up threshold calculations, the function $J_{GG}(\mathbf{t})$ is rearranged to be uni-dimensional (i.e,. a function of one variable) whatever the number of classes K; thus, the proposed MoGG method has a linear complexity. In fact, the proposed algorithm can find optimal thresholds in time $\mathcal{O}(NL)$ operations of EM algorithm estimation (see algorithm 2). Consequently, for high values of K (e.g., $K > 5$), the proposed mixture methods consume less time than standard Otsu's and median extension methods.

Figures 3.12, 3.14, and 3.17 show examples of multi-thresholding ($K = 3$), where we can observe that $J_G(\mathbf{t})$ and $J_{GG}(\mathbf{t})$ functions are uni-dimensional while $J_O(\mathbf{t}, \mathbf{g})$ and $J_M(\mathbf{t}, \mathbf{g})$ are two-dimensional (i.e., they are defined by two variables $\mathbf{t}$ and $\mathbf{g}$).

3.6 Remarks and discussion

In this section, we present some remarks and caveats about the approaches studied in this chapter, and thresholding-based segmentation in general. These revolve around the multi-modality of criterion function used for threshold determination. We observed in some cases that the best threshold is not given by the global minimum but by a local minimum. This limitation was already observed for the standard Otsu's method in [65]. It has been demonstrated that the objective function may not only be multi-modal, but more importantly, if it is multi-modal, its global maximum is not guaranteed to give a correct threshold. For the median-based extension [148], the authors used a two-component Laplace mixture to simulate data,

where the two classes have greatly disparate sizes (with a ratio of 99 to 1). They observed that sometimes for the standard Otsu's method and its median based extension, local minima of$J(\mathbf{t})$ provide better thresholds than do global minima.

To illustrate the behavior of MoG and MoGG criterion function against Otsu's and median extension ones, we conducted the same experiments as in [65] and [148]. Two datasets were randomly generated, each of which is a two-component Laplace and Gaussian mixture, respectively, where the two classes have the following parameter settings. For the left class, we used: $\mu_1 = 80, \sigma_1 = 15, \pi_1 = 0.99$, while for the right class, we used: $\mu_2 = 190, \sigma_2 = 20, \pi_2 = 0.01$. These parameters guarantee an inter-class ratio of 99 to 1. Figures 3.18 and 3.19 show thresholding results and multi-modal criterion functions ($J_O(\mathbf{t})$, $J_M(\mathbf{t})$, $J_G(\mathbf{t})$ and $J_{GG}(\mathbf{t})$) obtained by all methods for the two cases of Gaussian and Laplace datasets. For standard Otsu's and median extension methods, the local minima at 150 in figure 3.18 gives better thresholds. Functions $J_O(\mathbf{t})$ and $J_M(\mathbf{t})$ are multi-modal and the local minimum indicates a better threshold than does the global minimum. Consequently, the two methods fail both to find a good threshold. However, for the mixture-based methods (MoG and MoGG), despite the multi-modality of criterion functions $J_G(\mathbf{t})$ and $J_{GG}(\mathbf{t})$, the two methods give a good threshold and successfully separate the two classes $\mathcal{C}_1$ and $\mathcal{C}_2$. In Figure 3.19, the fact that MoGG method gives better threshold (at about 162) than that given by MoG method (at about 130) is due to the shape parameter λ which allows MoGG method to fit the Laplace distribution better than the MoG method.

We note finally that to resolve the problem of multi-modality of the criterion function, a simple valley check can be used to enable the thresholding method to be extended to estimate thresholds reliably over a larger range of inter-class size ratios [65]. For instance, a threshold t^* that does not satisfy $h(t^*) < h(\overline{x}_1(t^*))$ and $h(t^*) < h(\overline{x}_2(t^*))$ will be ignored. For MoG and MoGG multi-class thresholding methods, we propose the following "valley check" : let $\overline{x}$ the mixture mean vector estimated by EM algorithm, for $r = \{1, ..., K-1\}$ a threshold t_r^* must satisfy:

$$\overline{x}(\sum_{i=1}^{r} K_i) < t_r^* < \overline{x}(\sum_{i=1}^{r+1} K_i), \tag{3.27}$$

where K_i is the component number of class C_i (see algorithm 2).

3.7 Conclusion

A new thresholding approach, based on the Mixture of Generalized Gaussian model (MoGG method), is presented in this chapter. The approach generalizes previous methods to multi-thresholding and multi-modal classes. It has been successfully

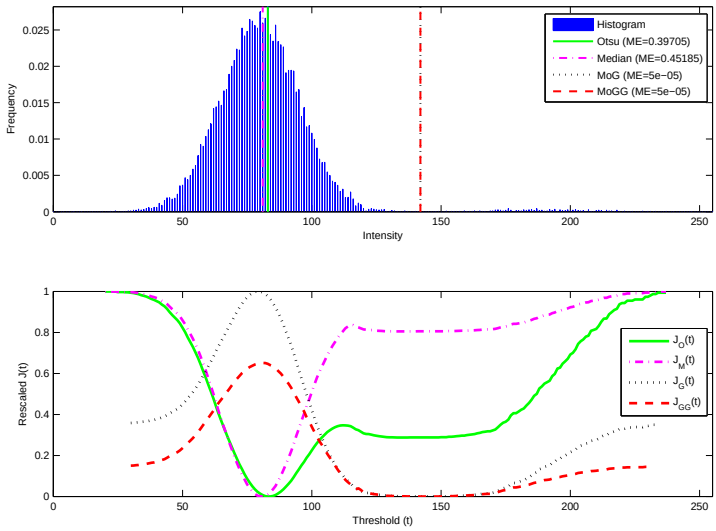


FIGURE 3.18: Thresholds and criterion function for the Otsu's, the median-based extension, MoG and MoGG methods applied for segmentation data simulated from mixtures of two Gaussian distributions.

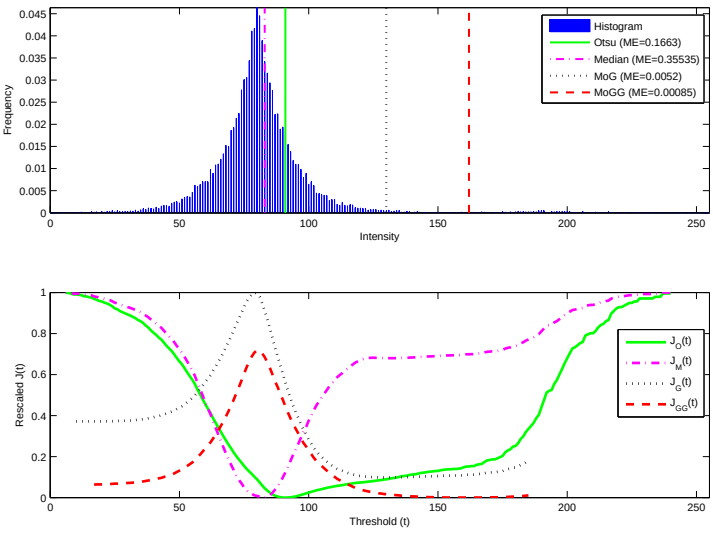


FIGURE 3.19: Thresholds and Criterion function for the Otsu's, the median-based extension, MoG and MoGG methods applied for segmentation data simulated from mixtures of two Laplace distributions.

tested on segmentation of real images (NDT-images [118]) and randomly generated data sets. Experiments have shown the performance of the proposed approach

and showed that it can achieve more optimal thresholds than the standard Otsu's method [97], the median based extension method [148], as well as thresholding based on Gaussian mixture models.

4

Video Foreground Segmentation Fusing Temporal & Spatial Information

4.1 Introduction

In this chapter, we propose an approach for background modeling and subtraction that is efficient for coping with complex background dynamics, cast shadows and illumination changes, and performs very well for other challenges such as jitter and PTZ camera effects. Our model combines temporal and spatial information for BS.

Temporal information is modeled locally using an online-learned mixture of generalized Gaussian (MoGG) distributions [3]. In fact, MoGGs are more effective than GMMs to fit a wide range of data histograms (e.g., with *leptokurtic* and *platykurtic* modes) and is more robust to noise and outliers. Herein, we propose a new procedure for real-time online updating of the MoGG parameters in the context of BS. Moreover, we enhance the MoGG capability to model different background dynamics by introducing background/foregound co-occurrence analysis. This drastically decreases the amount of false positives caused by complex background dynamics along with more flexibility to model variable pixel state duration.

Spatial information is incorporated to the BS model through multi-scale inter-frame correlation analysis and histogram matching. Our approach not only allows for dissociating changes due to shadows and illumination changes from those

of moving objects, but also for enhancing accuracy of background modeling and subtraction in the presence of noise, highlights and complex background dynamics. Furthermore, we introduce a global technique, based on frame displacement estimation, to deal with PTZ camera motion effects. Experiments on standard datasets have shown that our method outperforms several recent state-of-the-art methods on several of the aforementioned challenges.

Our method for video foreground segmentation brings several contributions to the state of the art by proposing a new scheme allowing to fuse spatial and temporal information for more robust BS. In addition to deal with several complex background dynamics, it deals efficiently with shadows, illumination changes, camera jitter and PTZ camera effects. Finally, it is optimized to process videos with near real-time capability. We briefly summarize our contributions as follows:

1. we combine temporal and spatial information modeling to efficiently cope with challenges such as cast shadows, illumination changes and non-static backgrounds.

2. a new scheme is proposed for temporal information modeling by coupling MoGGs and objects/background co-occurrence analysis. This allows accurate modeling of various background dynamics (e.g., fast foreground/background switching, fountains, etc.).

3. we model spatial information using inter-frame spatial structure and histogram analysis. Spatial information makes our BS method less sensitive to shadows and illumination changes.

4. we introduce a global technique that deals with pan-tilt-zoom (PTZ) camera effects in videos.

5. we develop several procedures to optimize the computation time of our algorithm, as well as a new method for adapting the learning rate for the MoGG parameters.

This chapter is organized as follows: Section 4.2 describes our approach combining temporal and spatial information modeling for BS. Section 4.3 presents our experimental results. We end the chapter with a conclusion and future work perspectives.

4.2 Temporal/spatial information modeling

The proposed algorithm is composed of temporal and spatial modules interacting with each other for efficient BS (see Fig. 4.1). Temporal information is modeled by combining MoGGs and co-occurrence analysis, which allows for accurate

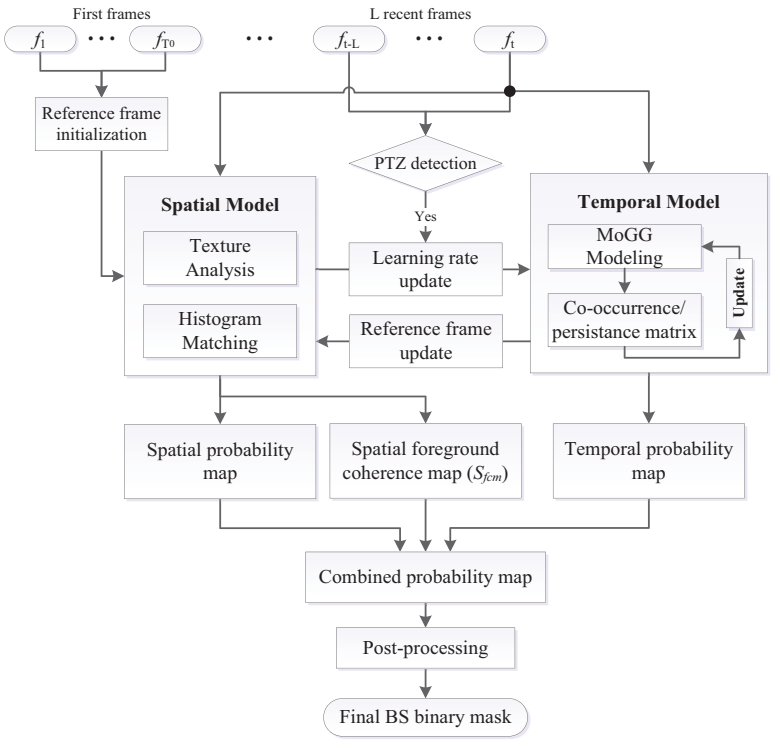


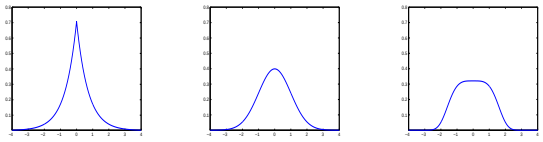
FIGURE 4.1: The proposed algorithm architecture for BS.

representation of various complex background dynamics. Spatial information is incorporated to the method using correlation analysis and histogram matching which mitigate effects of cast shadows, highlights, illumination changes and PTZ camera effects. This information is also used to derive an adaptive scheme to estimate the learning rate of the MoGG parameters. This scheme contributes also to accelerate the convergence rate of the background model and prevent it from rapidly absorbing objects.

4.2.1 Basic temporal information modeling using MoGGs

The MoGG model has flexibility to accurately fit different histogram shapes while ensuring robustness to noise and/or outliers which cause heavy-tailed distributions (e.g., see Fig. 4.3) [3, 6]. The one dimensional generalized Gaussian density (GGD) is defined in $\mathbb{R}$ as follows:

$$p(X|\theta) = A(\lambda, \sigma) \exp\big(- B(\lambda) \, |(X - \mu)/\sigma|^{\lambda} \big), \tag{4.1}$$



$\lambda = 1$ $\lambda = 2$ $\lambda = 3$

FIGURE 4.2: Different shapes of the GGD distribution as a function of the parameter λ ($\mu = 0, \sigma = 1$).

where $\theta = \{\mu, \sigma, \lambda\}$ is the set of GGD parameters, $A(\lambda, \sigma) = \frac{\lambda\sqrt{\Gamma(3/\lambda)/\Gamma(1/\lambda)}}{(2\sigma\Gamma(1/\lambda))}$ and $B(\lambda) = [\Gamma(3/\lambda)/\Gamma(1/\lambda)]^{\lambda/2}$; $\Gamma(.)$ being the gamma function. The parameters μ and σ are the GGD location and dispersion parameters. The parameter λ controls the kurtosis of the pdf and determines whether its shape is peaked or flat (see Fig. 4.2). To model temporal changes in video, we consider the history of each pixel (x, y) at time t as $\{\vec{X}_0, ..., \vec{X}_t\}$. Each vector $\vec{X}_t$ is D-dimensional $\vec{X}_t = (X_{1,t}, ..., X_{D,t}) \in \mathbb{R}^D$ ($D = 3$ for RGB color). Suppose that the history of the pixel at time t is modeled as a mixture of K components where, given that the dimensions of $\vec{X}_t$ are independent in each class, the probability of observing the vector $\vec{X}_t$ is given as [3]:

$$p(\vec{X}_t) = \sum_{i=1}^{K} \omega_{i,t} * \Pi_{d=1}^{D} p(X_{d,t}|\vec{\theta}_{i,d,t}), \tag{4.2}$$

where $\vec{\theta}_{i,d,t} = (\mu_{i,d,t}, \sigma_{i,d,t}, \lambda_{i,d,t})$ are parameters describing the dimension d of the ith component of the mixture, $\omega_{1,t}$,..., $\omega_{K,t}$ are the weights of components such that $\sum_{i=1}^{K} \omega_{i,t} = 1$ and K is a parameter that represents the maximum number of foreground/background components.

Fig. 4.3 presents an example comparing GMM and MoGG modeling of video data. The first row shows sample frames of two videos where the gray level history of the pixel in the center of the white square is clearly non-Gaussian. Note that the outliers in this example are caused by cast shadows. The second and third rows show each pixel temporal histogram and its fitting using a GMM and a MoGG, respectively. We can observe that the MoGG model can fit better the histogram shapes than the GMM. This is due to the shape parameter λ that makes MoGG model less sensitive to outliers and give it flexibility to fit accurately heavy tailed and flat/picked histogram modes.

We assume that at frame I_{t+1}, a pixel (x, y) have value $\vec{X}_{t+1}$ and a match is found with one of the components of the mixture (let's say with component k) if

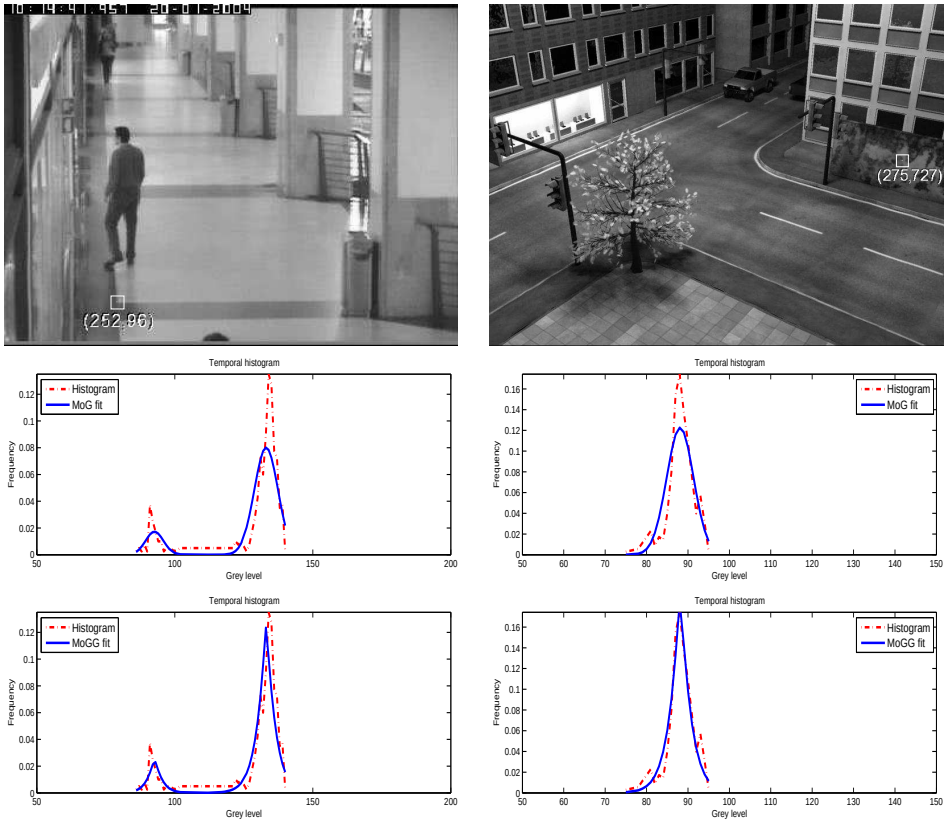


FIGURE 4.3: Sample frame (top row), temporal histogram with GMM fitting (middle row) and temporal histogram with MoGG fitting (bottom row) for two sequences: 'CAVIAR' [24] that includes indoor and outdoor light (left column), and 'SABS' dateset [22], 'Basic' sequence that is a synthetic video (right column). The temporal histograms were calculated from 400 and 350 frames, respectively, and the pixel of interest is the center of the white box at each frame. For the CAVIAR frame $K = 3$, and for SABS frame $K = 1$.

we have the following condition:

$$p(\vec{\theta}_{k,t}|\vec{X}_{t+1}) > \tau, \text{ with } \quad k = \arg\max_i \{p(\theta_{i,t}|\vec{X}_{t+1})\}, \tag{4.3}$$

where τ is a given threshold and $p(\theta_{i,t}|\vec{X}_{t+1})$ is the posterior probability of the ith mixture component. If a match is found, only the parameters of component k are updated. Otherwise, a new component of the mixture is created. Note that an online updating method has been proposed in [3] using the EM algorithm. However, the procedure uses Fisher scoring which incurs a huge computation time to calculate the likelihood derivatives. Here, we propose a faster procedure based on statistical moments for online updating the MoGG parameters. Since $\sum_{i=1}^{K} \omega_{i,t} = 1$,

the weights are updated as follows [125]:

$$\omega_{i,t+1} = (1-\rho) * \omega_{i,t} + \rho * \delta(i=k), i = 1, ..., K \tag{4.4}$$

where δ is the delta function and ρ is a learning parameter. After this updating, we normalize all the weights. The entries of the shape parameter vector $\vec{\lambda}_k$ are updated using the following property:

$$\left[\frac{\sigma_{k,d,t}}{\mathbb{E}\big[\mid X_{k,d} - \mu_{k,d,t} \mid\big]}\right]^2 = \Gamma\left(1/\lambda_{k,d,t}\right)\Gamma\left(3/\lambda_{k,d,t}\right)/\Gamma^2\left(2/\lambda_{k,d,t}\right), \tag{4.5}$$

where $X_{k,d}$ are values of the dth dimension of $\vec{X}$ assigned to component k until time t, $\mu_{k,d,t}$ is the location parameter of the same component and $\mathbb{E}\big[\mid X_{k,d} - \mu_{k,d,t} \mid\big]$ is the mean of centered absolute values (MAV), given as:

$$\mathbb{E}\big[\mid X_{k,d} - \mu_{k,d,t} \mid\big] = \frac{1}{N_k}\sum_{s=1}^{t} \mid X_{k,d,s} - \mu_{k,d,t} \mid, \tag{4.6}$$

where N_k is the number of data assigned to component k. Using the shorthand $\tilde{X}_{k,d,t}$ to designate $|X_{k,d,t} - \mu_{k,d,t}|$, the MAV is updated online as follows:

$$\mathbb{E}_{(t+1)}\big[\tilde{X}_{k,d}\big] = (1-\rho) * \mathbb{E}_{(t)}\big[\tilde{X}_{k,d}\big] + \rho * \tilde{X}_{k,d,t+1}. \tag{4.7}$$

Consequently, for a matched component k, $\vec{\lambda}_k$ can be efficiently updated for each dimension using Eq. (4.5) via a quick look-up table search [121]. The location parameter of the same component is updated using Eq. (4.8) as follows [3]:

$$\mu_{k,d,t+1} = \frac{\sum_{s=1}^{t+1} \tilde{X}_{k,d,s}^{(\lambda_{k,d,t+1}-2)} * X_{k,d,s}}{\sum_{s=1}^{t+1} \tilde{X}_{k,d,s}^{(\lambda_{k,d,t+1}-2)}} = \frac{\alpha_{k,d}(t+1)}{\beta_{k,d}(t+1)}, \tag{4.8}$$

where $\lambda_{k,d,t+1}$ is the shape parameter for dimension d computed at time $t+1$. The terms $\alpha_{k,d}(.)$ and $\beta_{k,d}(.)$ can be updated online using the following equations:

$$\alpha_{k,d}(t+1) = \alpha_{k,d}(t) + X_{k,d,t+1} * \tilde{X}_{k,d,t}^{(\lambda_{k,d,t+1}-2)} \tag{4.9}$$

$$\beta_{k,d}(t+1) = \beta_{k,d}(t) + \tilde{X}_{k,d,t}^{(\lambda_{k,d,t+1}-2)}. \tag{4.10}$$

Finally, the scale parameter $\sigma_{k,d,.}$ is updated in frame I_{t+1} using the following

online equation:

$$\sigma_{k,d,t+1} = \Big[(1-\phi) * (\sigma_{k,d,t})^{\lambda_{k,d,t}} + \phi * \lambda_{k,d,t+1} * A(\lambda_{k,d,t+1}) * (\tilde{X}_{k,d,t})^{\lambda_{k,d,t+1}}\Big]^{1/\lambda_{k,d,t+1}}. \tag{4.11}$$

The parameter ρ represents a learning rate in all the above equations where $\rho = \phi/\omega$ and ϕ is named the learning factor. This factor is adaptively estimated using the spatial information as explained in section 4.2.4. See Appendix A for a detailed description of the MoGG parameters derivations.

4.2.2 Temporal information co-occurrence and persistence modeling

In the original GMM-based BS and its variants [17], background models are constituted by the components with the largest weight values (i.e., ω parameters). This achieves good results only if background patterns are stable over time. The performance of the model will decrease in case of fast intermittent switching of object/background components over time. Fig. 4.4 illustrates this fact by taking two samples in different locations of the image. The video is 'fountain01' from the 'dynamicBackground' category of the CDnet dataset [142]. The first location is illustrated in Fig. 4.4-(a) where the background (grass) is well separated from the object (black car). Indeed, the grass component weight in the mixture overpasses 80%. The second location is illustrated in Fig. 4.4-(b) where the background is made of dynamic random appearances of the grass-ground and the water drop fountain. The rapid intermittent switching between the two has prevented the background from converging rapidly to the optimal one made of two components.

To circumvent this problem, we propose to use the *co-occurrence* and *persistence* of mixture components for accelerating convergence of the background model. Let p be a pixel at position (x, y), K is the number of components and $\mathbf{A}$ be a $K \times K$ matrix with each element a_{ij} giving the number of times the pixel p is labelled with components c_i and c_j at time t and $t+1$, respectively, with $1 \leq i, j \leq K$. We call $\mathbf{A}$ "component co-occurrence matrix" (CCM).

For illustration, consider the two scenarios presented in Fig. 4.4. Let us set $K = 7$ for the pixel p (the center of the square) with mixture component labels activated from frame 651 to frame 750. In Fig. 4.4-(a), the spotted pixel at position $(240, 110)$ represents a stable green grass ground with a black car passing over it. The matched MoGG temporal labels and the CCM $\mathbf{A}$ are given in Tab. 4.1-(a) and Tab. 4.1-(b), respectively. We can observe that the majority of the CCM entries are null and, consequently, all the co-occurrence weights in Tab. 4.1-(e) are also null. This is because there is practically one dominant stable background

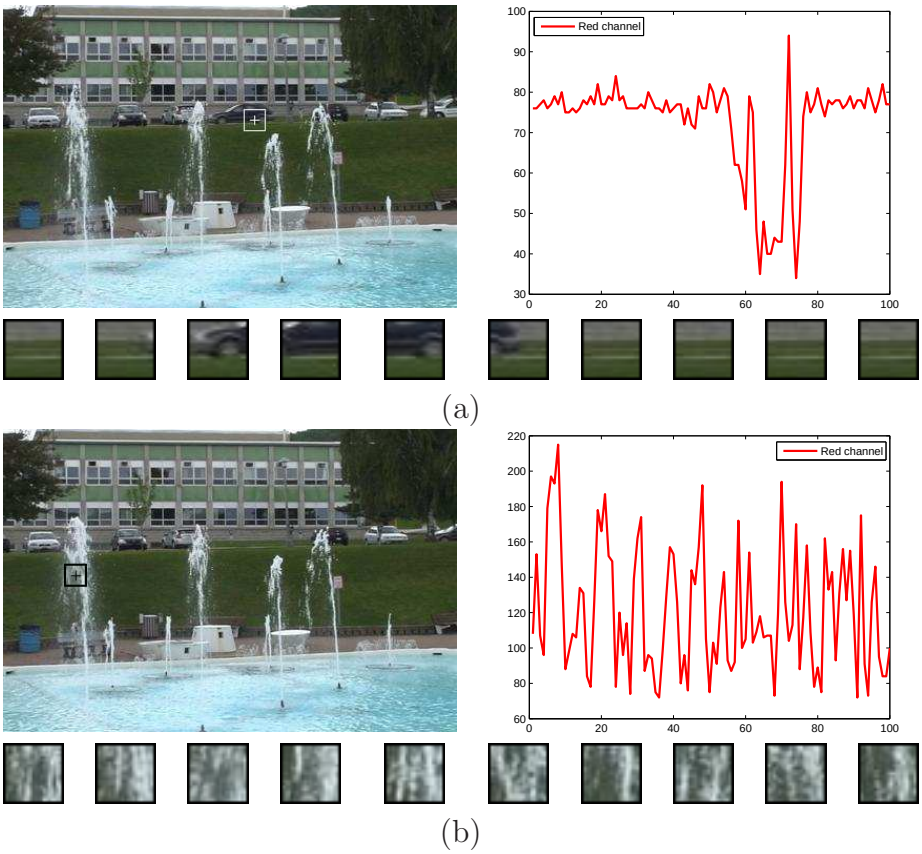


FIGURE 4.4: An illustration for static and dynamic backgrounds. (a) The pixel at the white box illustrates a static background. In the top left: a sample frame from the CDnet dataset [142], 'dynamicBackground' category, 'fountain01' sequence. In the top right: Red channel history of the spotted pixel (frames: 651 to 750). (b) A dynamic background illustrated by the pixel at the black box. Captions are the same as in (a).

component (the grass) and there is little co-occurrence with the other foreground components (the black car crossing the road).

In Fig. 4.4-(b), the selected pixel is characterized by rapid intermittent switch between the grass ground and water of the fountain. The correspondent timeline labels and CCM are given in Tab. 4.2-(a) and Tab. 4.2-(b), respectively. For example, a_{11} (top left) is the number of times that the pixel p with component c_1 appears after the same component and a_{14} (top middle) is the number of times that the pixel p with component c_4 appears after component c_1. Indeed, the pixel switches rapidly between several components which can explain why all CCM entries are relatively high.

Component *persistence* is a complementary concept to the co-occurrence information. At a given position p, we keep for each mixture component c a count

Frames	Activated components																			
651 - 670	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
671 - 690	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
691 - 710	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	3	3
711 - 731	4	4	3	5	3	5	5	3	3	3	2	6	6	7	3	1	1	1	1	1
731 - 750	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(a)

650	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

(b)

729	1	0	0	0	0	0
0	1	1	0	0	1	0
1	1	3	1	2	0	0
0	0	1	1	0	0	0
0	0	2	0	1	0	0
0	0	0	0	0	1	1
0	0	1	0	0	0	0

(c)

c_1	0.9964
c_2	0.0006
c_3	0.0019
c_4	0.0003
c_5	0.0006
c_6	0.0003
c_7	0.0000

(d)

0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000

(e)

0.9964
0.0006
0.0019
0.0003
0.0006
0.0003
0.0000

(f)

TABLE 4.1: Co-occurrence model related to the example at Fig. 4.4-(a). (a) MoGG activated component labels. (b) CCM at frame #651. (c) CCM at frame #750. (d) MoGG temporal weights (ω_i). (e) Co-occurrence weights (η_i). (f) Combined MoGG and Co-occurrence weights (π_i).

Frames	Activated components																			
651 - 670	1	1	1	1	2	6	3	7	1	1	1	1	1	1	1	1	1	1	2	1
671 - 690	3	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
691 - 710	1	1	1	1	1	1	1	3	1	1	1	1	1	1	1	1	1	1	1	1
711 - 731	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1	1	1	1	1
731 - 750	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1

(a)

504	9	9	13	7	7	1
8	0	0	0	0	0	0
9	0	0	0	0	0	0
14	0	0	3	1	0	0
6	0	0	2	1	0	0
7	0	0	0	0	2	0
0	1	0	0	0	0	0

(a)

586	13	12	13	7	7	1
11	0	0	0	0	1	0
12	0	0	0	0	0	1
14	0	0	3	1	0	0
6	0	0	2	1	0	0
7	0	1	0	0	2	0
1	1	0	0	0	0	0

(b)

c_1	0.9322
c_2	0.0338
c_3	0.0165
c_4	0.0051
c_5	0.0024
c_6	0.0094
c_7	0.0006

(c)

0.0000
0.2353
0.2353
0.2647
0.1275
0.1373
0.0000

(d)

0.4661
0.1346
0.1259
0.1349
0.0649
0.0733
0.0003

(e)

TABLE 4.2: Co-occurrence model related to the example at Fig. 4.4-(b). Captions as in Table 4.1.

reflecting the number of successive occurrences of c in time. In other words, if c is matched in two successive frames t and $t+1$, its persistence is incremented by 1, otherwise it is reset to 1. It follows that stable components (background or foreground) will tend to have high persistence values. A K-element vector υ is generated at each position p where each element υ_i, $i = 1, ..., K$, represents the correspondent component persistence value. After updating the CCM and extracting

the co-occurrence weights η_i as detailed in Algorithm 3, we divide each η_i by its corresponding persistence value υ_i. The new vector entries η_i encodes approximate probabilities of the components switching, in a similar way as the Markov chain transition matrix. These weights are combined with the MoGG temporal weights ω_i to build new component weights π_i. Contrarily to past works based on GMMs [17], our model assigns high weights to components that occur successively in time or have high switching rate with other components. This allows to significantly reduce false positives caused by background dynamics, such as fast swaying tree leaves, fountains, camera jitter, etc. The detailed procedure for updating the CCM and combining the MoGG and co-occurrence weights is outlined in Algorithm 3.

Algorithm 3: Compute the temporal combined weights π_i.

Data: MoGG models, co-occurrence matrices (CCM), persistence (υ), K, M.
Result: Component temporal combined weights π.
for *each pixel* p **do**
  Let $\mathbf{A}$ *be the CCM and* c_{old}, c_{new} *be the old and new activated components at pixel* p;
  $\mathbf{A}(c_{old}, c_{new}) \leftarrow \mathbf{A}(c_{old}, c_{new}) + 1$; `// update the CCM A`
  if $c_{old} = c_{new}$ **then**
    $\upsilon_i \leftarrow \upsilon_i + 1$; `// update the persistence vector`
  else
    $\upsilon_i \leftarrow 1$;
  end
  $\mathbf{B} \leftarrow (\mathbf{A} + \mathbf{A}^T)/2$; `// average between A and its transpose` $\mathbf{A}^T$
  $diag(\mathbf{B}) \leftarrow 0$; `// reset the diagonal of matrix B to zero`
  for *each component index* $i = 1 : M$ **do**
    let C *be the set of* M *component indices in descending order of* ω_i *weights*;
    `// take the co-occurrence weights` η_i `as the` c`-th row from` $\mathbf{B}$
    $\eta_i \leftarrow \mathbf{B}(c, :)$ where $c \leftarrow C(i)$;
    `// divide the weights` η_i `by the persistence` υ_i
    $\eta_i \leftarrow \eta_i / \upsilon_i$; normalize: $\eta_i \leftarrow \eta_i / (\sum_j \eta_j)$;
    `// compute the combined temporal co-occurrence weights`
    $\pi_i \leftarrow \nu\omega_i + (1 - \nu)\eta_i$; normalize: $\pi_i \leftarrow \pi_i / (\sum_j \pi_j)$;
  end
  $\pi \leftarrow \frac{1}{M} \sum_{i=1}^{M} \pi_i$;`// compute the final combined weights`
end

4.2.3 Spatial information modeling

Spatial information is added to our approach using local structure (or texture) and color distribution. This information is relatively stable under soft shadows and illumination changes and will enforce our BS to overcome these challenges.

4.2.3.1 Correlation analysis

The spatial structure conformity with background is done by multi-scale correlation analysis between patches. We recall that the normalized cross correlation (NCC) between two vectors $\vec{v}_1$ and $\vec{v}_2$ is defined as:

$$NCC(\vec{v}_1, \vec{v}_2) = \frac{\vec{v}_1 \cdot \vec{v}_2}{\parallel \vec{v}_1 \parallel \parallel \vec{v}_2 \parallel} \tag{4.12}$$

where $\parallel \vec{v} \parallel = \sqrt{\vec{v} \cdot \vec{v}}$ is the norm of $\vec{v}$. The NCC is invariant to linear scaling of the form $\vec{v'} = \gamma \vec{v}$, where $\gamma \in \mathbb{R}^*$. For our BS algorithm, for each pixel we approximate the current background reference by the mean of the component with the highest co-occurrence weight value. The resulting reference frame is named I and is given as follows:

$$I = \mu_{t+1} \left(\arg\max_i \left(\pi_{i,t+1} \right) \right). \tag{4.13}$$

Using the correlation between the reference frame I and the current frame I_{t+1}, we can compare the local structure between the current and the reference frames. We then derive an approximation of the spatial foreground/background probabilities. For more reliable estimation of these probabilities, we use multiple window sizes surrounding each pixel. Hence, S correlation maps are computed: NCC_1, NCC_2,..., NCC_S for square blocks of size $N_1 \times N_1$, $N_2 \times N_2$,..., $N_S \times N_S$, respectively, where $N_1 < N_2 < ... < N_S$ (typically $S = 3$). These maps are obtained by factorizing correlations over color channels as:

$$NCC_j = \prod_{d=1}^{D} NCC_{j,d}, \tag{4.14}$$

where $NCC_{j,d}$ corresponds to the correlation calculated using window size j and color channel d. Then, the maximum correlation among the different scales is retained:

$$NCC_f = \max_{j=1..S} NCC_j. \tag{4.15}$$

Note that to reduce the computational cost of computing the correlation maps, the integral image technique [135] is used. Finally, the spatial foreground and background probabilities are approximated using the lenient functions: $p_{s,f}(\vec{X}_{t+1}) \simeq \exp(-\xi * NCC_f)$ and $p_{s,b}(\vec{X}_{t+1}) \simeq 1 - \exp(-\xi * NCC_f)$, where ξ is a constant controlling the sensitivity of the probability to the spatial correlation.

Fig. 4.5 shows the NCC_f map obtained for a sample of frames from the Change Detection dataset [142]. The reference and original frames are shown in the first and second rows, respectively. The third row shows the obtained NCC_f map where darker regions are pixel surroundings which are to those of the reference frame. By opposite, brighter regions are pixel surroundings with high foreground spatial probability.

4.2.3.2 Histogram matching

Spatial information can also be exploited through local color distribution. This can be useful when a video contains dynamic backgrounds (i.e., waving trees, water fountain, camera jitters, etc.) where the local structure of the background may slightly change but not the color distribution. Suppose that we have a reference image I at the fame I_{t+1}. Let R and R_{t+1} be the regions centered around a pixel (x, y) in frames I and I_{t+1}, respectively, H and H_{t+1} their respective histograms and N_{bins} is the number of bins in the histograms H_t and H_{t+1}. We use the Bhattacharyya distance $d(H_t, H_{t+1})$ to compare H_t and H_{t+1} as follows:

$$d(H_t, H_{t+1}) = \sqrt{1 - \sum_{i=1}^{N_{bins}} \sqrt{H_t(i) * H_{t+1}(i)}}; \tag{4.16}$$

We compute the histograms at different scales using window sizes $W_1 < W_2 < ... < W_S$ (typically $S = 3$) and all D color channels. We use the integral histogram technique [101] to accelerate histogram calculation. The reference frame I is computed using Eq. (4.13). The histogram distance map $HIST_f$ is given by Eq. (4.17), where $HIST_{s,d}$ corresponds to the distance map calculated with reference I at scale s for color channel d.

$$HIST_f = \prod_{s=1}^{S} \left(\max_{d=1..D} HIST_{s,d} \right). \tag{4.17}$$

The histogram final distance map $HIST_f$ is multiplied by the correlation map $\exp(-\xi * NCC_f)$ to provide a final spatial map $SMAP_f = \exp(-\xi * NCC_f) * HIST_f$.

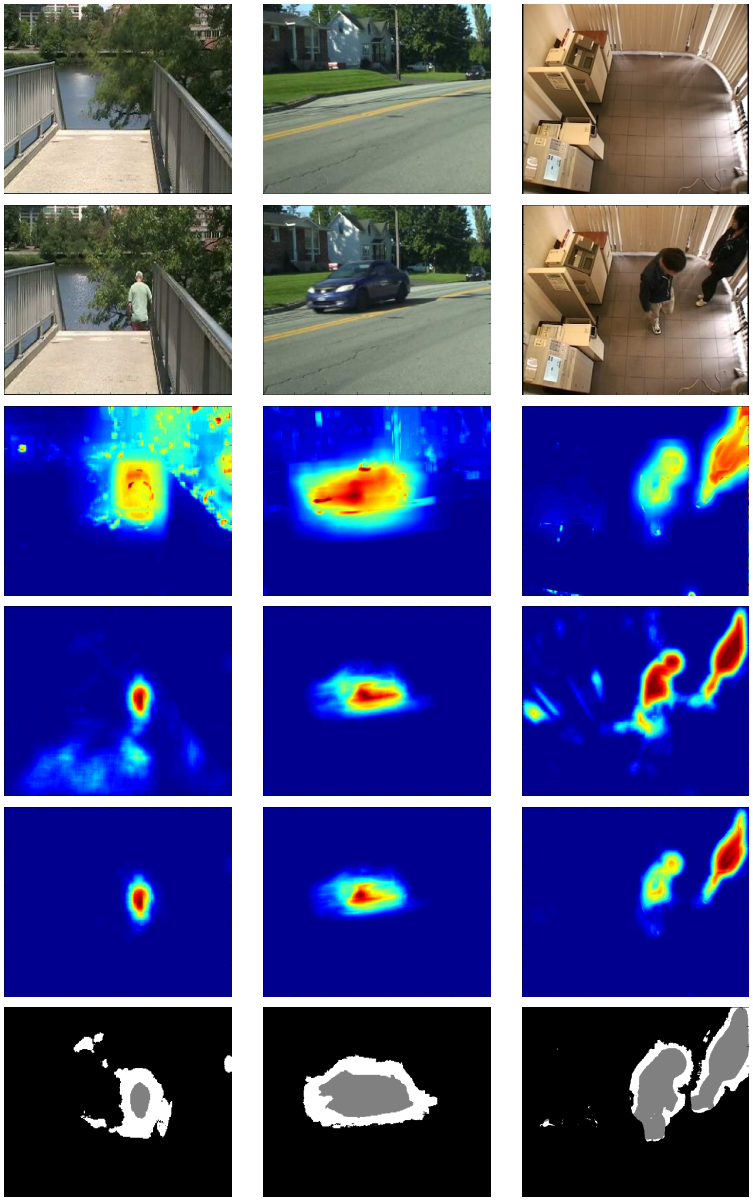


FIGURE 4.5: Spatial maps obtained for sample frames. Rows from top to down represent the reference frame, current frame, NCC map, histogram map, spatial combined map and learning factor map (the black, white and gray colors represent ϕ_{low}, ϕ_{avg} and ϕ_{high} respectively). (a) frame #2700 from 'overpass', (b) frame #850 from the 'bungalows' sequence and (c) frame #1080 from the 'copyMachine' sequence.

4.2.4 Adaptive learning rate for MoGG modeling

To obtain the learning factor used by each MoGG temporal model, a bi-level thresholding operation is carried out on the final spatial map $SMAP_f$ using two

thresholds T_1 and T_2. Consequently, the learning factor map should contain three levels of learning factors: $\phi_{low} < \phi_{avg} < \phi_{high}$ that correspond to high plausibility of foreground, unknown and high plausibility background, respectively. The dynamic estimation of the learning factor allows the temporal MoGG update procedure to assign a low learning factor ϕ_{low} to stopped objects or objects with slow motion. Background regions will be assigned a high learning factor ϕ_{high} that enables those regions to be quickly integrated in the background model. The third value ϕ_{avg} is assigned to pixels not identified strongly as either foreground or background according to the spatial map $SMAP_f$. The learning parameter ρ of Eq. (4.4) is updated for each pixel using the equation $\rho = \phi/\omega$.

Fig. 4.5 shows NCC maps, histogram matching and the resulting learning factor for the sample frames. We can observe that the correlation analysis (CA) and the histogram matching (HM) are complementary in their nature. For example, in the 'overpass' frame, CA detects the tree leaves as false positives. On the other hand, HM failed to remove the person shadow and illumination changes that caused false detections. However, we can observe that most of these false detections are removed by combining the two spatial maps and, therefore, a appropriate learning factor map has been generated. Finally, the stopped object challenge given by the waiting woman in the 'copyMachine' sequence can not be integrated in the background model due to the low learning factor ϕ_{low} assigned to the woman pixels.

4.2.5 Detection of PTZ camera effect scenarios

To detect the presence of PTZ camera effect in a given sequence, a global-level processing is performed based on the displacement estimation between two successive frames using cross-correlation. At each time t, both the previous and current frames are compared each other using a two-step cross-correlation algorithm. In the first step, we compute the correlation coefficients between overlapping patches of the two frames [124]. In the second step, we find the best-match positions using the maximum of cross-correlation values obtained for all possible displacement in the search matrix. Finally, PTZ effects are detected using the average of displacements calculated in a temporal window size L, which is formulated as:

$$d_{ptz} = \frac{1}{L} \sum_{i=t-L}^{t} \|\vec{d_i}\| \tag{4.18}$$

where $\vec{d_i}$ is the displacement vector computed between the two frames I_{i-1} and I_i.

The presence of a PTZ effect is decided by a threshold test on the displacement average d_{ptz}. If d_{ptz} overpasses a given threshold ϵ then a PTZ scenario is started

and, consequently, the learning rate is set to a high value α_{ptz} for the next N_{ptz} frames to allow the new background to be quickly absorbed to the background model.

4.2.6 The overall background subtraction algorithm

Suppose that at time t, the model in Eq. (4.2) has generated k_f and k_b components associated with the foreground (f) and background (b), respectively, where $k_f + k_b = K$. The foreground and background probabilities are given by $p_{t,f}(\vec{X}_t)$ and $p_{t,b}(\vec{X}_t)$, respectively.

Firstly, the presence of PTZ camera effect is checked as explained in Section 4.2.5. If a PTZ scenario is started then a high value is assigned to the learning rate. Otherwise, the learning rate is updated as explained in Section 4.2.4. At each location in frame I_{t+1}, we may have one of the following scenarios:

1. If the vector $\vec{X}_{t+1}$ is matched with one of the mixture components, the matched component parameters ($\omega_{k,t+1}$, $\mu_{k,t+1}$, $\sigma_{k,t+1}$ and $\beta_{k,t+1}$) are updated using Eqs. (4.4) to (4.11). The CCM and persistence values are updated such as detailed in Section 4.2.2. Finally, the matched component's label (b:background or f:foreground) is assigned to the pixel.

2. If no match is found, a new component $K + 1$ is created for the mixture model with parameters set as follows: $\omega_{K+1,t+1} = \alpha$, $\mu_{K+1,d,t+1} = \vec{X}_{t+1}$, $\sigma_{K+1,d,t+1} = \sigma_0$ and $\lambda_{K+1,d,t+1} = \lambda_0$, where σ_0 and λ_0 are initial scale and shape parameters, respectively. The CCM and persistence entries that correspond to this component are reset to zero.

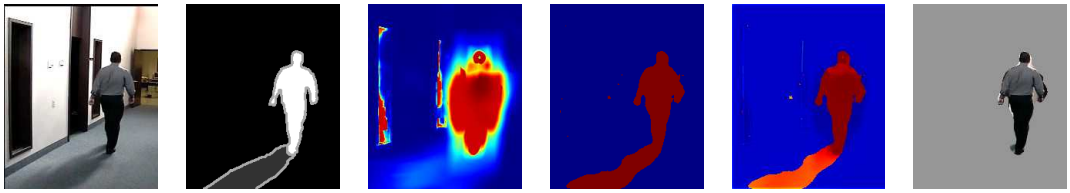
Next, the mixture components are sorted in descending order of temporal co-occurrence weights ($\pi_{i,t+1}$) computed by Algorithm 3 and the new background temporal model $p_{t+1,b}(.)$ is formed using the first B largest mixture components, where we have.

$$B = \arg\min_{b} \left(\sum_{i=1}^{b} \pi_{i,t+1} > T \right). \tag{4.19}$$

The threshold T is the minimum portion of data considered to belong to the dynamic background (typically $T = 0.8$). At this step, the reference frame used by the spatial module is constructed by taking the mean parameter of the first component among the B components resulting in Eq. (4.19) as the pixel value (see Eq. (4.13)).

Next, based on the two temporal and spatial probabilities (i.e. $p_{t,f}$ and $p_{s,f}$, respectively), the current pixel will be assigned label f if:

$$\mathcal{S}_{fcm} * p_{s,f}(\vec{X}_{t+1}) + (1 - \mathcal{S}_{fcm}) * p_{t,f}(\vec{X}_{t+1}) \geq \delta \tag{4.20}$$



(a) (b) (c) (d) (e) (f)

FIGURE 4.6: Shadow removing using correlation analysis (Sample frame #1200 from the 'CDnet' dataset, 'shadow' category, 'cubicle' sequence). (a) Original frame. (b) Ground truth frame. (c) Temporal map. (d) Spatial correlation map. (e) Temporal and spatial combination map. (f) Final detection.

where δ is a threshold and $\mathcal{S}_{fcm}$ (spatial foreground coherence map) is a smoothed map of the correlation map $p_{s,f}$ using an average filter of 5×5 pixels size. Otherwise, it is assigned label b (i.e., a potential shadow or highlight).

Finally, a post-processing step is performed through the binary masks generated after the temporal and spatial combination. The "salt and pepper noise" is a common problem that arise in BS. First, we apply the median filter to reduce this type of noise. Then, a morphological correction is applied to smooth silhouettes and fill their "internal holes". This allows also to remove small wrongly detected artifacts from the resulting binary masks.

The combination of temporal and spatial models is demonstrated in Fig. 4.6, where the correlation analysis is used to remove casting shadows. We can see that the shadow casted by the walking person is wrongly classified as foreground by the temporal model. However, by using the spatial information, the shadow is detected and removed. On the other hand, the temporal model helps to precisely detect the the walking person silhouette.

4.3 Experimental results

We present experiments conducted by using two benchmark datasets: Change Detection Dataset [142] and SABS [22] datasets. To assess the merit of our approach, we compared our obtained results with those produced by recent state-of-the-art methods.

4.3.1 Parameter setting

Our tests were implemented using the Matlab environment with some optimization using MEX C++ subroutines. On a PC with Intel Core i7 2.93 GHz CPU and 16 Go of RAM and and MS Windows 7 operating system, the proposed prototype

Dataset/Module	MoGG	Co-occurrence	NCC analysis	Histogram matching	PTZ detection
CDnet, SABS.	$K = 7$, $\tau = 0.002$, $\sigma_0 = 20.0$, $\lambda_0 = 1.0$, $\phi_{low} = 10^{-5}$, $\phi_{avg} = 10^{-3}$, $\phi_{high} = 0.05$.	$M = 1$, $\nu = 0.5$, $T = 0.80$.	$N_1 = 11$, $N_2 = 35$, $N_3 = 65$, $\xi = 0.82$, $\delta = 0.9$.	$W_1 = 13$, $W_2 = 33$, $W_3 = 53$, $N_{bins} = 16$, $T_1 = 0.2^3$, $T_2 = 0.5^3$.	$L = 10$, $\epsilon = 6$, $N_{ptz} = 15$, $\alpha_{ptz} = 0.1$.

TABLE 4.3: Parameter setting of the proposed algorithm used in our experimentations.

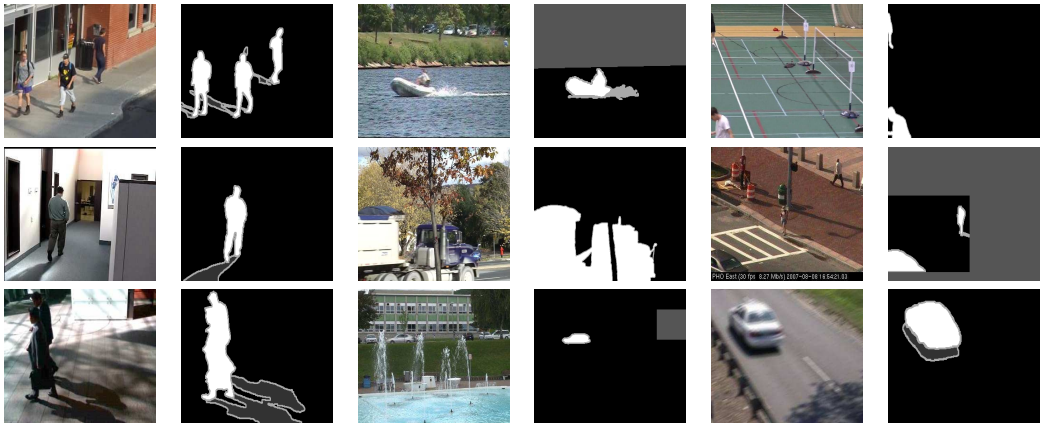
runs at 5 fps for sequences of RGB color frames with size of 320×240 pixels. Most of the processing time is dedicated to the multi-scale correlation and histogram calculation and the updating procedure for the MoGG temporal model. The choice of optimal parameters is critical to the evaluation task. Preliminary experiments have been conducted on the datasets to adjust the best set of parameters. Tab. 4.3 shows the optimal parameters selected for the proposed algorithm.

4.3.2 The Change Detection dataset

This dataset has been proposed recently in [44, 142] to address the shortcomings of previous datasets in terms of challenges and ground truth availability. It provides 53 videos with indoor and outdoor scenes with a variety of moving objets such as boats, cars, trucks and pedestrians (Fig. 4.7). The videos have been acquired in different scenarios: baseline, dynamic backgrounds, camera jitter, shadows, intermittent object motion, thermal, challenging weather, low frame-rate, night, PTZ camera motion and air turbulence. They are grouped into 11 categories according to the type of challenge each video exhibits. The spatial resolution of the videos varies from 320×240 to 720×576. The frame number of each sequence varies from 900 to 7000 frames. We run our algorithm on the whole dataset, but we have focused our attention to the following four challenges:

- *Shadows*: This category contains 2 indoor and 4 outdoor videos illustrating hard as and soft shadows. Some shadows are quite thin while others occupy most of the scene. Also, some shadows are cast by moving objects such as pedestrians and cars while others are cast by trees or buildings.

- *Dynamic background*: This category consists of 6 videos depicting outdoor scenes with strong background motion. 2 videos show boats on flowing water, 2 videos show cars passing next to a fountain, and the last 2 show pedestrians, cars and trucks passing in front of a tree shaken by the wind.



(a) (b) (c) (d) (e) (f)

FIGURE 4.7: Sample frames from the Change Detection dataset. From top to down: (a) 'shadow' category: frames #1030, #1630 and #1100 from the 'busStation', 'cubicle' and 'peopleInShade' sequences respectively. (b) Ground truth for the frames in column (a). (c) 'dynamicBackground' category: frames #2000, #2600 and #1150 from 'boats', 'fall' and 'fountain01' sequences respectively. (d) Ground truth for the frames in column (c). (e) 'cameraJitter' category: frames #700, #1055 and #1110 from the 'badminton', 'sidewalk' and 'traffic' sequences respectively. (f) Ground truth for the frames in column (e).

- *Camera jitter*: This category contains one indoor and 3 outdoor videos captured by vibrating cameras. Jitter magnitude varies from one video to another.
- *PTZ*: this category includes 4 videos with the following PTZ motion modes: slow continuous camera pan, intermittent pan, two-position patrol-mode PTZ, and zoom-in/zoom-out.

4.3.3 Quantitative study of the proposed temporal/spatial modules

Tab. 4.4 shows the results obtained by the application of the proposed modules separately for the 'shadow' and 'dynamicBackground' categories of CDnet dataset. We note that the videos in the same CDnet dataset category may include different kinds of challenges. We can observe that, for the shadow 'category', the use of MoGGs with co-occurrence gives similar results in terms of F-measure compared to using only the temporal MoGGs. However, the co-occurrence model helps to improve the precision for the 'dynamicBackground' category by removing false detections results from the rapid switching background that characterizes videos

Category	Method	Re	Sp	FPR	FNR	PWC	Pr	F
shadow	MoGG	0.8365	0.9852	0.0148	0.1635	2.2438	0.7202	0.7642
	MoGG+CooC	0.7895	0.9887	0.0113	0.2105	2.1345	0.7614	0.7632
	MoGG+CooC+Spatial modules	0.9610	0.9908	0.0092	0.0390	1.0220	0.8490	0.8997
dyn.Back.	MoGG	0.8590	0.9956	0.0044	0.1410	0.5780	0.6356	0.6895
	MoGG+CooC	0.8274	0.9977	0.0023	0.1726	0.4134	0.7104	0.7340
	MoGG+CooC+Spatial modules	0.9224	0.9987	0.0013	0.0776	0.1868	0.8408	0.8749

TABLE 4.4: Evaluation metrics obtained by the different proposed modules for the 'shadow' and 'dynamicBackground' categories from CDnet dataset.

of this category. We can also observe that the spatial information has improved both recall and precision metrics in the two categories. This is justified, on the one hand, by the correlation analysis that helps to increase the precision by removing false detections caused by shadows and illumination changes. On the other hand, adaptive updating of the learning rate by the histogram matching improves the recall evaluation metric by preventing the foreground objects to be absorbed in the background model.

4.3.4 Quantitative results for the proposed method

To conduct a quantitative comparison between the proposed model and recent state-of-the-art approaches, we use the evaluation metrics provided by the CDnet dataset [44, 142]. The seven metrics used are Recall (Re), Specificity (Sp), False Positive Rate (FPR), False Negative Rate (FNR), Percentage of Wrong Classifications (PWC), Precision (Pr) and F-measure (F).

Tab. 4.5 and 4.6 show obtained values of the above metrics using the proposed model compared to a set of recent methods published in the Change Detection website (July 2014). To show the merit of our algorithm to deal with the above-selected challenges, tests have been conducted on all videos of these categories. Tab. 4.5 gives results obtained by the proposed model for each video sequence from the aforementioned categories. We can note that the majority of videos show F-Measure metrics above 80% which indicates that the proposed approach is efficient in dealing with shadows and dynamic backgrounds. For the 'PTZ' category, the proposed algorithm gives good results for three sequences, namely: the 'continuousPan', the 'intermittentPan', and the 'twoPositionPTZCam' sequences. However, the proposed technique fails to detect the PTZ scenario for the 'zoomInZoomOut' sequence. This is due the fact that, in this video, the displacement between two consecutive frames is very small and therefore the PTZ detection and processing can not be started throughout all the frames of this sequence. Note that the results for all the CDnet videos given by the proposed approach are listed in the Appendix B, Table B.1.

Category	Video	Re	Sp	FPR	FNR	PWC	Pr	F
shadow	backdoor	0.9712	0.9992	0.0008	0.0006	0.0013	0.9633	0.9672
	bungalows	0.9954	0.9783	0.0217	0.0003	0.0207	0.7455	0.8525
	busStation	0.9351	0.9927	0.0073	0.0025	0.0094	0.8313	0.8802
	copyMachine	0.9369	0.9903	0.0097	0.0047	0.0134	0.8782	0.9066
	cubicle	0.9292	0.9967	0.0033	0.0014	0.0046	0.8500	0.8878
	peopleInShade	0.9983	0.9874	0.0126	0.0001	0.0120	0.8258	0.9039
dynamicBackground	boats	0.9786	0.9996	0.0004	0.0001	0.0005	0.9386	0.9582
	canoe	0.9583	0.9988	0.0012	0.0015	0.0027	0.9657	0.9620
	fall	0.9608	0.9961	0.0039	0.0007	0.0045	0.8172	0.8832
	fountain01	0.7825	0.9993	0.0007	0.0002	0.0008	0.4940	0.6056
	fountain02	0.9020	0.9999	0.0001	0.0002	0.0003	0.9463	0.9236
	overpass	0.9524	0.9983	0.0017	0.0006	0.0023	0.8833	0.9166
cameraJitter	badminton	0.8721	0.9979	0.0021	0.0045	0.0064	0.9368	0.9033
	boulevard	0.7610	0.9904	0.0096	0.0118	0.0204	0.7955	0.7779
	sidewalk	0.9524	0.9918	0.0082	0.0013	0.0092	0.7577	0.8440
	traffic	0.8962	0.9809	0.0191	0.0069	0.0244	0.7571	0.8208
PTZ	continuousPan	0.4199	0.9986	0.0014	0.0037	0.0050	0.6560	0.5120
	intermittentPan	0.8221	0.9985	0.0015	0.0025	0.0040	0.8879	0.8537
	twoPositionPTZCam	0.8701	0.9942	0.0058	0.0020	0.0077	0.6972	0.7741
	zoomInZoomOut	0.9979	0.4324	0.5676	0.0000	0.5664	0.0037	0.0074

TABLE 4.5: Evaluation performance metrics obtained by the application of the proposed method for the 'shadow', 'dynamicBackground', 'cameraJitter' and 'PTZ' category sequences.

Tab. 4.6 shows average values of the metrics for each category obtained by compared methods. The set of compared methods includes: (i) the five best ranked methods at the Change Detection 2014, namely: FTSG (Flux Tensor with Split Gaussian models) [140], SuBSENSE (Self-Balanced SENsitivity SEgmenter) [27], CwisarDH [46], Spectral360 [117], BinWang [138]. (ii) three state-of-the-art methods include the GMM (Gaussian Mixture Model) by Grimson et *al.* [125], the GMM (Gaussian Mixture Model) by Zivkovic et *al.* [160] and KDE (Kernel Density Estimation) [33] and (iii) the proposed approach (the highest results are shown in bold). We can observe that for the 'shadow' and 'cameraJitter' categories, the proposed approach gives the best results and outperforms most of the compared methods in terms of recall, PWC and F-measure. However, for the 'dynamicBackground' category, the proposed method can be ranked second after the FTSG method which is ranked first at the CDnet 2014 overall challenges. For the 'PTZ' category, our method gives the highest precision and F-measure metrics along with a good results in terms of the recall metric.

Tab. 4.7 shows results obtained by the proposed algorithm on all the the CDnet dataset categories as well as the overall results computed according the CDnet

Category	Method	Re	Sp	FPR	FNR	PWC	Pr	F
shadow	Proposed	**0.9610**	0.9908	0.0092	**0.0390**	**1.0220**	0.8490	**0.8997**
	FTSG [140]	0.9214	**0.9918**	**0.0082**	0.0786	1.1305	**0.8535**	0.8832
	SuBSENSE [27]	0.9529	0.9910	0.0090	0.0471	1.0668	0.8370	0.8890
	CwisarDH [46]	0.8786	0.9910	0.0090	0.1214	1.2770	0.8476	0.8581
	Spectral360 [117]	0.8898	0.9893	0.0107	0.1102	1.5682	0.8187	0.8519
	BinWang [138]	0.8297	0.9914	0.0086	0.1703	1.7537	0.8098	0.8128
	Stauffer [125]	0.7960	0.9871	0.0129	0.2040	2.1951	0.7156	0.7370
	Zivkovic [160]	0.7774	0.9878	0.0122	0.2226	2.1908	0.7232	0.7322
	KDE [33]	0.8541	0.9885	0.0115	0.1459	1.6844	0.7660	0.8030
dynamicBackground	Proposed	**0.9224**	0.9987	0.0013	**0.0776**	**0.1868**	0.8408	**0.8749**
	FTSG	0.8691	**0.9993**	**0.0007**	0.1309	0.1887	**0.9129**	**0.8792**
	SuBSENSE	0.7872	**0.9993**	**0.0007**	0.2128	0.3837	0.8768	0.8138
	CwisarDH	0.8144	0.9985	0.0015	0.1856	0.3270	0.8499	0.8274
	Spectral360	0.7819	0.9992	0.0008	0.2181	0.3513	0.8456	0.7766
	BinWang	0.9177	0.9956	0.0044	0.0823	0.4837	0.7990	0.8436
	Stauffer	0.8344	0.9896	0.0104	0.1656	1.2083	0.5989	0.6330
	Zivkovic	0.8019	0.9903	0.0097	0.1981	1.1725	0.6213	0.6328
	KDE	0.8012	0.9856	0.0144	0.1988	1.6393	0.5732	0.5961
cameraJitter	Proposed	**0.8704**	0.9903	0.0097	**0.1296**	**1.5086**	0.8118	**0.8365**
	FTSG	0.7717	0.9866	0.0134	0.2283	2.0787	0.7645	0.7513
	SuBSENSE	0.7495	0.9908	0.0092	0.2505	1.8282	0.8116	0.7694
	CwisarDH	0.7437	0.9931	0.0069	0.2563	1.7058	**0.8516**	0.7886
	Spectral360	0.6696	0.9906	0.0094	0.3304	2.0855	0.8387	0.7142
	BinWang	0.6505	**0.9938**	**0.0062**	0.3495	1.9125	0.8493	0.7107
	Stauffer	0.7334	0.9666	0.0334	0.2666	4.2269	0.5126	0.5969
	Zivkovic	0.6900	0.9665	0.0335	0.3100	4.4057	0.4872	0.5670
	KDE	0.7375	0.9562	0.0438	0.2625	5.1349	0.4862	0.5720
PTZ	Proposed	0.7775	0.8559	0.1441	0.2225	14.5767	**0.5612**	**0.5368**
	FTSG	0.6621	0.9814	0.0186	0.3379	2.1983	0.3411	0.3712
	SuBSENSE	**0.8244**	0.9435	0.0565	**0.1756**	5.7906	0.2949	0.3507
	CwisarDH	0.3833	**0.9968**	**0.0032**	0.6167	**0.9013**	0.4974	0.3410
	Spectral360	0.5584	0.9160	0.0840	0.4416	8.6236	0.3653	0.4016
	BinWang	0.5162	0.8808	0.1192	0.4838	12.4379	0.2085	0.1575
	Stauffer	0.6360	0.8294	0.1706	0.3640	17.3651	0.1248	0.1602
	Zivkovic	0.5954	0.7927	0.2073	0.4046	21.0492	0.0659	0.1041
	KDE	0.8086	0.6687	0.3313	0.1914	33.0496	0.0249	0.0476

TABLE 4.6: Evaluation metrics obtained by state-of-the-art as well as proposed method for the 'shadow', 'dynamicBackground', 'cameraJitter' and 'PTZ' categories from CDnet dataset.

evaluation methodology [142]. To make a comparison with a method combining temporal and spatial information, Tab. 4.8 shows results obtained by application of a patch-based approach proposed in [107]. From Tabs. 4.7 and 4.8, we can observe that our method achieves better results in terms of overall F-measure and overall precision metrics than [107]. However the patch-based approach surpasses our method in only the Recall metric. This is due to the fact that [107] does not deal explicitly with background subtraction challenges such as illumination changes, shadow, dynamic background and PTZ challenges.

Category/Metric	Re	Sp	FPR	FNR	PWC	Pr	F
badWeather	0.7079	0.9987	0.0013	0.2921	0.5504	0.8947	0.7815
baseline	0.9419	0.9934	0.0066	0.0581	0.7592	0.8600	0.8956
cameraJitter	0.8704	0.9903	0.0097	0.1296	1.5086	0.8118	0.8365
dyn. Back.	0.9224	0.9987	0.0013	0.0776	0.1868	0.8408	0.8749
int.Obj.Mot.	0.4744	0.9077	0.0923	0.5256	11.8726	0.5810	0.3885
lowFramerate	0.6396	0.9942	0.0058	0.3604	1.7770	0.5977	0.5785
nightVideos	0.5501	0.9812	0.0188	0.4499	2.8090	0.3969	0.4372
PTZ	0.6396	0.9941	0.0058	0.3603	1.7770	0.5977	0.5785
shadow	0.9610	0.9908	0.0092	0.0390	1.0220	0.8490	0.8997
thermal	0.7681	0.9930	0.0070	0.2319	1.3989	0.8771	0.7727
turbulence	0.7949	0.9977	0.0023	0.2051	0.3710	0.7136	0.6943
Overall	0.7643	0.9728	0.0271	0.2356	3.3484	0.7258	0.7001

TABLE 4.7: Evaluation performance metrics obtained by application of the proposed method for all CDnet categories as well as the overall results.

Category/Metric	Re	Sp	FPR	FNR	PWC	Pr	F
badWeather	0.3939	0.9972	0.0028	0.6061	1.1915	0.8088	0.4557
baseline	0.9742	0.9966	0.0034	0.0258	0.4305	0.9064	0.9384
cameraJitter	0.8480	0.9856	0.0144	0.1520	2.0182	0.7203	0.7784
dyn. Back.	0.8982	0.9927	0.0073	0.1018	0.8253	0.6294	0.6823
int.Obj.Mot.	0.7444	0.8469	0.1531	0.2556	14.4944	0.4493	0.4955
lowFramerate	0.9177	0.9822	0.0178	0.0823	1.9655	0.5266	0.5887
nightVideos	0.8174	0.9600	0.0400	0.1826	4.2120	0.3871	0.4933
PTZ	0.7887	0.7613	0.2387	0.2113	23.8913	0.0491	0.0896
shadow	0.9792	0.9879	0.0121	0.0208	1.2570	0.7856	0.8669
thermal	0.3169	0.9920	0.0080	0.6831	6.1843	0.7035	0.3957
turbulence	0.8081	0.9992	0.0008	0.1919	0.2262	0.7483	0.7680
Overall	0.7715	0.9547	0.0453	0.2285	5.1542	0.6104	0.5957

TABLE 4.8: Evaluation performance metrics obtained by the application of the Reddy method [107] for all CDnet categories as well as the overall results.

Fig. 4.8 shows a sample of foreground masks generated by each compared method. The original frames and ground truth masks are shown in the first and second rows, respectively. The third row shows the foreground masks given by our proposed method. The rest of methods in Tab. 4.6 are shown in the other rows. Columns of Fig. 4.8 represent 6 sample sequences from the studied categories. These sequences are as follows: 'traffic' and 'sidewalk' from the 'cameraJitter' category, 'canoe' and 'fountain01' from the 'dynamicBackground' category and 'busStation' and 'cubicle' from the 'shadow' category. The sequences in the first and second column are characterized by unstable cameras. We can observe that

our method significantly avoids false positives caused by camera jitter and efficiently detects the car in the 'traffic' sequence as well as the waiting person in the 'sidewalk' video.

The dynamic background challenge is presented in columns 3 and 4. The 'fountain01' sequence contains a water fountain and cars moving over the dynamic background. The 'canoe' sequence represents a water rippling situation. As can be seen, most of the false positive detections are eliminated by the proposed algorithm. The fifth column shows a sample frame from the 'busStation' sequence which consists of persons waiting in a bus station. This sequence is characterized by a hard shadow casted on the ground by the walking persons. Our approach detects accurately the walking man and separates a great amount of shadow from the ground. The last column shows a sample frame from the 'cubicle' sequence. This video sequence contains a person who walks through a cubicle corridor. This sequence is characterized by a hard shadow casted on the ground by the walking person as well as some highlights on the cubicle walls. We can note that the proposed method detects the person with a minimal amount of false detections due to shadows. These sample frames clearly show the advantage of combining spatial and temporal information to deal with the challenges such as casting shadows, illumination changes, dynamic backgrounds and camera jitter.

4.3.5 The SABS dataset

This dataset proposed in [22] contains synthetic videos for pixel-wise evaluation of BS methods. It includes 9 realistic scenarios: basic, dynamic background, bootstrapping, darkening, light switch, noisy night, camouflage, no Camouflage and Video compression. Each video sequence exhibits one or more challenges such as shadows, waving trees and traffic lights. The sequences have a resolution of 800×600 pixels and were acquired from a fixed viewpoint. High quality ground truth annotation is provided as color-coded foreground masks for every frame of each test video. Fig. 4.9 shows sample frames from this dataset.

To demonstrate the accuracy of the proposed model, experiments are conducted on the SABS dataset using the proposed method compared to the nine state-of-the-art methods cited in the SABS dataset website [1] which include: ViBe [8], SOBS [79], Zivkovic [159], Kim [63, 64], Li [72], McKenna [87], Oliver [96], Stauffer [125] and McFarlane [84]. In addition, we add a recently published method: SuBSENSE [27].

Tab. 4.9 presents quantitative results in terms of the maximal F-measure as cited in [22]. We can observe that our method gives competitive results for almost all scenarios. More precisely, (in terms of F-measure metric) the proposed method

[1] http://www.vis.uni-stuttgart.de/~hoeferbn/bse/

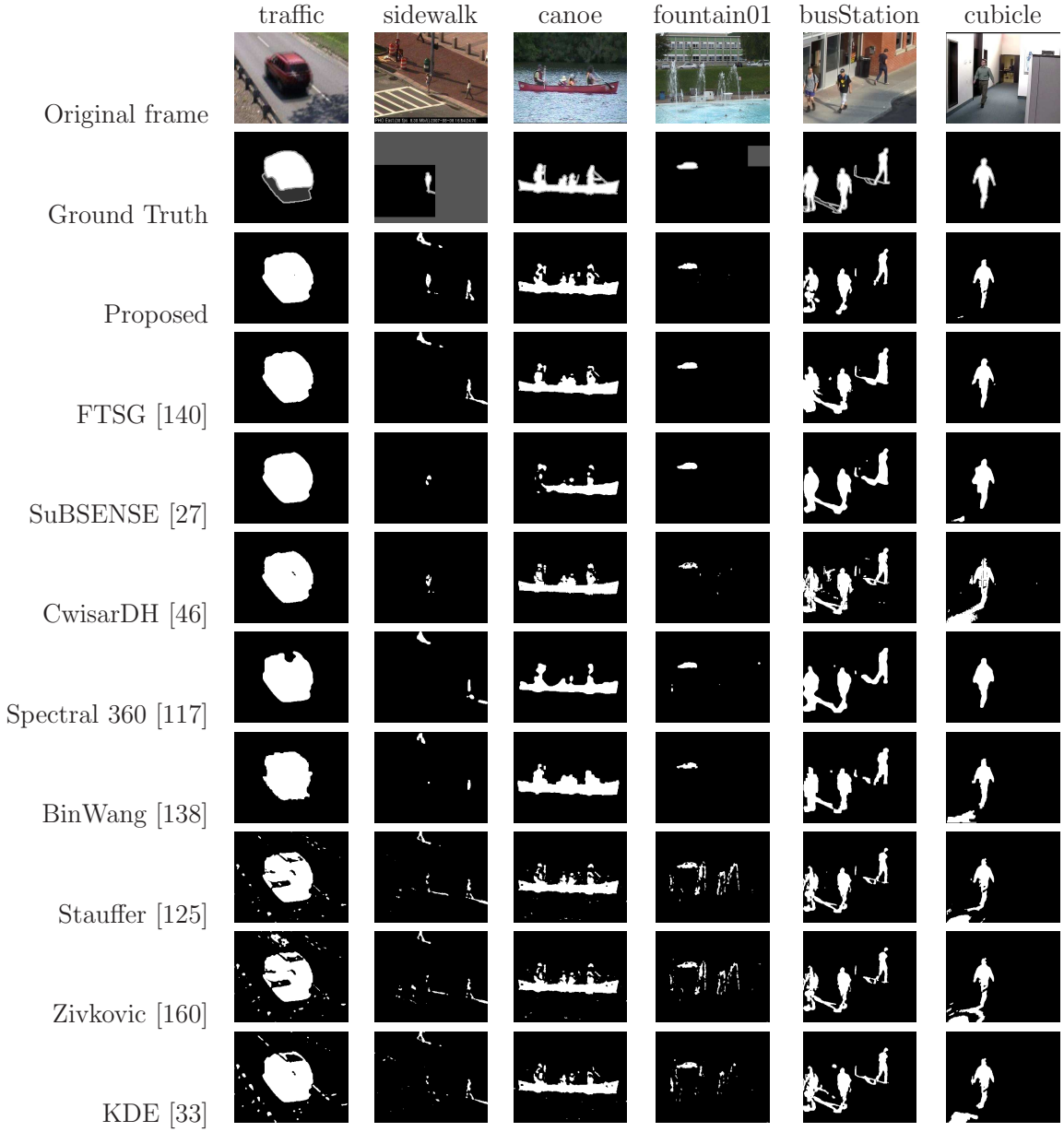
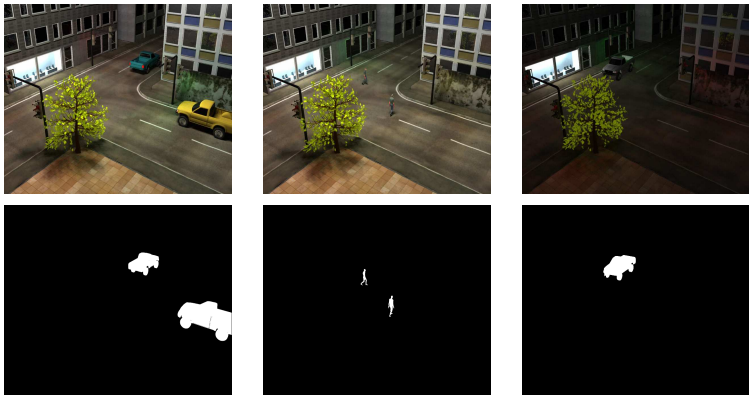


FIGURE 4.8: Background subtraction masks obtained for sample frames from the CDnet dataset [142] by application of different compared methods.

outperforms the compared ones for 5 sequences out of 9 which are: the Basic, Dynamic Background, Bootstrapping, Camouflage and H.264 (40kbps). However, the proposed method presents the second best F-measure evaluation for the 4 remaining sequences. Nonetheless, the average F-measure computed for all sequences is higher than the compared methods.



(a) (b) (c)

FIGURE 4.9: Sample frames from the SABS dataset. First row: (a) 'Basic' sequence: frame #105. (b) 'Camouflage' sequence: frame #315. (c) 'Light Switch' sequence: frame #375. Second row represents the ground truth masks.

Method	BA	DB	BO	DA	LS	NN	CA	NC	VC	AVG
Proposed	**0.860**	**0.821**	**0.815**	0.769	0.306	0.353	**0.844**	0.848	**0.846**	**0.718**
SuBSENSE [27]	0.813	0.735	0.772	**0.784**	0.286	**0.426**	0.843	**0.853**	0.792	0.700
ViBe [8]	0.761	0.711	0.685	0.678	0.268	0.271	0.741	0.799	0.774	0.632
SOBS [79]	0.766	0.715	0.495	0.663	0.213	0.263	0.793	0.811	0.772	0.610
Zivkovic [159]	0.768	0.704	0.632	0.620	0.300	0.321	0.820	0.829	0.748	0.638
Kim [63, 64]	0.582	0.341	0.318	0.342	-	-	0.776	0.801	0.551	0.530
Li [72]	0.766	0.641	0.678	0.704	**0.316**	0.047	0.768	0.803	0.773	0.611
McKenna [87]	0.522	0.415	0.301	0.484	0.306	0.098	0.624	0.656	0.492	0.433
Oliver [96]	0.635	0.552	-	0.300	0.198	0.213	0.802	0.824	0.669	0.524
Stauffer [125]	0.800	0.704	0.642	0.404	0.217	0.194	0.802	0.826	0.761	0.594
McFarlane [84]	0.614	0.482	0.541	0.496	0.211	0.203	0.738	0.785	0.639	0.523

TABLE 4.9: F-measure metrics obtained by application of the compared methods for the SABS dataset sequences [22]. The first column presents the compared methods and the rest of columns from left to right the sequences: Basic (BA), Dynamic Background (DB), Bootstrap (BO), Darkening (DA), Light Switch (LS), Noisy Night (NN), Camouflage (CA), No Camouflage (NC) and Video Compression (VC) H.264 codec with bitrate 40kbps/s. The last column represents the average of F-measure metric computed for each method for all sequences.

Fig. 4.10 shows some frames from the SABS dataset, the associated ground truth as well as the foreground masks obtained by the compared methods. It can be seen that the proposed model has effectively discriminated between backgrounds and moving objects. Indeed, thanks to adding multi-scale spatial information, the proposed model has been able to reduce false positives generated by the illumination changes casted on the wall by the traffic lights. On the other hand, parasites

generated by the waving tree, the Gaussian noise or due to compression artifacts are absorbed by the temporal co-occurrence module which has considerably improved precision.

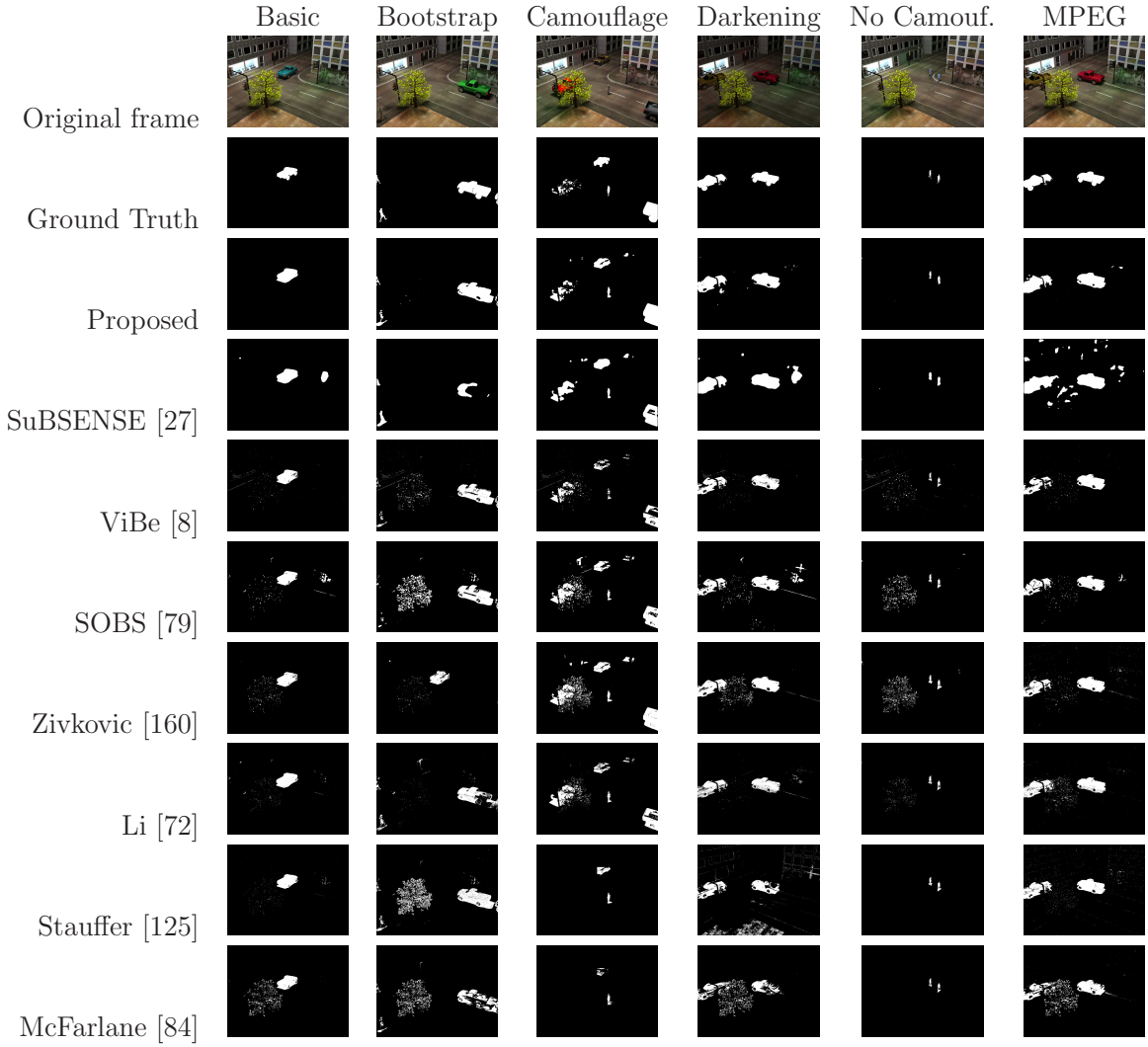


FIGURE 4.10: Background subtraction masks obtained for sample frames from the SABS dataset [22] by application of different compared methods.

4.4 Conclusion

A statistical approach for video background subtraction (BS) by combining temporal and spatial information is presented. The two types of information are fused

in an algorithm that performs efficient BS in the presence of cast shadows, illumination changes, complex background dynamics and camera jitter and PTZ effects. Our algorithm achieves accurate foreground detections compared to well-known methods. Future work will address analysis of hard shadows and other background subtraction challenging problems such as camouflage as well as speeding-up our algorithm.

5
Conclusion

The work in this thesis is aimed at dealing with the problem of foreground segmentation of still images and video sequences. The foreground segmentation is a significant problem confronted in numerous computer low-level vision applications. The proposed approaches are based on statistical modeling of image histograms and video adaptive backgrounds, although several other image processing operations are performed. Furthermore, the algorithms are designed to work reliably in uncontrolled environments with a view to achieving real-time performances.

To address the low-level task of foreground segmentation in still images, various methods proposed in the literature (e.g. the standard Otsu's method [97], Kittler and Illingworth's method [66], and Xue and Titterington [148]) assume a uni-modal histogram shape for classes and use the distribution parameters to approximate their distributions. The multi-modality that characterizes the background and/or foreground in real-world image segmentation has not been adequately explored. Another limitation of the standard methods lies in the assumption that class data is Gaussian. In several image examples, one can find histogram modes that are skewed, sharply peaked or heavy-tailed, making the assumption of Gaussian-distributed classes not realistic. As a result, the performance of the uni-modal class techniques can be affected by the class multi-modality or by the fact that the data distribution is non-Gaussian. For instance, in many foreground segmentation problems (e.g., medical images, and remote sensing), the image foreground and/or background region may have a multi-modal non-Gaussian distribution.

The first contribution of this work involves addressing the above-mentioned limitations by effectively performing segmentation for multi-modal classes with

arbitrarily shaped modes. The aforementioned state-of-art techniques based on using single probability density functions (pdf's) are generalized to mixtures of generalized Gaussian distributions (MoGG's). The merits of employing generalized to mixtures of generalized Gaussian include: (i) an additional degree of freedom that controls its kurtosis. (ii) histogram modes, ranging from sharply peaked to flat ones, can be accurately represented using this model. (iii) skewed and multi-modal classes are accurately represented using mixtures of GGDs.

The proposed objective function (described in Chapter 3) allows to find the optimal thresholds for multi-modal classes of data. It also extends easily to arbitrary numbers of classes ($K > 2$) with reasonable computational time. It has been successfully tested on segmentation of Non-Destructive Testing images (NDT-images) [118] and randomly generated data sets. Experiments have shown the performance of the proposed approach and showed that it can achieve more optimal thresholds than the standard Otsu's method [97], the median based extension method [148], as well as thresholding based on Gaussian mixture models.

In addition, the foreground segmentation is a significant task in many video processing applications. The state-of-the-art approaches in the video foreground segmentation are known to be sensitive to numerous phenomena, such as varying illumination, cast shadows, dynamic backgrounds and inherent image noise, which are inevitable in real-life surveillance videos. A statistical approach for foreground segmentation (background subtraction) of static camera videos is proposed to overcome these challenges (see Chapter 4). The proposed approach efficiently performs the foreground segmentation of image sequences by combining temporal and spatial information. These two types of information are integrated into an algorithm that performs effective foreground segmentation in the presence of a multitude of challenges such as cast shadows, illumination changes, complex background dynamics and PTZ effects. Experiments have been conducted using two datasets: the Change Detection dataset [142] and the SABS dataset [22]. The obtained evaluation results showed that the proposed algorithm achieves accurate foreground detections compared to well-known video background subtraction methods.

For future work, the proposed foreground segmentation approaches can be extended as follows:

- Extend the proposed multi-class histogram based approach in order to deal with color images. The foreground segmentation of color images is a prerequisite step in many vision applications, such as object segmentation, binarization, and segmentation of color documents, to cite a few.

- It is observed that no single algorithm can effectively perform for all image types. In order to improve the efficiency of foreground segmentation algorithm, one can exploit application-specific information from the foreground/background regions of the segmented images. For example, the color,

position and texture information of the foreground/background region can be used to guide the segmentation process.

- In order to generalize the image foreground segmentation algorithm, the objective function (used to select the optimal threshold) can be extended to more sophisticated forms. For example, we can use a multidimensional generalization of the mixture of generalized gaussian (MoGG) model by assuming a d-dimensional data vector [3]. In this case, the d-dimensional data vector can represent feature vector such as a 3-dimensional RGB pixel vector.

- Compute the class numbers (K_1, K_2) automatically using evaluation functions to control the foreground segmentation process and dynamically estimate a suitable parameters (K_1, K_2), based on a segmentation evaluation measure.

- The video foreground segmentation approach proposed in Chapter 4 performs the background subtraction process accurately with the assumption that the camera is static. The algorithm could be extended to deal with sequences captured from moved cameras.

- The performance of the video foreground segmentation algorithm in Chapter 4 can be improved by addressing the analysis of hard shadows and other background subtraction challenges such as camouflage and ghosting problems.

- The proposed co-occurrence model can be enhanced and extended to handle other features such as colors, histograms, and textures. For example, instead of considering the class component of each pixel in a fountain sequence, one can consider the co-occurrence of patch histogram in the presence of falling water drops. Furthermore, the co-occurrence model can be integrated with other standard BS models such as kernel density estimation (KDE) algorithm to overcome difficult background dynamics challenge.

- Speeding-up the proposed foreground segmentation algorithms by optimizing the proposed techniques and procedures, along with implementing them using sophisticated architectures such as parallel processors and GPUs.

A

Determination of the online update equations for the MoGG model

Considering the location and scale parameter equations given in [3], the online estimations given in the Eq. (4.8) and (4.11) are derived as follows:

A.1 The location parameter μ:

We have the following formula for estimating the location parameter at time t and $t+1$[3]:

$$\mu(t) = \frac{\sum_{i=1}^{t} |x_i - \mu|^{\lambda-2} x_i}{\sum_{i=1}^{t} |x_i - \mu|^{\lambda-2}} \tag{A.1}$$

and

$$\mu(t+1) = \frac{\sum_{i=1}^{t+1} |x_i - \mu|^{\lambda-2} x_i}{\sum_{i=1}^{t+1} |x_i - \mu|^{\lambda-2}}. \tag{A.2}$$

Therefore, we define:

$$\alpha(t) = \sum_{i=1}^{t} |x_i - \mu|^{\lambda-2} x_i \tag{A.3}$$

and

$$\beta(t) = \sum_{i=1}^{t} |x_i - \mu|^{\lambda-2}, \tag{A.4}$$

and by replacing Eqs. (A.3) and (A.4) in Eq. (A.2), we have

$$\mu(t+1) = \frac{\alpha(t) + |x_{t+1} - \mu|^{\lambda-2} x_{t+1}}{\lambda(t) + |x_{t+1} - \mu|^{\lambda-2}} \tag{A.5}$$

A.2 The scale parameter σ:

The scale parameter is defined by the formula:

$$\sigma(t) = \left[\frac{\lambda A(\lambda)}{t} \sum_{i=1}^{t} |x_i - \mu|^{\lambda} \right]^{1/\lambda} \tag{A.6}$$

For time $t+1$, we have:

$$\sigma(t+1) = \left[\frac{\lambda A(\lambda)}{t+1} \sum_{i=1}^{t+1} |x_i - \mu|^{\lambda} \right]^{1/\lambda}. \tag{A.7}$$

By replacing $\sum_{i=1}^{t} |x_i - \mu|^{\lambda}$ with its equivalent from Eq. (A.6), we have

$$\sigma(t+1) = \left[\frac{\lambda A(\lambda)}{t+1} \left[\frac{t\sigma(t)^{\lambda}}{\lambda A(\lambda)} + |x_{t+1} - \mu|^{\lambda} \right] \right]^{1/\lambda} \tag{A.8}$$

Therefore, we obtain:

$$\sigma(t+1) = \left[(1-\phi)\sigma(t)^{\lambda} + \phi \lambda A(\lambda) |x_{t+1} - \mu|^{\lambda} \right]^{1/\lambda} \tag{A.9}$$

where: $\phi = \frac{1}{1+t}$ is named the learning factor.

A.3 The mean of centered absolute value (MAV):

The mean of centered absolute value of the MoGG distribution (MAV) can be obtained as follows

$$\begin{aligned}
\mathbb{E}_{t+1}\big[\mid X \mid\big] &= \frac{1}{(t+1)\omega_{t+1}}\sum_{i=1}^{t+1}\mid x_i-\mu\mid \\
&= \frac{1}{(t+1)\omega_{t+1}}\Big(\sum_{i=1}^{t}\mid x_i-\mu\mid+\mid x_{t+1}-\mu\mid\Big) \\
&= \frac{t\omega_t}{(t+1)\omega_{t+1}}\mathbb{E}_t\big[\mid X \mid\big]+\frac{\mid x_{t+1}-\mu\mid}{(t+1)\omega_{t+1}} \\
&= \frac{(1-\phi)(\omega_{t+1}-\phi)}{(1-\phi)\omega_{t+1}}\mathbb{E}_t\big[\mid X \mid\big]+\frac{\phi\mid x_{t+1}-\mu\mid}{\omega_{t+1}} \\
&= \Big(1-\frac{\phi}{\omega_{t+1}}\Big)\mathbb{E}_t\big[\mid X \mid\big]+\frac{\phi}{\omega_{t+1}}\mid x_{t+1}-\mu\mid \\
&= \big(1-\rho\big)\mathbb{E}_t\big[\mid X \mid\big]+\rho\mid x_{t+1}-\mu\mid \qquad \text{(A.10)}
\end{aligned}$$

where $\rho=\frac{\phi}{\omega_{t+1}}$ and $\phi=\frac{1}{1+t}$ are the learning parameter and the learning factor respectively.

B

Evaluation metrics for all the CDnet dataset sequences

The following table summarizes the evaluation performance metrics obtained by application of the proposed method for all the Change Detection (CDnet 2014) dataset [142] categories and sequences.

We note that the methodologie in [142] is used to evaluate the BS results. The evaluation metrics in the head of Table B.1 are explained as: Re: Recall, Sp: Specificity ,FPR:Specificity, FNR:False Negative Rate, PWC: Percentage of Wrong Classifications, Pr: Precision, and F: F-measure.

Category	Video	Re	Sp	FPR	FNR	PWC	Pr	F
badWeather	blizzard	0.7177	0.9986	0.0014	0.0033	0.0047	0.8584	0.7818
	skating	0.8607	0.9982	0.0018	0.0073	0.0086	0.9620	0.9085
	snowFall	0.4518	0.9995	0.0005	0.0044	0.0048	0.8771	0.5964
	wetSnow	0.8013	0.9986	0.0014	0.0026	0.0039	0.8814	0.8394
baseline	PETS2006	0.9905	0.9965	0.0035	0.0001	0.0036	0.7888	0.8782
	highway	0.9842	0.9944	0.0056	0.0010	0.0063	0.9165	0.9491
	office	0.9608	0.9833	0.0167	0.0029	0.0182	0.8105	0.8792
	pedestrians	0.8320	0.9993	0.0007	0.0017	0.0023	0.9242	0.8757
cam.Jitter	badminton	0.8721	0.9979	0.0021	0.0045	0.0064	0.9368	0.9033
	boulevard	0.7610	0.9904	0.0096	0.0118	0.0204	0.7955	0.7779
	sidewalk	0.9524	0.9918	0.0082	0.0013	0.0092	0.7577	0.8440

	traffic	0.8962	0.9809	0.0191	0.0069	0.0244	0.7571	0.8208
dyn.Background	boats	0.9786	0.9996	0.0004	0.0001	0.0005	0.9386	0.9582
	canoe	0.9583	0.9988	0.0012	0.0015	0.0027	0.9657	0.9620
	fall	0.9608	0.9961	0.0039	0.0007	0.0045	0.8172	0.8832
	fountain01	0.7825	0.9993	0.0007	0.0002	0.0008	0.4940	0.6056
	fountain02	0.9020	0.9999	0.0001	0.0002	0.0003	0.9463	0.9236
	overpass	0.9524	0.9983	0.0017	0.0006	0.0023	0.8833	0.9166
int.Obj.Motion	abandonedBox	0.5483	0.9772	0.0228	0.0228	0.0434	0.5484	0.5484
	parking	0.0665	0.9998	0.0002	0.0782	0.0724	0.9593	0.1243
	sofa	0.7108	0.9890	0.0110	0.0132	0.0231	0.7470	0.7284
	streetLight	0.3329	0.9992	0.0008	0.0340	0.0332	0.9527	0.4933
	tramstop	0.4121	0.5246	0.4754	0.1286	0.4956	0.1594	0.2299
	winterDriveway	0.7757	0.9567	0.0433	0.0017	0.0446	0.1191	0.2065
lo.Fr.rate	port_0_17fps	0.6578	0.9993	0.0007	0.0001	0.0008	0.2293	0.3400
	tramCrossroad_1fps	0.9560	0.9918	0.0082	0.0013	0.0092	0.7680	0.8518
	tunnelExit_0_35fps	0.3334	0.9867	0.0133	0.0188	0.0312	0.4145	0.3696
	turnpike_0_5fps	0.6113	0.9990	0.0010	0.0312	0.0299	0.9791	0.7527
nightVideos	bridgeEntry	0.3835	0.9684	0.0316	0.0089	0.0399	0.1485	0.2141
	busyBoulvard	0.3289	0.9907	0.0093	0.0246	0.0326	0.5645	0.4157
	fluidHighway	0.5057	0.9741	0.0259	0.0071	0.0325	0.2188	0.3054
	streetCornerAtNight	0.6887	0.9943	0.0057	0.0015	0.0072	0.3775	0.4877
	tramStation	0.7353	0.9897	0.0103	0.0075	0.0173	0.6683	0.7002
	winterStreet	0.6583	0.9703	0.0297	0.0104	0.0389	0.4036	0.5004
PTZ	continuousPan	0.4199	0.9986	0.0014	0.0037	0.0050	0.6560	0.5120
	intermittentPan	0.8221	0.9985	0.0015	0.0025	0.0040	0.8879	0.8537
	twoPositionPTZCam	0.8701	0.9942	0.0058	0.0020	0.0077	0.6972	0.7741
	zoomInZoomOut	0.9979	0.4324	0.5676	0.0000	0.5664	0.0037	0.0074
shadow	backdoor	0.9712	0.9992	0.0008	0.0006	0.0013	0.9633	0.9672
	bungalows	0.9954	0.9783	0.0217	0.0003	0.0207	0.7455	0.8525
	busStation	0.9351	0.9927	0.0073	0.0025	0.0094	0.8313	0.8802
	copyMachine	0.9369	0.9903	0.0097	0.0047	0.0134	0.8782	0.9066
	cubicle	0.9292	0.9967	0.0033	0.0014	0.0046	0.8500	0.8878
	peopleInShade	0.9983	0.9874	0.0126	0.0001	0.0120	0.8258	0.9039
thermal	corridor	0.9154	0.9974	0.0026	0.0029	0.0053	0.9244	0.9198
	diningRoom	0.9207	0.9894	0.0106	0.0075	0.0165	0.8906	0.9054
	lakeSide	0.1801	0.9996	0.0004	0.0160	0.0161	0.9041	0.3003
	library	0.9424	0.9853	0.0147	0.0138	0.0229	0.9388	0.9406
	park	0.8821	0.9931	0.0069	0.0024	0.0091	0.7274	0.7973

TABLE B.1: (continued)

turbulence	turbulence0	0.8486	0.9923	0.0077	0.0003	0.0079	0.1728	0.2871
	turbulence1	0.6316	0.9991	0.0009	0.0014	0.0023	0.7379	0.6806
	turbulence2	0.9478	1.0000	0.0000	0.0000	0.0000	0.9937	0.9702
	turbulence3	0.7517	0.9994	0.0006	0.0041	0.0046	0.9502	0.8393

TABLE B.1: Quantitative results obtained by application of the proposed algorithm for all the CDnet videos.

References

[1] M.S. Allili, N. Bouguila and D. Ziou. Finite Generalized Gaussian Mixture Modeling and Applications to Image and Video Foreground Segmentation. *CRV*:183-190, 2007.

[2] M.S. Allili and D. Ziou. Globally Adaptive Region Information for Color-Texture Image Segmentation. *Pattern Recognition Letters*, 28(15):1946-1956, 2007.

[3] M.S. Allili, D. Ziou and N. Bouguila. Finite General Gaussian Mixture Modelling and Application to Image and Video Foreground Segmentation. *J. of Electronic Imaging*, 17(1):001-013, 2008.

[4] A. Amato, M. Mozerov, F. Roca and J. Gonzalez. Robust Real-Time Background Subtraction Based on Local Neighborhood Patterns. *EURASIP J. on Advances in Signal Processing*, Article ID 901205, 2010.

[5] A. Amato, I. Huerta, M. Mozerov, F. Roca, and J. Gonzalez. Moving cast shadows detection methods for video surveillance applications. *Augmented Vision and Reality*, 6(2014):23-47, 2014.

[6] M. Baccar, L. A. Gee and M.A. Abidi. Reliable Location and Regression Estimates with Application to Range Image Segmentation. *J. of Mathematical Imaging and Vision*, 11(3):195-205, 1999.

[7] D. Baltieri, R. Cucchiara, and R. Vezzani. Fast background initialization with recursive hadamard transform. *International Conference on Advanced Video and Signal Based Surveillance*, 2010.

[8] O. Barnich and M.V. Droogenbroeck. Vibe: A Powerful Random Technique to Estimate the Background in Video Sequences. *IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, 945-948, 2009

[9] Y. Bazi, L. Bruzzone and F. Melgani. Image thresholding based on the EM algorithm and the generalized Gaussian distribution. *Pattern Recognition*, 40(2):619-634, 2008.

[10] BEHAVE dataset: http://homepages.inf.ed.ac.uk/rbf/BEHAVE/

[11] B. Bhanu and J. Peng. Adaptive Integrated Image Segmentation and Object Recognition. *IEEE Trans. on Systems, Man, and Sybernetics-PART C*, 30(4):427-441, 2000.

[12] G.-A. Bilodeau, J.-P. Jodoin, and N. Saunier. Change detection in feature space using local binary similarity patterns. *International Conference on Computer and Robot Vision (CRV)*, 106112, 2013.

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

[14] A. Boulmerka, M.S. Allili. Thresholding-based segmentation revisited using mixtures of generalized Gaussian distributions. *Int'l Conf. on Pattern Recognition*. 2894-2897, 2012.

[15] A. Boulmerka, M.S. Allili, S. Ait-Aoudia. A generalized multiclass histogram thresholding approach based on mixture modelling. *Pattern Recognition*. 47(3):1330-1348, 2014.

[16] T. Bouwmans, F. El Baf and B. Vachon. *Statistical Background Modeling for Foreground Detection: A Survey*. World Scientific, 2010.

[17] T. Bouwmans. Traditional and Recent Approaches in Background Modeling for Foreground Detection: An Overview. *Computer Science Review*, 11(12):31-66, 2014.

[18] T. Bouwmans, F. Porikli, B. Höferlin and A. Vacavant. *Background Modeling and Foreground Detection for Video Surveillance*. CRC Press, 2015.

[19] A. C. Bovik. *The Handbook of Image and Video Processing, Second Edition*. Academic Press, 2005.

[20] G-E.P. Box and G.C. Tiao. *Bayesian Inference in Statistical Analysis*. Wiley Classics Library, 1992.

[21] A. D. Brink and N. E. Pendock. Minimum cross entropy threshold selection. *Pattern Recognition*, 29(1):179-188, 1996.

[22] S. Brutzer, B. Höferlin and G. Heidemann. Evaluation of Background Subtraction Techniques for Video Surveillance. *IEEE Conf. on Computer Vision and Pattern Recognition*, 1937-1944, 2011.

[23] D. Butler, V. Bove, and S. Shridharan. Real time adaptive foreground/background segmentation. *EURASIP Journal on Applied Signal Processing*, 14:2292-2304, 2005.

[24] CAVIAR dataset: http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

[25] C.-I Chang, Y. Du, J. Wang, S.-M. Guo and P.D. Thouin. Survey and comparative analysis of entropy and relative entropy thresholding techniques. *IEE Proc.-Vis. Image Signal Process*, 153(6):837-850, 2006.

[26] P.-L. St-Charles, G.-A. Bilodeau and R. Bergevin. Flexible Background Subtraction With Self-Balanced Local Sensitivity. *IEEE Workshop on Change Detection*,414-419, 2014.

[27] P.-L. St-Charles, G.-A. Bilodeau and R. Bergevin. *SuBSENSE: A Universal Change Detection Method With Local Adaptive Sensitivity. IEEE Trans. on Image Processing*, 24(1):359-373, 2015.

[28] Y.-L. Chen, B.-F. Wu, H.-Y. Huang and C.-J. Fan. A Real-Time Vision System for Nighttime Vehicle Detection and Traffic Surveillance, *IEEE Transactions on Industrial Electronics*, 58(5):2030-2044, 2011.

[29] Z. Chen and T. Ellis. A self-adaptive Gaussian mixture model. *Computer Vision and Image Understanding*, 122(2014):35-46, 2014.

[30] R. Cucchiara, C. Grana, M. Piccardi and A. Prati. Detecting Moving Objects, Ghosts, and Shadows in Video Streams. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1337-1342, 2003.

[31] A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society.* Series B (Methodological) 39(1):1-38, 1977.

[32] B. Dey and M. K. Kundu. Robust Background Subtraction for Network Surveillance in H.264 Streaming Video. *IEEE Trans. on Circuits and Systems for Video Technology*, 23(10):1695-1703, 2013.

[33] A. Elgammal, D. Harwood and L. Davis. Non-parametric Model for Background Subtraction. *European Conf. on Computer Vision*, 751-767, 2000.

[34] T. Elguebaly and N. Bouguila. Background Subtraction Using Finite Mixtures of Asymmetric Gaussian Distributions and Shadow Detection. *Machine Vision Application*, 25(5): 1145-1162, 2014.

[35] J. Fan. Notes on Poisson distribution-based minimum error thresholding. *Pattern Recognition Letters*, 19(5-6), 425431, 1998.

[36] S.-K.S. Fan, Y. Lin and C.-C. Wu. Image thresholding using a novel estimation method in generalised Gaussian distribution mixture modelling. *Neurocomputing*, 72(1-3):500-512, 2008.

[37] S.-K.S. Fan and Y. Lin. A fast estimation method for the generalized Gaussian mixture distribution on complex images. *Computer Vision and Image Understanding*, 113(7):839-853, 2009.

[38] M. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.* 24(3):381-396, 2002.

[39] W. T. Freeman, K. Tanaka, J. Ohta and K. Kyuma. Computer Vision for Computer Games. *Int'l Conf. on Automatic Face and Gesture Recognition*, 100-105, 1996.

[40] N. Friedman and S. Russell. Image Segmentation in Video Sequences: a Probabilistic Approach. *Conf. Uncertainty in Artificial Intelligence*, 175-181, 1997.

[41] K. S. Fu, and J. K. Mui. A survey on image segmentation. *Pattern Recognition*, 13(1):3-16, 1981.

[42] T. Gao, Z. Liu, W. Gao, and J. Zhang. A robust technique for background subtraction in traffic video. *International Conference on Neural Information Processing*, 736-744, 2008.

[43] C.A. Glasbey. An Analysis of Histogram-Based Thresholding Algorithms. *CVGIP: Graphical Models and Image Processing*, 55(6):532-537, 1993.

[44] N. Goyette, P. Jodoin, F. Porikli, J. Konrad and P. Ishwar. Changedetection.net: A New Change Detection Benchmark Dataset. *IEEE Conf. on Computer Vission and Pattern Recognition Workshop*, 1-8, 2012.

[45] N. Greggio, A. Bernardino, C. Laschi, P. Dario, J. Santos-Victor, Self-adaptive Gaussian mixture models for real-time video segmentation and background subtraction, *Int. Conf. on Intelligent Systems Design and Applications*, 983-989, 2010.

[46] M. De Gregorio and M. Giordano. Change Detection with Weightless Neural Networks. *IEEE Workshop on Change Detection*, 409-413, 2014.

[47] C. Guyon, T. Bouwmans, and E. Zahzah. Foreground Detection Via Robust Low Rank Matrix Decomposition including Spatio-temporal Constraint. *Computer Vision - ACCV Workshops*, (1)2012:315-320, 2012.

[48] B. Han and L. S. Davis. Density-Based Multi-Feature Background Subtraction with Support Vector Machine. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(5):1017-1023, 2012.

[49] M. Haque, M. Murshed, and M. Paul, Improved Gaussian mixtures for robust object detection by adaptive multi-background generation. *Int. Conf. on Pattern Recognition*, 1-4, 2008.

[50] R. M. Haralick and L. G. Shapiro. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 29(1), 100-132, 1985.

[51] M. Heikkilä and M. Pietikaïnen. A Texture-Based Method for Modeling the Background and Detecting Moving Objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, (28)4: 657-662, 2006.

[52] M. Hofmann, P. Tiefenbacher and G. Rigoll. Background Segmentation with Feedback: The Pixel-Based Adaptive Segmenter. *IEEE Conf. on Computer Vision and Pattern Recognition Workshop*, 38-43, 2012.

[53] T. Horprasert, D. Harwood and L.S. Davis. A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection. *IEEE International Conference on Computer Vision*, 1-8, 1999.

[54] J.-W.Hsieh, W.-F. Hu, C.-J. Chang, and Y.-S. Chen. Shadow elimination for effective moving object detection by Gaussian shadow modeling. *Image and Vision Computing*, (21):505-516, 2003.

[55] J-B. Huang, C.-S. Chen. Moving Cast Shadow Detection Using Physics-Based Features. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2310-2317, 2009.

[56] P.J. Huber and E.M. Ronchetti. *Robust Statistics*, Wiley Series in Probability and Statistics, 2nd Edition, 2009.

[57] i-LIDS Dateset: http://www.gov.uk/imagery-library-for-intelligent-detection-systems/

[58] M. Izadi, P. Saeedi, Robust region-based background subtraction and shadow removing using color and gradient information, *Int. Conf. on Pattern Recognition*, 15, 2008.

[59] F. Jiulun and X. Winxin. Minimum Error Thresholding: A Note. *Pattern Recognition Letters*, 18(8):705-709, 1997.

[60] P. Jodoin, M. Mignotte and J. Konrad, Statistical Background Subtraction Using Spatial Cues. *IEEE Trans. on Circuits and Systems for Video Technology*, 17(12):1758-1763, 2007.

[61] P. Kaewtrakulpong and R. Bowden. An Improved Adaptive Background Mixture Model for Realtime Tracking with Shadow Detection. *European Workshop on Advanced Video Based Surveillance Systems*, 1-5, 2001.

[62] J. Kato, T. Watanabe, S. Joga, J. Rittscher, and A. Blake. An HMM-based Segmentation Method for Traffic Monitoring Movies. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(9):1291-1296, 2002.

[63] K. Kim, T. Chalidabhongse, D. Harwood and L. Davis. Background Modeling and Subtraction by Codebook Construction. *IEEE Int'l Conf. on Image Processing*, 3061-3064, 2004.

[64] K. Kim, T. Chalidabhongse, D. Harwood and L. Davis. Real-Time Foreground-Background Segmentation Using Codebook Model. *Real-Time Imaging*, 11(3):172-185, 2005.

[65] J. Kittler and J. Illingworth. On threshold selection using clustering criteria. *IEEE Transactions on Systems, Man, and Cybernetics*, 15 (5): 652-655, 1985.

[66] J. Kittler and J. Illingworth. Minimum Error Thresholding. *Pattern Recognition*, 19(1):41-47, 1986.

[67] S. Kullback. *Information theory and statistics*, Dover, 1968.

[68] T. Kurita, N. Otsu and N. Abdelmalek. Maximum likelihood thresholding based on population mixture models. *Pattern Recognition*, 25(10):1231-1240, 1992.

[69] R.K.-S. Kwan, A.C. Evans, G.B. Pike. MRI simulation-based evaluation of image-processing and classification methods. *IEEE Transactions on Medical Imaging*, 18(11):1085-97, 1999.

[70] D.-S. Lee. Effective Gaussian Mixture Learning for Video Background Subtraction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(5):827-832, 2005.

[71] A. Leone, C. Distante. Shadow detection for moving objects based on texture analysis, *Pattern Recognition*, 40(4):1222-1233, 2007.

[72] L. Li, W. Huang, I. Gu and Q. Tian. Foreground Object Detection From Videos Containing Complex Background. *ACM Int'l Conf. on Multimedia*, 2-10, 2003.

[73] L. Li, W. Huang, I. Yu-Hua Gu and Q. Tian. Statistical Modeling of Complex Backgrounds for Foreground Object Detection. *IEEE Trans. Image Processing*, 13(11):1459-1472, 2004.

[74] P.-S. Liao, T.-S. Chen and P.-C. Chung. Multi-level thresholding for image segmentation through a fast statistical recursive algorithm. *Journal Of Information Science And Engineering*, 17(5):713-727, 2001.

[75] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S.Z. Li. Modeling pixel process with scale invariant local patterns for background subtraction in complex scene. *IEEE CVPR*, 1301-1306, 2010.

[76] H. Lin, T. Liu, and J. Chuang. A probabilistic SVM approach for background scene initialization. *International Conference on Image Processing*, 3:893-896, 2002.

[77] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. *ECCV*: 740-755, 2014.

[78] David Lowe. Web site of industrial vision applications and products: http://www.cs.ubc.ca/spider/lowe/vision.html

[79] L. Maddalena and A. Petrosino. A Self-Organizing Approach to Background Subtraction for Visual Surveillance Applications. *IEEE Trans. on Image Processing*, 17(7):1168-1177, 2008.

[80] V. Mahadevan, W. Li, V. Bhalodia and N. Vasconcelos. Anomaly Detection in Crowded Scenes. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1975-1981, 2010.

[81] S. Marron and M. P. Wand. Exact Mean Integrated Squared Error. *The Annals of Statistics*, 20(2):712-736, 1992.

[82] N. Martel-Brisson and A. Zaccarin. Learning and removing cast shadows through a multi-distribution approach. *IEEE PAMI*, 29(7): 1133-1146, 2007.

[83] D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues. *IEEE Trans. on Pattern Analysis and Machine Intelligence*: 26(5):530-549, 2004.

[84] N. McFarlane and C. Schofield. Segmentation and Tracking of Piglets in Images. *Machine Vision and Applications*, 8(3):187-193, 1995.

[85] J. M. McHugh, J. Konrad, V. Saligrama and P.-M. Jodoin. Foreground-Adaptive Background Subtraction. *IEEE Signal Processing Letters*, 16(5):390-393, 2009.

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

[87] S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld and H. Wechsler. Tracking Groups of People. *Computer Vision and Image Understanding*, 80(1):42-56, 2000.

[88] X.-L. Meng and D. Van Dyk. The EM Algorithman Old Folk-song Sung to a Fast New Tune. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59: 511567, 1997.

[89] F. Morii. A Note on Minimum Error Thresholding. *Pattern Recognition Letters*, 12(6): 349-351, 1991.

[90] G. Moser and B. Serpico. Generalized Minimum-Error Thresholding for Unsupervised Change Detection From SAR Amplitude Images. *IEEE Trans. on Geoscience and Remote Sensing*, 44(10):2972-2982, 2006.

[91] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[92] N. Nacereddine, S. Tabbone, D. Ziou and L. Hamami. Asymmetric Generalized Gaussian Mixture Models and EM Algorithm for Image Segmentation. *Int'l Conf. on Pattern Recognition*, 4557-4560, 2010.

[93] S. Nadimi and B. Bhanu. Physical Models for Moving Shadow and Object Detection in Video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(8): 1079-1087, 2004.

[94] H.Y.T. Ngan and G.K.H. Pang and N.H.C. Yung. Automated fabric defect detection-A review, *Image and Vision Computing*, 29(7):442-458,2011.

[95] A-T. Nghiem, F. Bremond, M. Thonnat, and V. Valentin. Etiseo, performance evaluation for video surveillance systems. *Advanced Video and Signal Based Surveillance*, 476-481, 2007.

[96] N. Oliver, B. Rosario and A. Pentland. A Bayesian Computer Vision System for Modeling Human Interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):831-843, 2000.

[97] N. Otsu. A Threshold Selection Method From Gray-Level Histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9(1):62-66, 1979.

[98] N. R. Pal and D. Bhandari. Image thresholding: some new techniques. *Signal Processing*, 33(2): 139-158, 1993.

[99] M. Piccardi. Background Subtraction Techniques: A Review. *IEEE Int'l Conf. on Systems, Man and Cybernetics*, 3099-3104, 2004.

[100] J. Pilet, C. Strecha, and P. Fua. Making background subtraction robust to sudden illumination changes, *ECCV*. 567580, 2008.

[101] F. Porikli. Integral Histogram: a Fast Way to Extract Histograms in Cartesian Spaces. *IEEE Conf. on Computer Vision and Pattern Recognition*,(1):829-836, 2005.

[102] F. Porikli and C. Wren. Change detection by frequency decomposition: Wave-back. *International Workshop on Image Analysis for Multimedia Interactive Services*, 2005.

[103] P.W. Power and J.A. Schoonees. Understanding background mixture models for foreground segmentation, *Image and Vision Computing, New Zealand*, 2002.

[104] A. Prati, I. Mikic, M. M. Trivedi and R. Cucchiara. Detecting Moving Shadows: Algorithms and Evaluation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(7):918-923, 2003.

[105] V. Reddy, C. Sanderson, A. Sanin, and B.C. Lovell. Adaptive patch-based background modeling for improved foreground object segmentation and tracking. *Int. Conf. on Advanced Video and Signal Based Surveillance*, 172-179, 2010.

[106] V. Reddy, C. Sanderson, and C. Lovell. A Low-Complexity Algorithm for Static Background Estimation from Cluttered Image Sequences in Surveillance Contexts. *J. Image and Video Processing*, Article ID 164956, 1-14, 2011.

[107] V. Reddy, C. Sanderson and B. C. Lovell. Improved Foreground Detection via Block-Based Classifier Cascade With Probabilistic Decision Integration. *IEEE Trans. on Circuits and Systems for Video Technology*, 23(1):83-93, 2013.

[108] R. Redner and W. Homer. Mixture densities, maximum likelihood and the EM algorithm. *SIAM review*, 26(2), 195-239, 1984.

[109] J. Rittscher, J. Kato, S. Joga, and A. Blake. A Probabilistic Background Model for Tracking. *European Conf. on Computer Vision*, 336-350, 2000.

[110] P. Rosin and E. Ioannidis. Evaluation of global image thresholding for change detection. *Pattern Recognition Letters*, 24(14):2345-2356, 2003.

[111] C. Rother, V. Kolmogorov, A. Blake. GrabCut: Interactive Foreground Extraction using Iterated Graph Cuts. *ACM Transactions on Graphics (SIGGRAPH'04)*, 2004.

[112] P.K. Sahoo, S. Soltani, A.K.C. Wong and Y.C. Chen. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing*, 41(2):233-260, 1988.

[113] P.K. Sahoo and G. Arora. Image thresholding using two-dimensional Tsallis-Havrda-Charvát entropy. *Pattern Recognition Letters*, 27(6):520-528, 2006.

[114] A. Sanin, C. Sanderson, and B. C. Lovell. Improved shadow removal for robust person tracking in surveillance scenarios. International Conferenceon Pattern Recognition, 141-144, 2010.

[115] A. Sanin, C. Sanderson and B. C. Lovell. Shadow Detection: A Survey and Comparative Evaluation of Recent Methods. *Pattern Recognition*, 45(4):1684-1695, 2012.

[116] A.E. Savakis. Adaptive Document Image Thresholding Using Foreground and Background Clustering. *IEEE Int'l Conference on Image Processing*, (3):785-789, 1998.

[117] M. Sedky, M. Moniri and C. C. Chibelushi. Spectral-360: A physical-Based Technique for Change Detection. *IEEE Workshop on Change Detection*, 405-408, 2014.

[118] M. Sezgin and B. Sankur. Survey Over Image Thresholding Techniques and Quantitative Performance Evaluation. *Journal of Electronic Imaging*, 13(1):146-165, 2004.

[119] M. Shah, J. Deng, and B. Woodford. Illumination Invariant Background Model Using Mixture of Gaussians and SURF Features. *ACCV Workshops* (1):308314, 2012.

[120] M. Shah, J. D. Deng and B. J. Woodford. Video Background Modeling: Recent Approaches, Issues and our Proposed Techniques. *Machine Vision Applications*, 25(5):1105-1119, 2014.

[121] K. Sharifi and A. Leon-Garcia. Estimation of shape parameter for generalized Gaussian distibution in subband decomposition of video. *IEEE Trans. on Circuits and Systems for Video Technology*, 5(1):52-56, 1995.

[122] M. Shoaib, R. Dragon and J. Ostermann. Shadow Detection for Moving Humans Using Gradient-Based Background Subtraction. *IEEE IUnt'l Conf. on Acoustics, Speech and Signal Processing*, 773-776, 2009.

[123] Y. Singh, P. Gupta, and V.S. Yadav. Implementation of a self-organising approach to background subtraction for visual surveillance approach. *Int. J. Comput. Sci. Network Secur.*, 10(3):136-143, 2010.

[124] M. Sonka, V. Hlavac and R. Boyle. *Image Processing, Anlayais and Machine Vision*, Thomson, 2008.

[125] C. Stauffer and W. E. L. Grimson, Learning Patterns of Activity Using Real-Time Tracking. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):747-757, 2000.

[126] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. Bouhman. Topology free hidden markov models: Application to background modeling. *IEEE International Conference on Computer Vision*, 294 - 301 , 2001.

[127] R. Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York Inc, 2010.

[128] X. Tang and C. von der Malsburg. Figure-Ground Separation by Cue Integration. *Neureal Computation*, 20(6):1452-1472, 2008.

[129] A. Tavakkoli, M. Nicolescu, and G. Bebis. Novelty detection approach for foreground region detection in videos with quasi-stationary backgrounds. *International Symposium on Visual Computing*, 40-49, 2006.

[130] F. Tiburzi, M. Escudero, J. Bescos, and J. Martinez. A ground truth for motion-based video-object segmentation. *IEEE Int. Conf. Image Processing*, 17-20, 2008.

[131] K. Toyama, J. Krumm, B. Brumiit, and B. Meyers. Wallflower: Principles and practice of background maintenance. *International Conference on Computer Vision*, 255261, 1999.

[132] P. Turaga, R. Chellappa, V. S. Subrahmanian and O. Udrea. Machine Recognition of Human Activities: A Survey. *IEEE Trans. on Circuits and Systems for Video Technology*, 18(11):1473-1488, 2008.

[133] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequievre. A benchmark dataset for outdoor foreground/background extraction. *ACCV Workshops*, 291-300, 2012.

[134] R. Vezzani and R. Cucchiara. Video surveillance online repository (visor): an integrated framework. *Multimedia Tools and Applications*, 50(2):359-380, 2010.

[135] P. Viola and M. J. Jones. Robust Real-Time Face Detection. *Int'l J. of Computer Vision*, 57(2):137-154, 2004.

[136] L. P. J. Vosters, C. Shan, and T. Gritti. Background subtraction under sudden illumination changes. *Int. Conf. on Advanced Video and Signal Based Surveillance*, 384-391, 2010.

[137] VSSN06 dataset: http://imagelab.ing.unimore.it/vssn06/

[138] B. Wang and P. Dudek. A Fast Self-tuning Background Subtraction Algorithm. *IEEE Workshop on Change Detection*, 395-398, 2014.

[139] J. Wang, G. Bebis, and R. Miller. Robust video-based surveillance by integrating target detection with tracking. *IEEE Workshop on Object Tracking and Classification Beyond the Visible Spectrum in conjunction with CVPR*, 137-144, 2006.

[140] R. Wang, F. Bunyak, G. Seetharaman and K. Palaniappan. Static and Moving Object Detection Using Flux Tensor with Split Gaussian Models. *IEEE Workshop on Change Detection*, 420-424, 2014.

[141] S. Wang, F.-L. Chung and F. Xionga. A novel image thresholding method based on Parzen window estimate. *Pattern Recognition*, 41(1):117-129, 2008.

[142] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth and P. Ishwar. CDnet 2014: An Expanded Change Detection Benchmark Dataset. *IEEE Workshop on Change Detection*, 387-394, 2014.

[143] B. White and M.Shah. Automatically Tuning Background Subtraction Parameters Using Particle Swarm Optimization. *IEEE Int'l Conf. on Multimedia and Expo*, 1826-1829, 2005.

[144] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-Time Tracking of the Human Body. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):780-785, 1997.

[145] C. Wren and F. Porikli. Waviz: Spectral similarity for object detection. *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2005.

[146] T. Xiang and S. Gong. Incremental and adaptive abnormal behaviour detection, *Computer Vision and Image Understanding*, 111(1):59-73,2008.

[147] J.-H. Xue, Y.-J. Zhang and X.G. Lin. Rayleigh-distribution based minimum error thresholding for SAR images. *Journal of Electronics (China)*, 16(4):336-342, 1999.

[148] J.-H. Xue and D.M. Titterington. Median-Based Image Thresholding. *Image and Vision Computing*, 29(9):631-637, 2011.

[149] J.-H. Xue and D.M. Titterington. Threshold selection from image histograms with skewed components based on maximum-likelihood estimation of skewnormal and log-concave distributions. *manuscript*, 2011.

[150] J.-H. Xue and Y.-J. Zhang. Ridler and Calvard's, Kittler and Illingworth's and Otsu's methods for image thresholding. *Pattern Recognition Letters*, 33(6):793-797, 2012.

[151] M. Yamazaki, G. Xu, and Y. Chen. Detection of moving objects by independent component analysis. *Asian Conference on Computer Vision*, 467-478, 2006.

[152] P.-Y. Yin and L.-H. Chen. A Fast Iterative Scheme For Multilevel Thresholding Methods. *Signal Processing*, 60(3):305-313, 1997.

[153] A. Yoneyama, C.-H. Yeh, and C.-C. Kuo. Robust vehicle and traffic information extraction for highway surveillance. *EURASIP Journal on Applied Signal Processing*. (2005):2305-2321.

[154] S. Yoshinaga, A. Shimada, H. Nagahara and R. Taniguchi. Object detection based on spatiotemporal background models. *Computer Vision and Image Understanding*, 122 (2014) 8491, 2014.

[155] D. Young and J. Ferryman. PETS metrics: Online performance evaluation service. *IEEE Int. Workshop on Performance Evaluation of Tracking Systems*, 317-324, 2005.

[156] S.X. Yu, R. Gross and J. Shi. Concurrent Object Recognition and Segmentation by Graph Partitioning. *Neural Information Processing Systems*, 1383-1390, 2002.

[157] Y. J. Zhang. *Advances in Image and Video Segmentation*, IRM Press, 2006.

[158] S.L. Zhao and H.J. Lee. A spatial-extended background model for moving blob extraction in indoor environments. *J. Inform. Sci. Eng.*, 25:1819-1837, 2009.

[159] Z. Zivkovic. Improved Adaptive Gaussian Mixture Model for Background Subtraction. *IEEE Conf. on Pattern Recognition*, 28-31, 2004.

[160] Z. Zivkovic and F. V. D. Heijden. Efficient Adaptive Density Estimation per Image Pixel for the Task of Background Subtraction. *Pattern Recognition Letters*, 27(7):773-780, 2006.