Enhancing the Potential of the Conventional Gaussian Mixture Model for Segmentation: from Images to Videos

Dibyendu Mukherjee
University of Windsor

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

Recommended Citation
https://scholar.uwindsor.ca/etd/5260

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

by

Dibyendu Mukherjee

APPROVED BY:

Dr. R. Laganiere, External Examiner
School of Electrical Engineering and Computer Science, University of Ottawa

Dr. D. Wu
School of Computer Science

Dr. H. Wu
Department of Electrical and Computer Engineering

Dr. E. A. Raheem
Department of Electrical and Computer Engineering

Dr. Q. M. J. Wu, Advisor
Department of Electrical and Computer Engineering

16 January, 2015
I. Co-Authorship Declaration

I hereby declare that this dissertation incorporates material that is result of joint research, as follows:

This dissertation also incorporates the outcome of a joint research undertaken in collaboration with Dr. Thanh Minh Nguyen under the supervision of professor Jonathan Wu. The collaboration is covered in Chapter 3, 4 and 5 of the dissertation. In all cases, the key ideas, primary contributions, experimental designs, data analysis and interpretation, were performed by the author, and the contributions of the co-authors were primarily through the provision of proof reading and reviewing the research papers regarding the technical content.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my dissertation, and have obtained written permission from each of the co-authors to include the above materials in my dissertation.

I certify that, with the above qualification, this dissertation, and the research to which it refers, is the product of my own work.

II. Declaration of Previous Publication

This dissertation includes 4 original papers that have been previously published/accepted in peer reviewed journals and conferences, as follows:

I certify that I have obtained a written permission from the copyright owners to
include the above published materials in my dissertation. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my dissertation does not infringe
upon anyone’s copyright nor violate any proprietary rights and that any ideas, tech-
niques, quotations, or any other material from the work of other people included in
my dissertation, published or otherwise, are fully acknowledged in accordance with
the standard referencing practices. Furthermore, to the extent that I have included
copyrighted material that surpasses the bounds of fair dealing within the meaning of
the Canada Copyright Act, I certify that I have obtained a written permission from
the copyright owners to include such materials in my dissertation.

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

Segmentation in images and videos has continuously played an important role in image processing, pattern recognition and machine vision. Despite having been studied for over three decades, the problem of segmentation remains challenging yet appealing due to its ill-posed nature. Maintaining spatial coherence, particularly at object boundaries, remains difficult for image segmentation. Extending to videos, maintaining spatial and temporal coherence, even partially, proves computationally burdensome for recent methods. Finally, connecting these two, foreground segmentation, also known as background suppression, suffers from noisy or dynamic backgrounds, slow foregrounds and illumination variations, to name a few.

This dissertation focuses more on probabilistic model based segmentation, primarily due to its applicability in images as well as videos, its past success and mainly because it can be enhanced by incorporating spatial and temporal cues. The first part of the dissertation focuses on enhancing conventional Gaussian Mixture Model (GMM) for image segmentation using Bilateral filter due to its power of spatial smoothing while preserving object boundaries. Quantitative and qualitative evaluations are done to show the improvements over a number of recent approaches.

The later part of the dissertation concentrates on enhancing GMM towards foreground segmentation as a connection between image and video segmentation. First, we propose an efficient way to include multiresolution features in GMM. This novel procedure implicitly incorporates spatial information to improve foreground segmentation by suppressing noisy backgrounds. The procedure is shown with Wavelets, and gradually extended to propose a generic framework to include other multiresolution decompositions. Second, we propose a more accurate foreground segmentation method by enhancing GMM with the use of Adaptive Support Weights (ASW) and Histogram of Gradients (HOG). Extensive analyses, quantitative and qualitative ex-
Experiments are presented to demonstrate their performances as comparable to other state-of-the-art methods.

The final part of the dissertation proposes the novel application of GMM towards spatio-temporal video segmentation connecting spatial segmentation for images and temporal segmentation to extract foreground. The proposed approach has a simple architecture and requires a low amount of memory for processing. The analysis section demonstrates the architectural efficiency over other methods while quantitative and qualitative experiments are carried out to show the competitive performance of the proposed method.
to my

mother and father

and my loving wife
Acknowledgements

I express my sincere gratitude to my advisor, Dr. Q.M. Jonathan Wu for giving me the opportunity to work under his supervision as well as for his guidance and support in my research. I like to thank Dr. Esam Abdel-Raheem for taking time out of his busy schedule to come over and guide me whenever I requested. I sincerely thank Dr. Dan Wu for guiding me when I was a novice in the field of research and helping me begin writing my first research document. I also thank Dr. Huapeng Wu for guiding me as a committee member and helping with his invaluable comments. I like to convey my sincere gratitude to Dr. Behnam Shahrrava, who provided through his courses, in-depth knowledge, which I would never gain by simply going through the text books. I heartily thank our department graduate secretary Ms. Andria Ballo, who has helped me in so many situations that I cannot possibly list in this limited space. Finally, I like to convey my earnest gratitude to the University of Windsor and Ontario Ministry of Training, Colleges and Universities for bestowing me with the prestigious Ontario Graduate Scholarship (OGS) for international students.

The person, who guided me to choose my field of research, continuously provided help in reviewing my works, and always stayed a perfect example of a researcher, inspiring me to go on, is Dr. Thanh Minh Nguyen. I like to convey my sincere thanks to him. The next person, who actually made it possible for me take the step towards pursuing a doctoral degree and continuously motivated me throughout this strenuous and long journey, is my wife Ashirbani Saha. As we are graduating together sharing an unique bond as a couple and as two colleagues helping each other in their struggles, I like to thank her for staying besides me.

I would not have come this far without the help from my fellow lab members, and would like to thank them for supporting me. A special thank goes to the department technician Mr. Frank Cicchello, who helped me whenever I requested, even during
his most busy hours.

I like to convey my regards, and sincere gratitude to my parents, my brother, and my in-laws. Under inexpressible circumstances and financial difficulties, my parents let me join the University and pursue my doctoral studies. I cannot thank them enough for their sacrifices and patience. I convey my gratitude to my mother and father-in laws for their continuous support and unfathomable trust. My parents and in-laws showed me how ideal parents take care of their children in the hardest of times.

Finally, I thank the researchers, whose works were the stepping stones and guidelines for me to proceed working in my research. I convey my sincerest regards to Google, Wikipedia and the researchers around the world, helping us to free our minds, grow our knowledge and come out of the darkness of ignorance, false beliefs and judgements. They have helped me believe that this gradual progress will lead to self-awareness and help us make a better world.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Declaration of Co-Authorship / Previous Publication</td>
<td>iii</td>
</tr>
<tr>
<td>Abstract</td>
<td>vi</td>
</tr>
<tr>
<td>Dedication</td>
<td>vii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>ix</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xvii</td>
</tr>
<tr>
<td>List of Acronyms</td>
<td>xix</td>
</tr>
<tr>
<td><strong>1 Introduction</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Image Segmentation</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Foreground Segmentation</td>
<td>9</td>
</tr>
<tr>
<td>1.3 Video Segmentation</td>
<td>18</td>
</tr>
<tr>
<td>1.4 Evaluation of Segmentation</td>
<td>21</td>
</tr>
<tr>
<td>1.4.1 For Image Segmentation</td>
<td>22</td>
</tr>
<tr>
<td>1.4.2 For Foreground Segmentation</td>
<td>25</td>
</tr>
<tr>
<td>1.4.3 For Video Segmentation</td>
<td>28</td>
</tr>
<tr>
<td>1.5 Motivation</td>
<td>28</td>
</tr>
<tr>
<td>1.6 Scope of this Work</td>
<td>29</td>
</tr>
<tr>
<td>1.7 Problem Statement</td>
<td>30</td>
</tr>
<tr>
<td>1.8 Objective</td>
<td>32</td>
</tr>
<tr>
<td>1.9 Organization of Thesis</td>
<td>32</td>
</tr>
<tr>
<td><strong>2 Literature Review</strong></td>
<td>33</td>
</tr>
<tr>
<td>2.1 Review on Image Segmentation</td>
<td>33</td>
</tr>
<tr>
<td>2.2 Review on Foreground Segmentation</td>
<td>37</td>
</tr>
<tr>
<td>2.3 Review on Video Segmentation</td>
<td>44</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Video Segmentation</td>
</tr>
<tr>
<td>7.1.4</td>
<td>General Summary</td>
</tr>
<tr>
<td>7.2</td>
<td>Scope for Future Work</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Image Segmentation</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Foreground Segmentation</td>
</tr>
<tr>
<td>7.2.3</td>
<td>Video Segmentation</td>
</tr>
<tr>
<td>References</td>
<td></td>
</tr>
<tr>
<td>Appendix A: IEEE Permission to Reprint</td>
<td>178</td>
</tr>
<tr>
<td>Vita Auctoris</td>
<td></td>
</tr>
</tbody>
</table>
# List of Tables

1.1 Datasets used for experiments .............................................. 27  
2.1 Notational simplifications for foreground and video segmentation . . . 56  
3.1 Performance of the proposed image segmentation method with varying  
level of noise and varying spatial variance ............................... 65  
3.2 Comparison of performance of the proposed image segmentation method  
with other methods for real-world colour images, in terms of PRI ........ 66  
3.3 Region benchmarking with the best image segmentation methods on  
BSDS500 ........................................................................... 67  
4.1 Quantitative evaluations of MRGMM on datasets: CMS, SZTAKI ........ 93  
4.2 Average metric values of MRGMM and other methods for the Car  
sequence ............................................................................ 97  
4.3 BMC: Average metric values of MRGMM and other methods over all  
datasets .............................................................................. 98  
4.4 CDW: Average ranking of MRGMM and other methods for each dataset 99  
4.5 Average ranking of MRGMM and other methods over all datasets for  
BMC and CDW ................................................................. 99  
4.6 Comparison of MRGMM and other methods in terms of Computational  
speed .................................................................................... 100  
4.7 Comparison of Codebook and KDE with their Wavelet based imple-  
mentations (I) ................................................................. 103  
4.8 Comparison of Codebook and KDE with their Wavelet based imple-  
mentations (II) ................................................................. 104  
5.1 Quantitative evaluations of ADMGMM on datasets (I): CMS, SZTAKI 124
5.2 Quantitative evaluations of ADMGMM on datasets (II): CMS, SZTAKI
5.3 BMC: Average metric values of ADMGMM and other methods over all datasets ........................................ 126
5.4 CDW: Average ranking of ADMGMM and other methods for each dataset ........................................ 126
5.5 Average ranking of ADMGMM and other methods over all datasets for BMC and CDW ................................. 129
5.6 Comparison of ADMGMM and other methods in terms of Computational speed ........................................ 130
6.1 Average metric values of GBH, SWA and the proposed video segmentation method for 8 datasets from Label Propagation database[1] ... 143
List of Figures

1.1 Perception of segmentation ........................................ 3
1.2 Noise Perturbation (NP) ........................................... 6
1.3 Improper Object Edge Segmentation (IOES) ..................... 7
1.4 Overlapping Intensities (OI) ...................................... 7
1.5 Contrast Variations (CV) and Less Prominent Object Boundaries (LPOB) .................................................. 8
1.6 Noisy Background (NB) ............................................ 12
1.7 Dynamic Background (DB) ......................................... 13
1.8 Slow Foreground (SF) ............................................... 15
1.9 Radial Motion (RM) ................................................ 15
1.10 Inadequate Background/Congested Foreground (IBCF) ........ 16
1.11 Illumination Variation (IV) ........................................ 17
1.12 Temporal Coherence ................................................ 20

3.1 Qualitative evaluations of the proposed image segmentation method on a synthetic image ........................................ 69
3.2 Qualitative evaluations with other low pass filters on a synthetic image ................................................ 70
3.3 Qualitative evaluations of the proposed image segmentation method on colour images ........................................ 71

4.1 The comparison of means acquired from K-means, GMM and WavGMM ................................................ 77
4.2 An objective validation of WavGMM performance .................. 79
4.3 A graphical representation of WavGMM and MRGMM ............ 85
4.4 Qualitative evaluations of MRGMM on CAVIAR video sequence 87
4.5 Qualitative evaluations of MRGMM on Postman video sequence .... 89
4.6 Qualitative evaluations of MRGMM on CMS video sequence ...... 91
4.7 Qualitative evaluations of MRGMM on SZTAKI video sequences ... 92
4.8 Qualitative evaluations of MRGMM on Car video sequence . . . . . . . 95

5.1 A graphical representation of ADMGMM . . . . . . . . . . . . . . . . 114
5.2 Comparison on mode selection of GMM and ADMGMM . . . . . . . 115
5.3 Qualitative evaluation of performance improvement using the back-
ground layers of ADMGMM . . . . . . . . . . . . . . . . . . . . . . 115
5.4 Qualitative evaluations of ADMGMM on CAVIAR video sequence . . 117
5.5 Qualitative evaluations of ADMGMM on CMS and ATON video se-
quences . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118
5.6 Qualitative evaluations of ADMGMM on ATON and SZTAKI video
sequences . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121

6.1 Automatic cluster formation, propagation and removal for the pro-
posed video segmentation method . . . . . . . . . . . . . . . . . . . . 135
6.2 A graphical representation of the proposed video segmentation algorithm . 138
6.3 Qualitative evaluations of the proposed video segmentation method on
Label Propagation datasets . . . . . . . . . . . . . . . . . . . . . . . 139

xviii
## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACC</td>
<td>Accuracy</td>
</tr>
<tr>
<td>ADMGMM</td>
<td>Advanced Distance Measure based Gaussian Mixture Model</td>
</tr>
<tr>
<td>ADMGMM_NI</td>
<td>Advanced Distance Measure based GMM with no Iterations</td>
</tr>
<tr>
<td>ASW</td>
<td>Adaptive Support Weights</td>
</tr>
<tr>
<td>BMC</td>
<td>Background Models Challenge</td>
</tr>
<tr>
<td>CBIR</td>
<td>Context-Based Image Retrieval</td>
</tr>
<tr>
<td>CBVR</td>
<td>Context-Based Video Retrieval</td>
</tr>
<tr>
<td>CDW</td>
<td>Change Detection Workshop</td>
</tr>
<tr>
<td>CMS</td>
<td>Carnegie Mellon Test Image Sequence</td>
</tr>
<tr>
<td>CONT</td>
<td>Contourlet Transform</td>
</tr>
<tr>
<td>ContGMM</td>
<td>Contourlet based Gaussian Mixture Model</td>
</tr>
<tr>
<td>CRF</td>
<td>Conditional Random Field</td>
</tr>
<tr>
<td>CRFGM</td>
<td>Conditional Random Field based Gaussian Mixture Model</td>
</tr>
<tr>
<td>CT</td>
<td>Curvelet Transform</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>Curvelet based Gaussian Mixture Model</td>
</tr>
<tr>
<td>CV</td>
<td>Contrast Variations</td>
</tr>
<tr>
<td>D</td>
<td>D-Score</td>
</tr>
<tr>
<td>DB</td>
<td>Dynamic Background</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>DR</td>
<td>Detection Rate</td>
</tr>
<tr>
<td>EGMM</td>
<td>Effective Gaussian Mixture Model</td>
</tr>
<tr>
<td>EM</td>
<td>Expectation-Maximization</td>
</tr>
<tr>
<td>F</td>
<td>F-Measure</td>
</tr>
<tr>
<td>FNR</td>
<td>False Negative Rate</td>
</tr>
<tr>
<td>FPR</td>
<td>False Positive Rate</td>
</tr>
<tr>
<td>FASOM</td>
<td>Fuzzy Adaptive Self-Organizing Maps</td>
</tr>
<tr>
<td>FCM</td>
<td>Fuzzy C-Means Clustering</td>
</tr>
<tr>
<td>GB</td>
<td>Efficient Graph-Based Image Segmentation</td>
</tr>
<tr>
<td>GBH</td>
<td>Efficient hierarchical graph-based video segmentation</td>
</tr>
<tr>
<td>GCE</td>
<td>Global Consistency Error</td>
</tr>
<tr>
<td>GMG</td>
<td>Godbehere, Matsukawa and Goldberg</td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian Mixture Model</td>
</tr>
<tr>
<td>HOG</td>
<td>Histogram of Gradients</td>
</tr>
<tr>
<td>HVS</td>
<td>Human Visual System</td>
</tr>
<tr>
<td>IBCF</td>
<td>Inadequate Background/Congested Foreground</td>
</tr>
<tr>
<td>IOES</td>
<td>Improper Object Edge Segmentation</td>
</tr>
<tr>
<td>IV</td>
<td>Illumination Variation</td>
</tr>
<tr>
<td>JC</td>
<td>Jaccard Coefficient</td>
</tr>
<tr>
<td>KDE</td>
<td>Kernel Density Estimate</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LPOB</td>
<td>Less Prominent Object Boundaries</td>
</tr>
<tr>
<td>LRE</td>
<td>Local Refinement Error</td>
</tr>
<tr>
<td>MAP</td>
<td>Maximum A Posteriori</td>
</tr>
<tr>
<td>MCC</td>
<td>Matthew’s Correlation Coefficient</td>
</tr>
<tr>
<td>MCR</td>
<td>Misclassification Ratio</td>
</tr>
<tr>
<td>MR</td>
<td>Multiresolution</td>
</tr>
<tr>
<td>MRF</td>
<td>Markov Random Field</td>
</tr>
<tr>
<td>MRGMM</td>
<td>Multiresolution based Gaussian Mixture Model</td>
</tr>
<tr>
<td>MSHVS</td>
<td>Mean-Shift based Hierarchical Video Segmentation</td>
</tr>
<tr>
<td>NB</td>
<td>Noisy Background</td>
</tr>
<tr>
<td>NP</td>
<td>Noise Perturbation</td>
</tr>
<tr>
<td>NYS</td>
<td>Spectral grouping using the Nyström method</td>
</tr>
<tr>
<td>OI</td>
<td>Overlapping Intensities</td>
</tr>
<tr>
<td>PR</td>
<td>Precision</td>
</tr>
<tr>
<td>PRI</td>
<td>Probabilistic Rand Index</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>PWC</td>
<td>Percentage of Wrong Classification</td>
</tr>
<tr>
<td>RC</td>
<td>Recall</td>
</tr>
<tr>
<td>RM</td>
<td>Radial Motion</td>
</tr>
<tr>
<td>ROI</td>
<td>Region Of Interest</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SAGMM</td>
<td>Self-Adaptive Gaussian Mixture Model</td>
</tr>
<tr>
<td>SD</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>SF</td>
<td>Slow Foreground</td>
</tr>
<tr>
<td>SMM</td>
<td>Students-t Mixture Model</td>
</tr>
<tr>
<td>SP</td>
<td>Specificity</td>
</tr>
<tr>
<td>SSIM</td>
<td>Structural Similarity based Image Quality Measure</td>
</tr>
<tr>
<td>SVFMM</td>
<td>Spatially Variant Finite Mixture Model</td>
</tr>
<tr>
<td>SWA</td>
<td>segmentation by weighted aggregation</td>
</tr>
<tr>
<td>T2FMRF_UM</td>
<td>Type-2 Fuzzy GMM and Markov Random Field based method with Uncertain Mean</td>
</tr>
<tr>
<td>T2FMRF_UV</td>
<td>Type-2 Fuzzy GMM and Markov Random Field based method with Uncertain Variance</td>
</tr>
<tr>
<td>VISIONSYS</td>
<td>A vision-based system for elderly patients monitoring</td>
</tr>
<tr>
<td>VoI</td>
<td>Variation of Information</td>
</tr>
<tr>
<td>WavGMM</td>
<td>Wavelet based Gaussian Mixture Model</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>Wavelet based Gaussian Mixture Model with Variable Clustering</td>
</tr>
<tr>
<td>WT</td>
<td>Wavelet Transform</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

It is said “a picture is worth a thousand words”. Truly, an entire story can be expressed by a mere picture, while a picture needs a lot of words to be properly expressed. Of course, the interpretation of such a picture must be carried out by something that understands the contents of the picture. This leads us to a question: how do we interpret the contents of an image? An image is a snapshot of some objects simultaneously existing together. These objects can be natural like trees, car, mountain, people and a lot more, or artificially created using digital technologies. However they are made, the objects follow certain similarities by which, they are interpreted by a human eye. An object has total or partial uniformity of colour and/or texture. Human eyes have excellent capabilities of grouping similarities and recognizing a group as an object. Thus, the interpretation is divided into two distinct parts: grouping, and recognition. The parts can be separate or combined depending on the object or prior knowledge of it. A human perceives an image using acquired knowledge, prior experience along with spatial cues and sometimes, depth cues. Thus, for the Human Visual System (HVS), the recognition part can take place together with the grouping. Unfortunately, the same cannot be accomplished when providing artificial intelligence to a computer system to interpret an image. Even the best processors with fast computing powers do not come close to a human baby’s intelligence. For a machine, this task is achieved, although partially, through machine vision, a relatively new subdivision of artificial intelligence, closely related with providing perception or simply eyes to a machine through the process of machine learning, to follow a human perception towards real world. This field of research is growing enormously with the
improvement of processing powers in computer, better hardware and vision tools and technologies.

As stated, interpretation of an image is divided into grouping and recognition. While recognition is the process of learning and interpreting the groups, the grouping is, in a broader and a more technical sense, termed as *segmentation*. In images, segmentation can range from simple grouping of regions based on similarity of colour, texture or other cues, to semantic grouping of objects. The cues, commonly called as *features*, can range from simple pixels, edges to contours or higher levels of structures extracted from an image through some algorithms. A semantic segmentation often takes place by merging subgroups to form an object that is visually perceivable through HVS. Thus, semantic segmentation requires a human’s knowledge or learning, and can be thought of as a bridge between grouping and recognition. As recognition part is out of the scope of this dissertation, we concentrate only on segmentation, or more importantly, only the first part of interpretation.

### 1.1 Image Segmentation

In simple terms, image segmentation is the process of dividing an image into spatially coherent non-overlapping regions based on some similarities. When we talk about image segmentation, we mostly do not consider the machine learning. Thus, image segmentation mostly relies on spatial features in an image. As already stated, spatial features can be of various types. Pixels or edges can be considered as elementary features. Better segmentation can be achieved by using more complex features. The segments obtained through segmentation can be viewed as larger pixels as they are made up of pixels of similar nature. Thus, a new term has been coined by Ren and Malik [2] for these segments: *super-pixels*. Thus, an image segmentation algorithm actually segments an image into several super-pixels.

With extensive studies on improving image segmentation, a question would simply
come up: What is a good segmentation? The quality of segmentation is generally judged through a human eye, and mostly depends on the application area. For example, consider two systems for segmentation shown in Fig. 1.1: first one is used to segment a scene into separate objects based on colour and texture uniformity and proximity, while the second one specifically segments traffic signs in a scene. Both systems working on a common traffic scene would produce different results. The fact is: both segmentations are correct, while they are useful for different applications. Thus, techniques and evaluation methodologies for segmentation have diversified over time based on applications. Some of the important application areas of image segmentation are presented below.

- **Image Analysis:** As already stated, interpretation of image contents are possible through segmentation, followed by recognition. Based on the type of analysis to perform, segmentation would also be different.

- **Object Recognition:** In general, recognition identifies an object to be part of a group or *class* of objects. Hence, the process is also called *classification.*
Specific objects can be segmented, and recognized or classified. An example of such has already been discussed in case of traffic signal segmentation, where the traffic signal is segmented into a single object based on some elementary features like colour and texture; afterwards, it is recognized to belong to the class of traffic signals. A system used to identify the presence and location of a specific object in an image, needs to be equipped with a segmentation algorithm specifically made to segment the object in question, and a recognition algorithm to recognize the segmented object. Of course, recognition cannot be done unless the class of the object is known.

- **Context-Based Image Retrieval (CBIR):** CBIR has lately become very popular with Google’s image based search. The concept of CBIR is that, based on a query image, images with similar content would be retrieved from a database. The measure of similarity is important and research is being pursued to find algorithms that can extract similarities from images in line with HVS. Segmentation has a major role to play here in order to segment proper content from images.

- **Medical Imaging:** In medical imaging, segmentation of 3D MRI images is one of the important application areas. Often a diagnosis for disease requires detection of abnormalities or damaged cells in such images. An accurate segmentation algorithm can lower the burden of a medical practitioner.

- **Video Analysis:** Video analysis is primarily based on image analysis, and hence, based on image segmentation. Although, more discussion on video analysis would be provided in Sections 1.2 and 1.3, it needs to be stated that video segmentation, although quite different from, and complex compared to image segmentation, implicitly incorporates image segmentation as a starting point.

As with numerous application areas, numerous challenges and consequently, numerous techniques and evaluation methodologies have also emerged for image segmen-
tation. While the evaluation methodologies are detailed in Section 1.4 and a literature review on different techniques is provided in Chapter 2, the challenges being more generic, are discussed below. As the problems under the category of accuracy are frequently referred to in subsequent sections and chapters, each of them is abbreviated with unique specialty-keys (also provided in the acronym section).

1. Accuracy: The main and foremost challenge of image segmentation remains in meeting accuracy levels set by the HVS. Applications with zero tolerance on errors, specifically in the field of medical imaging, require highly accurate segmentation. Accuracy fails for a number of reasons. Several important challenges related to accuracy are discussed next:

   • Noise Perturbation (NP): Perhaps the most common enemy for natural image segmentation is the noise present in the images. Noise is part of a natural image, and it is particularly dominant at the object boundaries due to the discontinuities. High amount of noise may prevent proper detection of edges. Thus, image segmentation algorithms are also greatly affected by noise. An example is shown in Fig. 1.2 where, a gray-scale image is perturbed by some noise and segmented using conventional Gaussian Mixture Model (GMM). As depicted, the segmentation cannot remove the noise at all, and the original gray values are also modified due to segmentation.

   • Improper Object Edge Segmentation (IOES): A prominent reason for IOES is the lack of proper discontinuities at object boundaries. In an image, an object normally follows a uniform or gradual change in intensity, colour and/or texture. However, at object boundaries, where two objects overlap, these attributes change rapidly, creating discontinuities. Thus, discontinuities are treated as official borders or end-markers for segmentation. However, image segmentation algorithms fail to properly segment
object boundaries if these discontinuities are noisy or unclear, resulting in overlapped or erroneous region contours, especially in presence of noise. For example, the image in Fig. 1.3 shows a gray scale square with four sub-squares inside. The image is perturbed with a small amount of noise, and segmented using Fuzzy C-Means Clustering (FCM) (mentioned in Section 2.1) which is a very robust method against noise. However, if looked carefully, the erroneous segmentation along the edges of the squares can be observed.

- Overlapping Intensities (OI): Intensity overlap occurs due to non-uniform object boundaries. Such events are prominent for images containing a mixture of fluids or viscous materials. An example is an MRI image (Fig. 1.4) where the white matter, gray matter and the cerebrospinal fluid have overlapping boundaries. Due to complex overlapping, the contours of each segment gets highly twisted and intertwined with each other. If compared to the ground-truth segmentation at the middle of Fig. 1.4, the segmentation using conventional GMM provides erroneous contours. Clearly, such overlapped regions are difficult to separate.

- Contrast Variations (CV): Contrast variations on object planes are very
Figure 1.3 – Improper Object Edge Segmentation (IOES): left - a gray scale square with four intensity values (255, 170, 85, 0); center - perturbation by noise; right - segmented image using FCM

Figure 1.4 – Overlapping Intensities (OI): left - original MRI image; center - ground-truth segmentation; right - segmented image using conventional GMM (original image source: the Internet Brain Segmentation Repository (IBSR))
common. However, when segmenting, such variations often result in discontinuities and produce broken segments. As shown in Fig. 1.5, the image of the crocodile is segmented by conventional GMM. The contrast variation on the skin of the animal as well as in the surroundings renders them very difficult to segment. For this example, GMM was not able to actually segment the image.

- Less Prominent Object Boundaries (LPOB): Related to the previous challenge, less prominent boundaries result in lower contrast between two adjacent objects. This prevents proper separation of the objects. This is also depicted in the previous example of CV in Fig. 1.5. Due to the low contrast between the tail of the crocodile and the surroundings, the tail is not prominent and it is partly misclassified with the surroundings in the segmentation.

Apart from the ones presented in the list above, there are numerous other problems associated with maintaining high accuracy in segmentation. Some of the challenges are related and may not be properly separable in a challenging scenario. However, discussion of all possible scenarios or every problem
is a cumbersome job and beyond the scope (presented in Section 1.6) of this dissertation.

2. Automatic Processing: A number of techniques require the user to designate the parts to segment, in an image, using seed points, scribbles or approximate (or exact) boundaries. This type of segmentation, although more accurate, always require human intervention. From the point of view of automatic processing, such intervention is not desirable.

3. Computation: The challenge from the application point of view remains in the real-time processing. A number of application areas including multimedia imaging, recognition tasks and CBIR require fast segmentation. However, in the effort to improve the accuracy, often the execution speed suffers.

4. Evaluation of Segmentation: A major challenge relates with one of the fundamental question - how to evaluate a segmentation? For a good segmentation, the semantic gap between low-level image features and high-level semantic grouping needs to be as low as possible. Of course, the quantification of semantic gap depends on HVS as well as the application. However, for a generic evaluation, several metrics and databases are used. This part would be detailed in Section 1.4.

As stated above, image segmentation has its applications in video analysis. The scope of this dissertation encompasses both image and video segmentation. Thus, the next discussions cover two major types of video analyses: foreground segmentation (Section 1.2), and spatio-temporal video segmentation (Section 1.3).

1.2 Foreground Segmentation

When videos are considered, a factor of time comes into play. In video frames, image contents change with time. Hence, along with spatial dependencies, a temporal
dependency is evident. If the camera location or scene is not changed, objects in consecutive video frames may have gradual motion. Thus, an object’s position in a frame depends on its position in the previous frame as well as its speed of change. The idea of foreground segmentation begins with registering this change of position, and detecting the object from its change of position in subsequent frames.

Foreground segmentation is a sub-genre of video segmentation, in which, the object in question is segmented by primarily using its motion cues. Thus, it is also commonly referred to as moving object segmentation. However, as the background has no motion, it needs to be ignored while the moving objects are considered to be part of the foreground. Hence, the name foreground segmentation. As a majority of the methodologies concentrate on suppressing background to segment the moving foreground objects, the name background suppression is also appropriate. Nonetheless, the name “foreground segmentation” would be used throughout to signify the similarities to image and video segmentation while bridging the gap between them.

Foreground segmentation is not pure spatio-temporal video segmentation. It does not segment the video frames independently or jointly into distinct non-overlapping groups. Instead, it is mostly concerned with only segmenting each frame into two classes: foreground and background. Some algorithms also have three classes: foreground, background and shadow. Spatio-temporal video segmentation has a very complex architecture to maintain spatial as well as temporal consistency. In earlier times, computer hardware could not handle such high complexity in processing as well as high memory requirements to store huge video data and segments. However, the two (or three) class problem of foreground segmentation can have a simpler architecture, as will be broadly discussed later in the dissertation, and can be easily incorporated in low-grade hardware. Thus, foreground segmentation has been the next research area to progress after image segmentation. Due to its use in segmenting moving objects, it had been originally developed for surveillance, traffic monitoring and tracking. Some of the application areas are discussed next:
- Traffic Monitoring: Monitoring the traffic using a standby camera is probably the most common use of foreground segmentation [4, 5]. It is very important in traffic intersections, highways and parking lots, where chances of accidents and traffic rule avoidance are common.

- Surveillance: Foreground segmentation has been used for surveillance since the advent of the field. It was one of the application areas that instigated the research in this genre [6].

- Video Annotation: Video annotation has found increasing interests due to the rise of social media and video archives. Foreground segmentation has found its use in this domain to help the users to automatically extract meaningful information from videos and tag or annotate them as required [7].

- Human-Computer Interaction: Although almost every application in the field of computer vision requires a human-computer interaction at some point of time, some applications are specifically designed for interaction purposes. Most common examples of such interactions lie in the domain of virtual reality, augmented reality, augmented virtually, electronic games and computer-based learning [8].

- Gesture Recognition: Human gesture recognition is a growing research area. Foreground segmentation is used at an early stage to extract the human or simply the gesture to process further [9].

- Action Recognition: Similar to gesture recognition, action recognition requires the person performing the action to be segmented in order to proceed towards the recognition phase [10, 11].

Similar to Section 1.1, the domain of foreground segmentation consists of a number of challenges. Due to the change of architecture of algorithms used, the challenges are also quite different from the ones faced in image segmentation. The challenges, with specialty-keys whenever required, are presented below:
Figure 1.6 – Noisy Background (NB): left - original video frame; right - noisy segmentation using simple background subtraction (original video frame source: the Change Detection Workshop (CDW) datasets [12])

1. Accuracy: Similar to image segmentation, accuracy remains the biggest and most important challenge for foreground segmentation. The main hurdles preventing a high accuracy are described next:

- Noisy Background (NB): Noisy background refers to unwanted motions in background. Such motions can originate from high amount of noise due to the low quality of camera, camera jitter (Fig. 1.6), transmission errors, errors in compression etc. If such motions are detected by the technique, the foreground detection would get affected. Thus, a robust foreground segmentation technique should be able to separate NB from required foreground motion. An example of incorrect segmentation is shown in Fig. 1.6 where, the camera jitter produces noises. As the moving pedestrians are properly detected, the stationary zebra-crossing as well as some other parts of background are also detected as foreground. Sometimes, post-processing of the segmented video frame using morphological filters like erosion and opening can get rid of NB.

- Dynamic Background (DB): A dynamic background usually has moving
objects as part of the background. It is quite similar to NB, but the fundamental difference is in the origin of the movements. For DB, the background itself contains moving objects, while in case of NB, the movement is generated due to noises in the processes of capturing, transmission, compression and so on. Examples of dynamic background are objects having periodic nature or continuous flow of motion, like: tree leaves and branches moving due to wind, wheels of running vehicles, waves in water or water fountain. Effects of dynamic backgrounds cannot be removed by simple morphological filters as the motions are considerably high. Consider a case of movements in tree leaves as shown in Fig. 1.7. The parked car and the stationary pedestrian are not detected while the moving car and pedestrians are detected. However, the tree leaves have detectable movements due to wind and represent DB. As the leaves cover a considerable part of the video frames, they cannot be easily removed by morphological filters without affecting the other foreground objects.

- Slow Foreground (SF): Slow foreground results from any foreground object moving very slowly or staying at a place for some time so that the
segmentation technique misinterprets the object as background and cannot detect it. Sometimes, an object begins movement after staying at a place for a long time. After leaving the place, the object is detected in its motion. However, the background covered by the object while it was stationary, is also detected as part of the foreground. It occurs because the segmentation technique has no knowledge of this uncovered background from before, or the knowledge has been forgotten as the slow foreground object stayed at the same place for a long time. This is known as ghost effect because a foreground is detected where no foreground object exists, or simply a ghost of foreground exists. An example shown in Fig. 1.8 would clarify the phenomenon in a better way. The first frame shows two persons standing in the middle of the scene (view captured from the ceiling). After a while, one of them moves to a different place. However, due to staying at the previous location for a long time, the person was considered as part of background. As the original background is suddenly uncovered, it is detected as a foreground representing a “ghost”. The segmentation algorithm needs to update its knowledge periodically, and cannot keep the same knowledge for a long time in order to get accustomed to changing environments. This short-term knowledge and slow foreground together give rise to the ghost effect.

• Radial Motion (RM): An object moving towards or away from the camera approximately parallel to the camera axis, always has a part of itself covering some part of background. If the object has a long duration of radial movement, this covered part may be mistaken for background and would not be detected. The mailman in Fig. 1.9 has such a movement that causes him to occupy certain part of the video frames for a long time. After a while, the mailman is not properly detected and is considered to be part of the background.
Figure 1.8 – Slow Foreground (SF): left - first video frame with the two persons stationary; center - another video frame showing one of them as moving; right - segmentation of center frame using conventional GMM showing the “ghost” at the position of the moving person in first frame (original video frames source: the CAVIAR datasets)

Figure 1.9 – Radial Motion (RM): left - a video frame showing a mailman walking towards a camera; center - a frame later in time showing the mailman nearer to the camera; right - segmentation of center frame using conventional GMM and after removing noise using morphological opening for clarity (original video frames taken from the Postman video sequence)
Inadequate Background/Congested Foreground (IBCF): Train stations or highways are very common examples of congested foregrounds where some part of the background can be rarely seen as it is always covered by some foreground objects. This kind of congested foreground as shown in Fig. 1.10, prevents proper modeling of the background and hence, affects subsequent segmentation. If the background is not properly known, it cannot be separated from the foreground.

Illumination Variation (IV): Illumination variation is a prevalent phenomenon in natural scenes due to the gradual or sudden changes of incoming light from sun or any available light source over time, and shadows cast by foreground or background objects. A scene captured from a camera in two different times of a day can be visualized very differently due to the effect of varying amount and direction of light as shown in Fig. 1.11. The same scene captured in the morning (left) and in the evening (right) has different location of shadows and large illumination changes causing IV.
This is an example of gradual change in illumination. An example of sudden changes in illumination is an indoor scene where, a light bulb is turned on in a dark room. A robust technique must be able to accommodate such changes.

2. Automatic Processing: Some background modeling based methods require human intervention to specify the number of background modes. Some methods require a lot of parameter tuning to fit a particular scene. Thus, automatic processing is also a challenge for foreground segmentation.

3. Computation: Most of the application areas of foreground segmentation require real-time processing of a large amount of data. Even if off-line processing is allowed, a large video surveillance data of several hours requires high amount of memory and computation power to be processed. Thus, fast computation and low memory consumption are two major challenges for foreground segmentation techniques.

With the discussion on image segmentation and foreground segmentation, we are more familiar with the spatial and temporal coherence. With this knowledge, we
move on to the discussion on pure spatio-temporal video segmentation in the next section.

1.3 Video Segmentation

Spatio-temporal video segmentation, or in a general sense of the term, video segmentation refers to grouping similar contents in each video frame, label them and propagate the same label throughout the video frames to uniquely represent the same group over time. Thus, the segments are not necessarily pixel groups. They consist of pixel groups, represented by unique labels, covering a number of frames. In connection to super-pixels, these groups having spatial and temporal similarities, are termed supervoxels. Here, voxel is the combination of volume and pixel, and refers to a pixel in a 3-dimensional (3-D) space. This 3-D space is constructed by the video frames over time. A voxel, unlike a pixel denoted by its row and column coordinates, is denoted by its row, column as well as temporal coordinate (or simply the video frame number) with respect to other voxels. Similarly, supervoxel is a 3-D structure containing a group of voxels. Currently, most of the popular video segmentation methods are evaluated in terms of the quality and quantity of supervoxels produced by the segmentation. Mostly, the supervoxels need to have spatio-temporal consistency while being reasonably large in size and small in quantity. These factors determine the quality of the video segmentation. The quality of the video segmentation is measured through supervoxels in order to make the segmentation consistent with the objects present in the video. Too many segments would ruin the purpose while too few segments would lead to wrong interpretation. However, this evaluation methodology is not applicable for segmentation methods that do not represent the segmentation in terms of supervoxels, but simply colour or spatial values of segments. Further discussion on the evaluations are provided in Section 1.4.3.

The goal of video segmentation is to extract the objects in a video, required for a
number of applications. Some of the applications are listed as follows:

1. Activity Recognition: Activity recognition is related to action recognition but is different from it due to its collective nature. An activity is made up of actions from one or more agents’ as well as the environmental conditions [18]. An example: while the movement of a car is an action, parking the car in a parking lot is an activity. Thus, activity recognition can be considered as a superset of action recognition.

2. Object Tracking: Object tracking has always been a key area where a video is spatio-temporally segmented and the objects are tracked in the segments [19].

3. Context-Based Video Retrieval (CBVR) and Browsing: With reasonable progress in CBIR, the next inevitable step is CBVR. Video segmentation has its major application in this domain to segment logically separable objects through spatio-temporal voxels followed by recognition [20].

4. Semantic Analysis: Semantic analysis is highly related to the fields of CBVR as the elements of a video need to be classified into known objects in order to search for similar objects [21].

5. Visual Enhancement: The goal of visual enhancement is to improve the appearance and/or quality of the video. Context-based visual enhancement methods use spatio-temporal video segmentation in an intermediate stage [22].

Video segmentation merges the fields of image segmentation and foreground segmentation and takes a further step. Thus, the challenges of image segmentation and foreground segmentation are partially applicable as well. However, apart from these challenges, video segmentation has a number of unique challenges [23, 24] as follows:
Temporal Coherence: In image segmentation, segmented regions must show a spatial coherence, i.e. regions should be consistent with object boundaries. A similar constraint is put on video segmentation so that the segmented regions remain spatially as well as temporally coherent. Loss of temporal coherence would lead to inconsistency of object boundaries in successive frames. If a segment is uniquely represented by a label, and the label is visualized with a colour in the segmented video, it is desired to have the same segment to be represented by the same label and hence, same colour. Inconsistent coherence would lead to assignment of different labels for same segment in different video frames, and hence, a flickering of colour in subsequent frames. Notice that, in this case, the segments in subsequent frames are also not part of the same supervoxel. Such a phenomenon is depicted in Fig. 1.12 where two subsequent video frames from a single video sequence “Atonement” [24] are shown with their
segmentation done by one of the state-of-the-art methods: SWA [25] (discussed in Section 2.3. The frames are not segmented together in order to show the effect of temporal incoherence. Hence, the frames do not have any supervoxel in common and suffer from high amount of flickering. Very few algorithms [24, 25, 26] have achieved temporal consistency over a long run of video.

2. Automatic Processing: A video segmentation algorithm must automatically act on predefined similarity criteria and produce visually distinct regions over time. However, in practice, this part is hard to achieve. Many algorithms require a human intervention to provide some seed points or approximate boundary for segmentation.

3. Scalability: Most often, to achieve coherence over a long run of frames (at least over 5 frames), a high amount of memory and processing power are required. Only a few methods in current literature have been able to provide acceptable solutions to this problem. However, all of these methods (broadly discussed in Chapter 2) require a high amount of computation time.

4. Computation: In connection to previous challenge, computation complexity is a major bottleneck for video segmentation. Off-line methods process all frames of a single video together, requiring a heavy computational power. Graph based methods keep a large graph for the segments, and call for large memory to store such graphs. Thus, reduction of computation is one of the main areas to work on.

1.4 Evaluation of Segmentation

A question was raised in Section 1.1: What is a good segmentation? Prevalently, the quality of segmentation depends on the application as already shown in Fig. 1.1. However, for general classification into “distinct regions” corresponding to real-world
objects, a framework can be made. A framework requires specific data to test on, and
databases are also made. As one can expect, the evaluation for image segmentation,
foreground segmentation and video segmentation are not same. Thus, this section
is subdivided into three subsections in order to discuss the evaluations and related
databases for each category.

1.4.1 For Image Segmentation

The basic and simplest evaluation measure for image segmentation is the Misclassification Ratio (MCR) [27]. MCR is defined as follows:

\[
MCR = \frac{\text{number of misclassified pixels}}{\text{total number of pixels}} \times 100
\]  

(1.1)

The value of MCR is in the range of \([0, 100]\), where lower values indicate better segmentation results. If the ground-truth segmentation is available, this measure can be used to find the performance of a segmentation technique. Generally, a ground-truth segmentation assigns each segment or region in an image, to unique labels. To compare with the ground-truth, a segmentation technique needs to compute the segments and assign each segment to one of these labels. Thus, each pixel bears a label after the segmentation. If a pixel has a different label than the one assigned to it in the ground-truth, the pixel is determined as misclassified. Thus, MCR evaluates the segmentation in terms of the fraction or percentage of misclassified pixels.

In reality, each person has a different judgement and perception for segmentation. Hence, a ground-truth prepared by a single person most often would not suffice for others. Hence, popular databases have multiple ground-truth segmentation maps prepared by several persons for each image, denoted by \(G = G_1, G_2, \ldots, G_M\) where, \(M\) is the number of ground-truths available. The segmentation map to be evaluated is termed as \(G_{eval}\). One of the most popular technique for evaluating \(G_{eval}\) with multiple ground-truths presented in \(G\), is the Probabilistic Rand Index (PRI) [28].
PRI is given as follows:
\[
PRI(G, G_{eval}) = \frac{2}{N(N-1)} \sum_i \sum_{j>i} [c_{ij}p_{ij} + (1 - c_{ij})(1 - p_{ij})],
\]  
(1.2)

Where, \( N \) is the number of data points (here, the number of image pixels) and \( p_{ij} \) is the ground truth probability that pixels \( i \) and \( j \) belong to the same segment. Value of \( c_{ij} \) is equal to 1 if pixels \( i \) and \( j \) belong to the same segment in \( G_{eval} \), and equal to 0 otherwise. PRI takes a value in the range of \([0, 1]\). A score of 0 indicates absolutely bad segmentation with no similarity to ground-truth. That means, every pixel pair in the test segmentation map has opposite relationship to the corresponding pair in the ground-truth. Similarly, a score of 1 indicates that each pair of pixels in the test map has same relationship as the corresponding pair in the ground-truth. The advantage of using PRI is that the number of labels in \( G_{eval} \) need not be equal to the number of labels in any ground-truth map under \( G \).

Another popular measure is the Global Consistency Error (GCE) [29]. It is related to the consistency among segmentations. Consider one of the ground-truth maps \( G_m \) and the test map \( G_{eval} \) to be compared. For a given pixel \( p_i \), let the segments containing \( p_i \) in \( G_m \) and \( G_{eval} \) to be denoted by \( S(G_m, p_i) \) and \( S(G_{eval}, p_i) \), respectively. The Local Refinement Error (LRE) between \( G_m \) and \( G_{eval} \) for \( p_i \) is denoted as follows:
\[
LRE(G_m, G_{eval}; p_i) = \frac{|S(G_m, p_i)\setminus S(G_{eval}, p_i)|}{|S(G_m, p_i)|},
\]  
(1.3)

Where, \( |S_x \setminus S_y| \) denotes the set difference between sets (segments) \( S_x \) and \( S_y \). LRE is not symmetric and encodes a measure of refinement in one direction only. \( LRE(G_m, G_{eval}, p_i) \) is approximately 0 when \( G_m \) is a refinement of \( G_{eval} \) but not vice versa. GCE combines the LRE in both directions over all pixels and forces all local refinements to be in the same direction. It is defined as follows:
\[
GCE(G_m, G_{eval}) = \frac{1}{N} \min \left\{ \sum_i LRE(G_m, G_{eval}, p_i), \sum_i LRE(G_{eval}, G_m, p_i) \right\}.
\]  
(1.4)

GCE takes a value in the range of \([0, 1]\) with lower values indicating refinement and better segmentation and higher values representing inconsistent overlap of segments.
GCE is a measure tolerant to refinement of segmentation boundaries from one map to another map. However, it is only meaningful if the two segmentations have similar number of segments, which is not a general case.

Third measure to mention is the Variation of Information (VoI) [30]. It is based on the entropy of a segmentation $H(G_m)$ and the mutual information between two segmentations $I(G_m, G_{eval})$, and is represented as follows:

$$ \text{VoI}(G_m, G_{eval}) = [H(G_m) - I(G_m, G_{eval})] + [H(G_{eval}) - I(G_m, G_{eval})]. $$ (1.5)

The two terms represent the conditional entropies $H(G_m|G_{eval})$ and $H(G_{eval}|G_m)$. The first term measures the amount of information about $G_m$ that we lose, while the second measures the amount of information about $G_{eval}$ that we have to gain, when going from clustering $G_m$ to clustering $G_{eval}$. VoI does not require the number of segments to be equal for both maps. As it represents a distance between two segmentations, lower values represent similarity and hence, better segmentation. Out of the measures discussed, VoI is a true metric satisfying the metric axioms.

There are a number of other measures for evaluating image segmentation. However, for the scope of the work on image segmentation discussed in Chapter 3, MCR and PRI would suffice. Instead, GCE and VoI are used to evaluate video segmentation to provide more insight.

The second criterion to evaluate image segmentation is the database used. For image segmentation, one of the most popular benchmarking database is the Berkeley Segmentation Datasets [3]. The first version BSDS300 had 300 training and testing images with ground-truths made by human subjects [31]. The next version BSDS500 have 500 images including the first 300. Images from this database are used to evaluate the image segmentation technique proposed. Apart from this database, other artificial images are used to test tolerance against noise.
1.4.2 For Foreground Segmentation

Evaluation of foreground segmentation is more straight-forward as the target is to evaluate the extraction of the moving foreground. The ground-truth for each video frame is a binary image where, the regions representing foreground and background are designated by the values 1 and 0, respectively. There are a number of different measures that can be used for the quantitative performance evaluation including MCR. However, for video segmentation, the measures related to MCR have quite different terminology, as discussed by Chen and Ellis [32]. The authors have used five different metrics for comparison - Detection Rate (DR), False Positive Rate (FPR), Accuracy (ACC), Jaccard Coefficient (JC) and Matthew’s Correlation Coefficient (MCC). The definitions for them are provided below:

\[
\begin{align*}
\text{DR} &= \frac{TP}{TP + FN}; \\
\text{FPR} &= \frac{FP}{FP + TN}; \\
\text{ACC} &= \frac{TP + TN}{TP + FN + TN + FP}; \\
\text{JC} &= \frac{TP}{TP + FP + FN}; \\
\text{MCC} &= \frac{TP \times TN - FP \times FN}{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}.
\end{align*}
\]

where, TP, FP, TN and FN denote the number of true positives (foreground pixels correctly classified as foreground), false positives (background pixels wrongly classified as foreground), true negatives (background pixels correctly classified as background) and false negatives (foreground pixels wrongly classified as background) respectively. Out of the above five measures, ACC, JC and MCC are considered to be the best [32] and are used in the quantitative analysis. FPR is also used for average results to highlight the misclassification. For FPR, lower value represents better result while higher values represent better results for DR, ACC, JC and MCC.

Similar to image segmentation, there are other measures available for evaluation of foreground segmentation [12]. In literature, a number of databases for foreground segmentation have been proposed with varying level of challenges. To qualify the
effectiveness of a foreground segmentation method against such challenges, the creators of the databases have used specific sets of measures. Before discussion of the database-specific evaluation measures, a brief discussion on the databases is required.

There are a number of databases available for foreground segmentation. The most appropriate databases for the scope of this work are as follows: the CAVIAR datasets, the Carnegie Mellon Test Image Sequence (CMS) [33] and the SZTAKI and ATON surveillance benchmark set [15, 16, 17]. However, we have used the “Car” sequence from DynTex dynamic textures datasets [14] and the “Postman” sequence (not part of any database). The Background Models Challenge (BMC) [34] datasets, the Change Detection Workshop (CDW) [12] datasets and the Wallflower database [6] are three major databases in the domain of change detection and very useful for evaluating foreground segmentation. Out of these databases, CAVIAR datasets do not have any ground-truth maps and Wallflower datasets have ground-truth for a single frame in each video sequence. Also, the Car and Postman sequence do not contain ground-truths. However, as the Car sequence is very important in order to evaluate certain aspects of the techniques proposed in this work, we have manually created ground-truths for the frames 11-100 in order to quantify the performance on this dataset. The rest of the datasets have ground-truth and are used for quantitative analysis.

Each database is unique and has a number of challenging video sequences. As we have already specified the challenges in Section 1.2, the challenges are associated in Table 1.1 with the databases and datasets (that are not part of any database) using their specific specialty keys.

Among the databases, BMC and CDW provide their own evaluation benchmark with specific sets of evaluation measures. BMC use the following measures: Recall (RC), Precision (PR), F-Measure (F), Peak Signal-to-Noise Ratio (PSNR), D-Score (D) and Structural Similarity based Image Quality Measure (SSIM). CDW use the following measures: RC, Specificity (SP), FPR, False Negative Rate (FNR), Percentage of Wrong Classification (PWC), PR and F. Out of these metrics, low
<table>
<thead>
<tr>
<th>Database</th>
<th>Specialty key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAVIAR</td>
<td>NB; SF</td>
<td>Person standing for long time; moving slowly; exiting and entering scene</td>
</tr>
<tr>
<td>CMS</td>
<td>DB; NB; SF</td>
<td>Person walks across the scene; Car entering and exiting scene</td>
</tr>
<tr>
<td>SZTAKI &amp; ATON</td>
<td>DB; IV; NB; RM; SF</td>
<td>Cars moving on roads; person standing and moving; People moving inside and outside buildings</td>
</tr>
<tr>
<td>Car</td>
<td>IBCF; NB; NB; SF</td>
<td>Person getting out of van, coming towards camera and leaving towards car radially</td>
</tr>
<tr>
<td>BMC</td>
<td>DB; IV; NB; RM; SF</td>
<td>Cars moving on roads; People and animal moving; Scene changing over time; Dynamic backgrounds</td>
</tr>
<tr>
<td>CDW</td>
<td>DB; IBCF; IV; NB; RM; SF</td>
<td>Indoor and outdoor scenes; Scene changing</td>
</tr>
<tr>
<td>Wallflower</td>
<td>DB; IV; NB</td>
<td>People and objects standing and moving</td>
</tr>
</tbody>
</table>

Table 1.1 – Datasets used for experiments
values of D, FPR, FNR and PWC represent better results whereas, high values for others represent improvements. Each measure is unique and has a specific definition. Discussion of each one would make this discussion too cumbersome. Instead, interested readers are referred to the original papers for the databases to find the complete definitions for each measure.

1.4.3 For Video Segmentation

For supervoxel based methods, a supervoxel based analysis has been done before [23]. However, as the proposed method in Chapter 6 is not actually supervoxel based, a frame based evaluation is adapted. As each frame is segmented and segments are represented by unique intensity/colour values similar to image segmentation, evaluation measures applicable to image segmentation can be used. Hence, PRI, GCE and VoI are used to evaluate video segmentation for this work.

As there have been comparatively less amount of research in this domain, very few datasets have ground-truth segmentation available for each frame. The Label Propagation database from Chen [1] is mainly used for the evaluation. The datasets in this database have ground-truth segmentation maps for quantitative evaluation. Also, video sequences provided in [24] are used for qualitative comparison as these videos are very long and have a lot of variations.

1.5 Motivation

Segmentation represents one of the fundamental areas in computer vision and machine learning. As briefly discussed before, the fields of image segmentation, foreground segmentation and video segmentation have a broad range of application areas. Hence, these fields are well-explored. However, in terms of a human-eye, segmentation is still a highly ill-posed problem. Numerous works have been proposed, are being proposed and will be proposed to improve the quality and performance of segmentation. The
topic of segmentation chosen for this research was mainly due to its fundamental importance. It bridges the gap between image processing and high level computer vision or machine learning. Thus, a person with an image processing background interested to work in computer vision should have an initial domain knowledge of segmentation.

Thus, the far-reaching applicability, paramount significance, elemental nature and challenging area of segmentation collectively acted as the motivation behind this work. Choice of GMM for segmentation also has a number of reasons. They cannot be properly explained before a brief discussion on literature review in Chapter 2. Hence, it is broadly stated in Section 2.4.

### 1.6 Scope of this Work

The scope of the dissertation covers several techniques proposed to enhance the potential of conventional GMM towards improving its performance for image segmentation, foreground segmentation and video segmentation. Thus, the scope broadly covers GMM and some of its variants. Potential enhancements have been done individually for the three fields of segmentation covered in the scope. The enhancements are stated as follows:

1. **Image Segmentation:** An enhancement to the conventional GMM has been proposed using bilateral filter. The bilateral filter is applied using Markov Random Field (MRF) on the prior probabilities of each Gaussian distribution in the mixture model. Thus, a spatial constraint and filtering operation have been imposed to improve the quality of segmentation, even in the presence of noise.

2. **Foreground Segmentation:** An enhancement to the conventional GMM has been proposed by incorporating multiresolution decomposition on the data. This im-
Explicitly embodies spatial relationship among neighbouring pixels, and improves the quality of segmentation significantly.

3. Foreground Segmentation: An enhancement to the conventional GMM has been proposed with the inclusion of Adaptive Support Weights (ASW) and Histogram of Gradients (HOG) in the distance measure of the GMM. The advanced distance measure improves the clustering by GMM against background noise and unwanted outliers.

4. Video Segmentation: A novel approach has been proposed to use an enhanced GMM towards video segmentation while dynamically controlling the number of clusters.

To maintain the relevance to the scope, Chapter 2 consists of a brief literature review followed by a discussion on the application of the conventional GMM towards image segmentation and foreground segmentation. Subsequent four chapters discuss the enhancements, respectively. Due to the vastness of the research and the variety of challenges present in segmentation, the scope has been limited towards a number of common yet important problems in segmentation, as discussed next.

1.7 Problem Statement

In the essence of the previous discussion, the area of segmentation has aged considerably, and explored vastly, to turn up with a number of challenges requiring solutions. Keeping the past years of research into account, dealing with all of the problems in the limited scope of a PhD dissertation would be an inordinate challenge and possibly infeasible. Instead, the focus of the dissertation is to provide legitimate and practical solutions to some of the fundamental challenges by maintaining a balance between performance, scalability and accuracy. Specifically, the following problems have been discussed and worked on:
1. Image Segmentation: Three of the main problems for image segmentation are accuracy, computation and automatic processing. Unfortunately, the problems of accuracy and computation have an inverse relationship. Any technique targeted towards improving accuracy has a slower computational speed, and vice versa. Finally, to deal with the challenge of automatic processing, the techniques need to be free of manual interventions. As answering to all the challenges under accuracy would penalize computation, a subset of challenges is dealt with in the dissertation. The main focus is on the following challenges: NP, IOES and OI. Of course, as CV and LPOB are related, these challenges are also somewhat addressed. Thus, the main focus is to provide an automatic image segmentation approach which is computationally inexpensive while being robust against NP, IOES and OI.

2. Foreground Segmentation: Similar to image segmentation, foreground segmentation has a number of challenges. Also, it has a similar inverse relationship between accuracy and computation. The challenges of NB and DB are related while IBCF and RM may be resulted from SF. Finally, IV is completely different from any of the others. The dissertation focuses on providing automatic and computationally efficient foreground segmentation technique to handle most of the challenges related to accuracy.

3. Video Segmentation: Even though video segmentation has spent less time in development, its challenges are more well-defined. There are mainly four challenges: temporal coherence, automatic processing, scalability and computation. Temporal coherence representing accuracy, has an inverse relationship with computation as well as scalability. Coherent segmentation often requires large memory and processing power. However, the dissertation focuses on each of these problems to come out with a practical solution.
1.8 Objective

The objective of this PhD dissertation is to introduce some improvements in the fields of image, foreground and video segmentation through proposing a number of enhancements to the conventional GMM. The improvements are in terms of addressing the problems discussed in the problem statement (Section 1.7). Enhancements are proposed by imposing spatial and temporal constraints in the GMM through suggesting the use of several cues. For image segmentation, bilateral filtering through the use of MRF has been proposed to enhance GMM in order to improve the quality of segmentation. For foreground segmentation, two independent enhancements are proposed through use of multiresolution, and use of ASW with HOG. Finally, an efficient technique is proposed to merge a number of fundamental cues of image segmentation and foreground segmentation in order to enhance GMM towards video segmentation.

1.9 Organization of Thesis

The rest of the dissertation is organized as follows: Chapter 2 consists of an extensive literature review on image, foreground and video segmentation followed by the description of GMM applied towards image and foreground segmentation. The proposed enhancement to GMM using MRF for image segmentation is discussed in Chapter 3. An enhancement to GMM based on multiresolution for foreground segmentation is depicted in Chapter 4. Chapter 5 proposes an enhancement to GMM with an advanced distance measure based on ASW and HOG. Finally, the video segmentation method using GMM is introduced in Chapter 6 followed by drawing a conclusion and delineating some scopes for future work in Chapter 7.
Chapter 2

Literature Review

Segmentation has been a part of image analysis for more than 40 years. With so many application areas, it has been extensively studied. To discuss on related literature, it is more convenient to divide it in groups and subgroups for better understanding. Thus, past literature on segmentation has been divided in three primary sections: image segmentation (Section 2.1), foreground segmentation (Section 2.2) and video segmentation (Section 2.3).

2.1 Review on Image Segmentation

Rise of image segmentation techniques can be dated back to the proposal of first edge detection technique in 1965, with the introduction of Robert’s operator [35]. This was the first technique to extract meaningful “features” from an image. Since then, image segmentation has been experiencing continuous growth as well as diversity. Diversity came with the advent of colour image processing, extending from 2-D gray scale images to 3-channel colour images, and finally, to multi-channel images in current literatures. Thus, the research, application scope and diversity have expanded rapidly.

With such amount of massive research, discussion of even the major techniques would be a huge task. Instead, grouping techniques based on similarities or scope of application would make the discussion simpler and more straight-forward. However, grouping on such a large scale is also confusing due to several factors including fundamental similarities between groups, hybrid methodologies and subtle differences between methods residing in same group. Interested readers are encouraged to go
through the extensive discussions provided in [36]. Following related literature reviews, we can broadly divide the existing algorithms into the following categories:

1. Threshold based: These techniques rely on some global or local threshold values to segment an image into distinct regions. Often, such algorithms define some features on an image and threshold on the feature values to extract contours of segmented regions. Edge detection based techniques also fall in this group. Edge detectors like Roberts operator [35] (mentioned above), Canny [37], Sobel [38], Prewitt [39] are some of the examples. [40, 41, 42] also belong to this category. However, hard thresholding based methods, that depend on some constant threshold values, are very susceptible to noises, low contrast, low resolution and illumination problems as these methods cannot adapt to image contents. Often, bad choice of threshold leads to incorrect segmentation. Adaptive thresholding [43, 44, 45] can partially handle this problem. However, threshold based methods cannot provide proper solution to the stated problems [46].

2. Histogram based: Histogram has been one of the basic yet popular features in an image. It represents the frequency of occurrence or probability distribution of intensity values in an image. The peaks in a histogram represent the most frequent intensity levels in the image. A histogram containing multiple peaks, and hence, multiple clusters or lobes, can be thresholded to obtain segments. Example: an image having a bright object on a dark background has two dominant lobes. It can be thresholded into the object and the background by choosing an appropriate threshold between the two dominant lobes. Thus, histogram based methods [47] also belong to threshold based methods. However, they are very simple and popular, and deserve specific category. Otsu’s method [48] has been heavily used for automatic thresholding in histograms. However, these methods also suffer from the same problems as the threshold based methods.

3. Mean-shift based: Mean-shift analysis is a non-parametric, iterative procedure
introduced by Fukunaga [49] to determine the mode of a density function from sample values. This was generalized by Cheng for image analysis [50]. Finally, Comaniciu and Meer [51] extended this algorithm for colour image segmentation. There are improved variants for this algorithm in literature [52, 53, 54]. However, the main drawback of these algorithms is ignoring the spatial relationships in an image.

4. Clustering based: Clustering based methods divide an image into non-overlapping clusters based on some similarity criteria. Most common examples of such techniques are K-means [55, 56] and FCM [57, 58]. These two methods are very popular due to their implementation simplicity. Region-growing [59, 60], region split and merging [61] and watershed based methods [62, 63] also fall in this category. The category of clustering based methods is perhaps, the largest category in terms of members. In a general sense, Mean-shift is also a type of clustering, and has certain resemblance to methods in this category. Thus, similar to Mean-shift, these methods also mostly lack dependence on spatial constraints [64, 65, 66, 67].

5. Neural Network based: Artificial neural networks have been used to cluster images based on feature vectors extracted from such images [68, 69, 70]. Amartur et. al. [71] have used neural networks to segment an image by minimizing the distance between two feature vectors. The performance has been satisfactory for only a sub-group of images, as the method, in general, did not incorporate the spatial relationships in the images. These methods also require training of the neural network and generally suffer from under or over-training of the network.

6. Multi-scale based: Multi-scale based approaches exploit the idea that some of the image features are more dominant in coarser scale of an image, while some features are present in finer scales of an image. Here, scale represents the
resolution of an image. Some approaches incorporate local edge and regional information [72, 73, 74] while some approaches use multiresolution analysis for segmentation [75, 76]. These approaches use spatial information often combined with clustering [77] and have good performance. However, the level of decomposition plays a main role in performance, and thus, the methods require proper parameter tuning for different types of images.

7. Graph based: Graph based methods represent one of the most popular category of methods for image segmentation. The advent of such methods can be attributed to Greig [78], who, far back in 1989, proposed that the solution of Maximum A Posteriori (MAP) estimation for binary images can be computed using graph cuts. Although the idea did not draw too much attention, the idea was extended to N-D images and popularized by Boykov and Jolly [79]. They showed that graph cut can find the globally optimum segmentation based on a minimum cut algorithm. Also, a pioneering work on image segmentation had been done by Shi and Malik [80]. Since then, many works have been proposed based on graph [81, 82, 83, 84, 85]. In current literature, some of the best methods belong to this category. However, since these methods are based on minimizing segmentation costs, they remain susceptible to noises.

8. Statistical Model based: In recent years, statistical model based methods [86, 87, 88, 89, 90, 91, 92, 93] have been a popular substitute to graph based methods. These methods model the intensity distribution of an image using statistical tools, in turn modeling the noises and uncertainties in a probabilistic fashion. In this category, standard GMM has been one of the most popular methods [94, 95, 96, 97]. It is a flexible, simple yet powerful method to model multivariate data, and can be easily extended or enhanced. Although GMM has the advantages of simple architecture and fewer parameters, its results are susceptible to noise and illumination variations, as the conventional GMM does not take the spatial
dependency into account. In order to improve the performance, mixture models with MRF have been used [98, 99, 100]. However, as the parameters of GMM are determined through maximizing a log-likelihood function using Expectation-Maximization (EM) algorithm, the extensions using MRF make the job highly complex and computationally extensive. Various approximations have been proposed to handle this problem [101, 102]. Among these approximations, Spatially Variant Finite Mixture Model (SVFMM) [101] is one of the most popular approaches. However, these extensions with approximations remain more complex compared to the conventional model and susceptible to noise at segmentation boundaries. Also, conventional GMM gets easily affected by outliers. In search of a more sustainable model, Students-t Mixture Model (SMM) [103] has been proposed for image segmentation. SMM shows promising performance against outliers due to its heavily tailed distribution as compared to GMM. However, SMM also does not take the spatial dependency into account and suffers from similar problem.

Image segmentation being one of the highly studied types of segmentation in general, the related literature is also vast. This section has been an attempt to summarize the progress as clearly and gradually, as possible. Since the scope of this dissertation is based on GMM, a discussion on GMM and its variants are presented after the literature review on foreground segmentation and video segmentation.

2.2 Review on Foreground Segmentation

Foreground segmentation is a two class (or three class including shadows) problem under the hood of video segmentation. Though it is a relatively new field of research as compared to image segmentation, it has immensely grown in popularity due to its immediate application in tracking, surveillance, traffic monitoring as well as in several intermediate stages of segmentation. The area of foreground segmentation
ranges from simple foreground change detection to modeling the dynamic nature of the background. Based on the fundamental working principles of the methods in this area, the methods can be broadly classified into three distinct categories:

1. **Frame Difference based**: These methods calculate the difference between subsequent video frames, and obtain the amount of motion using a threshold on the difference. As an object moves, its position relative to the video frames changes over time. Thus, the frame difference is high where the object changes position, and it is low where a stable background exists [104, 105, 106]. Normally, for slow foreground, subsequent frame differencing would not yield a proper motion. Thus, three or multi-frame differencing is also applied [107]. Out of the works carried out, Wavelet based change detection methods [108, 109] have been a popular choice for their simplicity. The methods in this category are inexpensive in terms of execution speed. However, as they are only good for continuously moving objects, the methods often produce inaccurate or incomplete foregrounds, and suffer from noises due to dynamic background and slow foreground [110].

2. **Optical Flow based**: Optical flow refers to the instantaneous speed of pixels in the imaging surface. It is calculated using the temporal changes of each pixel in its neighbourhood and represented as a vector. The idea of optical flow was proposed by Gibson in 1950, even before the rise of image processing [111]. However, it was much later applied to image processing. The movements in the foreground of a video sequence create a 3-D velocity field. By computing the optical flow, this velocity field can be determined [112, 113]. Due to the nature of iterative computation, it is very expensive to compute optical flow. Without proper hardware for computation, it is not suitable for real-time processing.

3. **Virtual Coil based**: The name virtual coil was coined in reference to the similarity of working principle with electromagnetic induction coils. Induction coils
are triggered to produce magnetic flux in response to the variation of current
flow inside it. Similarly, the virtual coil is set as test lines or regions in images.
When vehicles pass this coil, the image in this RegionOfInterest (ROI) will
change. If the area of the image covered by the vehicle crosses certain thresh-
old, the vehicle is detected. The methods in this category [114, 115] have very
low computational cost. However, due to the detection being limited only to
the ROI, these methods cannot perform well in segmentation.

4. Background Subtraction based: This group of methods estimate the background
image and subtract each frame from this image. A threshold is applied to
the difference image to generate a foreground mask. The threshold can be
constant or dynamic depending on the method used. Due to the generation
of a background image, and subtracting each frame from the image, methods
in this category often suffer from noises due to dynamic backgrounds. The
use of single background image indicates that these methods are unimodal i.e.
use a single mode to represent background. Here, the term mode is used to
emphasize the fact that the most frequently used value(s) by a pixel represent
the background. If the background is represented by a number of values (either
a dynamic background, or a background represented by multiple modes having
same frequency), the methods in this category cannot properly represent these
multiple values.

Depending on the way of estimating the background image, these methods can
be categorized as follows:

(a) Temporal Averaging based: The methods in this subgroup keep the esti-
mate of background by computing a recursive updated average of a history
of pixel values over time. A learning rate is used to specify the weight be-
tween the current pixel value and the background pixel value for all pixels.
Finally, the foreground mask is obtained by subtracting current frame from

39
the estimated background. Although the algorithm has low computational
cost, the background is often affected by the appearance of the moving ob-
jects when the objects have different speeds of movement. A single learning
rate hardly suffice and most of the time, a tail behind moving objects is
produced. Averaging of instantaneous background is used in [116, 117] to
reduce tailing effect. Methods in this category [118, 119, 120, 121, 122]
are normally used only for low computational purposes. Recently, tem-
poral averaging has been combined with median estimation based back-
ground subtraction in: A vision-based system for elderly patients monitor-
ing (VISIONSYS) [123].

(b) Single Gaussian based: The pixel behaviour over time is represented by a
Gaussian distribution [124, 125, 126, 127]. Instead of using only mean for
temporal averaging, the variance of the Gaussian is also used. Thus, the
mean image and the variance image collectively represent the background.
A pixel is classified by locating its position with respect to the Gaussian
distribution. This is statistically equivalent to a dynamic threshold.

(c) Mode Estimation based: Median estimation has been used earlier in lit-
erature [128]. However, as stated earlier, mode is a better representative
of the background, or the most dominant background value (in case of
dynamic backgrounds). The mode is estimated in a constant time window
of $N$ frames. Mode based approaches are fast and relatively simpler in
implementation [129, 130]. However, the criterion for the methods to work
properly is that the background should be dominant in the time window.
If it is not, it won’t be detected. Thus, it is very sensitive to the window
length $N$ as well as the bin size of the histogram of values in the window. If
the bin size is too small and the background is spread over several values,
it would not produce a dominant peak. Again, a large bin would make it
harder to detect the correct value of the peak. Recently, multimodal form
of this methodology is introduced [131] with better results.

(d) $\Sigma - \Delta$ based: $\Sigma - \Delta$ based method was introduced in [13] and have been popular [132, 133, 134] due to its low complexity. It uses an approximation of the temporal median and the $\Sigma - \Delta$ variance to make a classification between background and foreground. The name comes from its similarity of operation to the $\Sigma - \Delta$ modulator based analog to digital converter of a continuously time varying signal. The mean and variance of the $\Sigma - \Delta$ are incremented or decremented at each time step by a value of 1, depending on the difference between the current pixel value and background. Finally, if the current pixel value is greater or less than that of the estimated background value by more than the $\Sigma - \Delta$ variance, it is classified as foreground. The main problem of the method is that pixels with continuous exposure to foreground will have high variance and subsequently the foreground detection would be lower.

(e) Kalman Filter based: Kalman filter has been used to estimate the background by temporally modeling the colour values of each pixel by a filter. The foreground is interpreted as noise for the filter. Illumination changes violate the principle of the filter as they represent non-Gaussian noises. However, solutions to the problem have been proposed in [135]. Foreground estimation using Kalman filter has shown good performances [136, 137]. A recent method based on Kalman filter has been proposed by Godbehere, Matsukawa and Goldberg (GMG) [138]. However, Kalman filter based methods suffer from high implementation complexity.

5. Graph based: A graph of a MRF can represent the problem of foreground segmentation by representing each pixel with a node in the graph. The sources represent the foreground and the background. A proper graph cut with smoothing constraint to prevent over-segmentation, can completely segment the source
and sink nodes and label each pixel to either foreground or background. Several applications of graph cut have been reported, including foreground segmentation [139, 140]. However, the time complexity and memory requirements are very high for practical applications.

6. Statistical Model based: Methods in this category model the background based on the temporal and spatial cues available. By modeling the background, each frame can be compared with the background to estimate the foreground motion. By keeping a model for the static or dynamic background, the moving objects in the foreground can be better segmented. Hence, methods in this category have better quality of results compared to the frame difference based or the background subtraction based methods. Due to the general definition of background modeling, this category covers a large number of methods [4, 141, 142, 143]. As the scope of the work is concentrated more on this category, some popular methods in this category are mentioned next.

(a) Kernel Density Estimate (KDE) based: KDE is a nonparametric method that can estimate the distribution of temporal values of a pixel over a given history [141, 144]. Each pixel is classified by calculating its probability of being part of the distribution or not. The Kernel estimator function is often chosen to be a Gaussian function. The colour channels are treated independently for simplicity. A pixel is classified as foreground if its probability is below a global threshold. KDE can properly represent static and dynamic backgrounds by modeling the real distribution of the values taken by a pixel over time. However, due to the global threshold, it often suffers from noises. Also, as KDE uses a small history to keep low computational cost, it cannot represent a long history, specifically for surveillance purposes.

(b) Codebook based: In [142, 145], a new type of nonparametric background
model was presented. In the codebook model, each pixel is represented by a codebook. A codebook is a compressed form of the background history over a long sequence of images. Each codebook comprises of numerous codewords that are made up of colour values transformed through some colour distortion metric. Spatial and temporal information have also been incorporated in advanced codebook based methods [146]. These methods take more time to learn and less memory to contain the codebook, and hence, are useful for practical surveillance applications. However, although runtime evolutions are possible for the methods [145], new codewords are not created with changes in the scene. Hence, the model cannot cope up if the background changes considerably for a long time. An example would be for abandoned objects.

(c) GMM based: GMM had been efficiently adapted for foreground segmentation by Stauffer and Grimson [4, 5]. The temporal history of a pixel is modeled by a mixture of Gaussian distributions. A pixel belongs to background if it is a part of a stable Gaussian distribution with high prior weight and low variance. Otherwise, it is part of the foreground. The conventional GMM can handle static as well as dynamic background using multiple Gaussians. Due to its effectiveness, a number of variants of the conventional model have been proposed in recent years [147, 148, 149, 150]. The Effective Gaussian Mixture Model (EGMM) [147] is one of the simpler and faster approaches. GMM has been well explored and applied for traffic analysis [151, 152]. However, the conventional model does not take the spatial dependency into account, and suffers from inaccurate segmentation. Taking that into account, Conditional Random Field based Gaussian Mixture Model (CRFGMM) [150, 153] has been proposed to use Conditional Random Field (CRF) incorporating pixel neighbourhood information in the learning process. Also, variable number of clusters has been proposed
along with shadow removal by Chen and Ellis as Self-Adaptive Gaussian Mixture Model (SAGMM) [32]. Recently, Type-2 Fuzzy GMM and Markov Random Field based method with Uncertain Mean (T2FMRF_Um) and uncertain variance (T2FMRF_Uv) have been proposed by Zhao et. al. Fuzzy logic has also been recently used by the method based on Fuzzy Adaptive Self-Organizing Maps (FASOM) [154].

Due to the limited scope of the dissertation, broad discussions on any of the related methods cannot be incorporated. However, as the scope mostly focuses on GMM, the conventional GMM is discussed in more details after the following discussion on related works for video segmentation.

2.3 Review on Video Segmentation

Spatio-temporal video segmentation is one of the areas that received less attention in earlier researches, due to the involvement of high complexity and memory requirements. Fortunately, the modern computer hardware has improved beyond expectations to provide the required architecture for the methods in this category. Hence, a number of methods [24, 25, 26, 155, 156] with impressive results have been proposed in recent times. In terms of application, video segmentation can be divided into two major categories as follows:

1. Noncausal: To keep coherence of voxel labels over a number of video frames, most of the methods in literature process several frames together. This requires that the methods are provided with the video frames prior to their execution. This is a noncausal approach as future video frames need to be present in order to segment current video frames. Most of the methods in literature can perform the same [157, 158, 159, 160] as the coherence is easier to keep if the complete data is available before processing. However, a high amount of processing power
in the range of 2 – 3 GHz as well as massive memory in the range of several gigabytes are required to keep processing data for a video of 500 frames with each frame having a dimension of 320 × 240 i.e. a standard surveillance video with duration of only 15 seconds. Clearly, methods in this category are not applicable in real-time or for long video sequences. In practical scenario, a surveillance video may range from 2 – 3 hours to several days. Thus, current research is mainly concentrated on achieving temporal coherence in a causal manner.

2. Causal: Causal methods do not require future frames in order to segment current video frames and maintain the coherence of segmentation for subsequent video frames. Due to the frames being provided in a streaming on-the-fly fashion, causal video segmentation is also termed as streaming video segmentation. The process is very difficult as the coherence cannot be properly propagated to future frames without knowing the frame contents beforehand. Very few approaches have been able to achieve the same [161, 162, 163, 164, 165].

Keeping coherence to the literature studies for image segmentation and foreground segmentation, we can broadly classify the proposed techniques for video segmentation based on the fundamental methodologies used as follows:

1. Mean-shift based: Mean-shift based methods consider each 3D point as a multi-dimensional feature point whose coordinates include the colour components, as well as the motion components of the 3D point. It has been applied for feature space analysis by Comaniciu and Meer [166]. Repeated mean-shift operation cluster the video in a spatio-temporal segmentation. The idea was initially proposed by Leung et. al. [167] and successfully extended by DeMenthon [157]. Since then, mean-shift has been well adapted for video analysis due to its low execution complexity and simple architecture. Wang et. al. proposed the use of anisotropic kernel mean-shift for image and video segmentation [168] as well
as for video tooning [159]. Freedman and Kisilev [169] applied a sampling-based fast mean-shift approach to a cluster of 10 frames as a larger set of image features. However, they have not used the temporal information. A recent topological approach based on mean-shift (henceforth termed as MSHVS) has been proposed by Paris and Durand for hierarchical segmentation [155].

2. Tracking based: These methods generally define segments at video frame-level. They use colour, motion or other spatial cues to enforce temporal coherence [170, 170]. Following the same line, Brendel and Todorovic [160] used contour cues. This allowed splitting and merging of segments to boost the tracking performance.

3. Kalman filter based: Kalman filter has also been applied for video segmentation [171, 172]. Though the methods are causal and have shown good performance, works in this category have not extended considerably. Thus, the existing works do not have improved results as compared to some of the recent methods for video processing.

4. Graph-based: Perhaps the most common and popular methods for spatio-temporal video segmentation fall in this category. Graph based methods have shown promising performance [156, 24]. Most of the methods generate and keep a supervoxel graph for the entire video sequence. For noncausal methods, the graph is constructed based on all video frames together [24, 25]. This requires a large memory and high amount of processing power. On the other hand, streaming methods in this category, generate a graph based on the first few frames, and update on future frames. Although the methodology reduces burden on the processor and memory, the quality also suffers [164, 165]. Among the graph-based methods, Efficient Graph-Based Image Segmentation (GB) [156] and Efficient hierarchical graph-based video segmentation (GBH) [24] methods have shown promising performance. In the sub-category of graph-cuts, SWA [25, 173] and
Spectral grouping using the Nyström method (NYS) [26] have proven their excellence. In a recent study carried out by Xu and Corso [23], they have compared 5 video segmentation methods (Mean-Shift based Hierarchical Video Segmentation (MSHVS), SWA, NYS, GB and GBH) based on supervoxel analysis for desirable video segment properties, such as spatiotemporal uniformity and coherence, explained variation, and spatiotemporal boundary extraction, on a human-labeled video benchmark. The study reports that SWA and GBH generate supervoxels with most desirable properties. In other words, SWA and GBH are the best methods for spatio-temporal video segmentation in current literature. However, both SWA and GBH are noncausal and require the total video sequence before computation. The approximation framework for GBH i.e. the streaming GBH [165] successfully segments a video sequence in a streaming fashion; however, its performance is lower in comparison to GBH.

5. Interactive: For interactive video segmentation, the user is often required to provide a graphical input in the form of seed pixels, scribbles or sometimes approximate boundaries in single or multiple frames to initiate or facilitate the video segmentation [174, 159]. It has recently gained popularity and shown significant progress [175, 158, 176, 177]. The segmentation is often of very high quality due to the presence of user input. However, the problem of manual intervention limits the use of these methods to specific domains.

6. Statistical model based: modeling has been adapted for video segmentation also. Paris et al. derived the equivalent tool of mean-shift image segmentation for video streams based on Gaussian Kernels [178], and achieved real-time performance without considering future frames. Generalized GMM has been successfully applied to spatio-temporal video segmentation in a noncausal mode [179, 180]. However, even after the success of GMM for image segmentation as well as foreground segmentation, it has not been successfully applied
to real-time spatio-temporal video segmentation. Also, the current statistical model based methods do not provide results comparable to those of GBH and SWA.

This concludes the brief discussion on related works in the fields of image segmentation, foreground segmentation and video segmentation. Subsequent discussions are dedicated to the scope of the dissertation.

2.4 Why Gaussian Mixture Model?

According to the related works summarized in the previous subsections, the two types of technologies common to image segmentation, foreground segmentation as well as video segmentation, are graph based techniques and statistical modeling techniques. Both of the categories have been researched on, and have shown promising performances. GMM is part of statistical modeling based techniques. The reasons behind choosing GMM for the task of segmentation are explained as follows.

1. Graph based techniques have performed reasonably well for image segmentation as well as video segmentation. However, their application towards foreground segmentation has been limited due to the implementation and execution complexity. As scalability and fast performance are the main requirements for a foreground segmentation algorithm, graph based techniques have been relatively less popular. On the other hand, statistical modeling of background, and in particular, modeling by GMM has been immensely popular due to its simple architecture, real-time performance and extensibility. The role of GMM in image segmentation is also of high importance and many variations of it has been proposed. Thus, GMM stood its ground for image segmentation as well as foreground segmentation.

2. As compared to graph based methods, statistical modeling have actually not
been popular in case of video segmentation. However, two reasons still exist for their use. Firstly, most of the graph based techniques still suffer from the inherent problems of video segmentation: temporal incoherence, large memory usage, high computational burden and hence, scalability issues. On the other hand, statistical modeling has shown to be scalable for both image segmentation and foreground segmentation. Secondly, GMM has not been properly researched on for video segmentation. Thus, there is room for improvement.

Based on the above reasons, GMM shows high potential for segmentation. The choice of GMM for the dissertation work is mainly based on its extensibility. It has been repeated shown in existing literature, that the capabilities of conventional GMM can be enhanced by incorporating spatial and/or temporal cues. This had been the fundamental motivation behind the choice of GMM and the effort to further enhance its potential.

Two subsequent sections are devoted to explaining the application of conventional GMM for image segmentation, and foreground segmentation. The applications are important, and both of the applications are joined in a hybrid methodology in Chapter 6 to propose a GMM based video segmentation algorithm.

### 2.5 Mathematical Notations

The best of efforts has been put to maintain a consistency of notations throughout the dissertation. Uppercase bold roman letters, such as \( M \), denote matrices. Column vectors are denoted by lowercase bold Roman letters such as \( \mathbf{v} \) whenever possible. In some cases, where symbols and math typefaces are involved, Roman form is not used. However, the lowercase bold form is maintained throughout the dissertation. All vectors are considered as column vectors, if not mentioned otherwise. Parameters and constants are denoted by uppercase letters, such as \( C \). Finally, variables are denoted by lowercase letters such as \( x \). The transpose is denoted by \( T \) such that \( \mathbf{v}^T \)
denotes a row vector.

For the rest of the dissertation, a colour image-frame matrix of dimension $P \times Q \times D$ is denoted as $F_{P \times Q \times D}$ or simply $F$ as the dimensions are generalized. Similarly, a video matrix of duration $M$ is represented as $V_{M \times P \times Q \times D}$ or simply as $V$. An image-frame at time $t$ is denoted as $F_t$. When all pixels in an image-frame are stacked in a row-wise manner, we get a matrix $X_{N \times D} = (x_1; x_2; \ldots; x_N)$ where, $N = P \times Q$ represents the data-size, and $x_i$ represents $i^{th}$ pixel at position $(u, v)$ in the original image. For a row-wise stacking, $i$ represents a linear index, and is related to pixel position $(u, v)$ as follows: $i^{th} = (u - 1) \times Q + v$. Similarly, the $i^{th}$ voxel at position $(t, u, v)$ in the original image-frame with respect to the video $V$, is denoted as $x_{t,i}$. The notation is kept similar to a pixel as the voxel $x_{t,i}$ is nothing but the pixel $x_i$ at time $t$. Thus, for the rest of the dissertation, the voxel $x_{t,i}$ would be referred to as the pixel $x_i$ at time $t$.

### 2.6 Image Segmentation based on Conventional Gaussian Mixture Model

Let, $x_i; i = (1, 2, \ldots, N)$, where each $x_i$ is of dimension $D$, denote an observation at the $i^{th}$ pixel of an image. The neighbourhood of the $i^{th}$ pixel is presented by $N_i$. The target is to associate each $x_i$ with a label in $(1, 2, \ldots, K)$. For this classification, standard GMM assumes that each observation $x_i$ is independent of the label $\Omega_j$. The density function $f(x_i \mid \pi, \Theta)$ at an observation $x_i$ is given by:

$$f(x_i \mid \pi, \Theta) = \sum_{j=1}^{K} \pi_j \Phi(x_i \mid \Theta_j) \quad (2.1)$$

where, $\pi = \{\pi_j\}; j = (1, 2, \ldots, K)$ is the set of prior distributions of probabilities where $\pi_j$ denotes the probability that pixel $x_i$ is in label $\Omega_j$ and satisfies the constraints:

$$0 \leq \pi_j \leq 1 \text{ and } \sum_{j=1}^{K} \pi_j = 1 \quad (2.2)$$
Also, $\Phi(x_i | \Theta_j)$ is a component of the gaussian mixture. Each component can be written in the form:

$$\Phi(x_i | \Theta_j) = \frac{1}{|\Sigma_j|^{1/2}} \exp \left\{ -\frac{\Delta^2}{2} \right\}$$

(2.3)

where $\Delta^2 = (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)$ is the squared Mahalanobis distance and $\Theta_j = \{\mu_j, \Sigma_j\}; j = (1,2,\ldots,K)$. The $D$-dimensional vector $\mu_j$ is the mean, the $D \times D$ matrix $\Sigma_j$ is the covariance, and $|\Sigma_j|$ denotes the determinant of $\Sigma_j$. From Eq. 2.1, the joint conditional density of the data set $X$ can be written as:

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

(2.4)

Given the joint conditional density from Eq. the log-likelihood function of the standard GMM [181] is given by:

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

(2.5)

Where $\Theta = \{\Theta_j\}; j = (1,2,\ldots,K)$. As can be observed from the log-likelihood function, GMM has a simple form with very few parameters. The EM algorithm is used to maximize the log-likelihood function in Eq. 2.5 as described in the next section.

### 2.6.1 Expectation-Maximization for Gaussian Mixture Model

To find the parameter values in order to maximize the log-likelihood presented in Eq. 2.5, we need to differentiate the log-likelihood with respect to each of the parameters and equate the derivatives to zero. Firstly, we do the same for the means $\mu_j$. The obtained expression is as follows:

$$\frac{\partial L(\Theta, \pi|X)}{\partial \mu_j} = \sum_{i=1}^{N} \frac{\pi_j \Phi(x_i | \Theta_j)}{\sum_{k=1}^{K} \pi_k \Phi(x_i | \Theta_k)} \Sigma_j^{-1}(x_i - \mu_j) = 0$$

(2.6)

Here, we have used the form of Gaussian distribution mentioned in Eq. 2.3. The part in the fractions at the right-hand side is the posterior probability $z_{ij}$ of $x_i$ to belong to label $\Omega_j$ as follows:
\[ z_{ij}^{(t)} = \frac{\pi_j^{(t)} \Phi(x_i | \Theta_j^{(t)})}{\sum_{k=1}^{K} \pi_k^{(t)} \Phi(x_i | \Theta_k^{(t)})} \]  

(2.7)

Where, \( t \) indicates the iteration step. As EM needs to converge through iterations, the parameters are updated iteratively. The iteration step \( t \) is superscripted in order to separate it from the time instant \( t \), which is a subscript. The solution of \( \frac{\partial L(\Theta, \pi | X)}{\partial \mu_j} = 0 \) yields the optimum value of \( \partial \mu_j \) at the \( (t + 1) \) iteration step in order to maximize the log-likelihood:

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

(2.8)

Here, \( N_j = \sum_{i=1}^{N} z_{ij}^{(t)} \) can be interpreted as the effective number of points assigned to cluster \( j \). Similarly, if we set the derivative of \( L(\Theta, \pi | X) \) in Eq. 2.5 with respect to \( \Sigma_j \) to 0, and simplify the final expression, we get the optimum value for \( \Sigma_j \):

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

(2.9)

Finally, we need to maximize \( L(\Theta, \pi | X) \) with respect to the prior distribution or weights \( \pi_j \). Here we must take account of the constraint in Eq. 2.2, which requires the prior distribution \( \pi_j \) to sum to one. This can be achieved by using a Lagrange multiplier \( \lambda \) and maximizing the following quantity:

\[ L(\Theta, \pi | X) + \lambda \left( \sum_{j=1}^{K} \pi_j - 1 \right), \]  

(2.10)

which yields the following expression:

\[ 0 = \sum_{i=1}^{N} \frac{\Phi(x_i | \Theta_j)}{\sum_{k=1}^{K} \pi_k \Phi(x_i | \Theta_k)} + \lambda. \]  

(2.11)

If we now multiply both sides by \( \pi_j \) and sum over \( j \) using the constraint in Eq. 2.2, we obtain \( \lambda = -N \). Using the value of \( \lambda \), rearranging Eq. 2.11 and making use of the definition of posterior probabilities in Eq. 2.7, we get:

\[ \pi_j^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} z_{ij}^{(t)} = \frac{N_j}{N}. \]  

(2.12)
Thus, the updated value of the $j^{th}$ prior weight is nothing but the fraction of data points associated to label $\Omega_j$. We summarize the steps of the EM algorithm for GMM below:

1. Initialize the means $\mu_j$, covariances $\Sigma_j$ and the prior weights $\pi_j$.

2. **E Step**: Evaluate the posterior probabilities $z_{ij}^{(t)}$ in Eq. 2.7 using the current parameter values.

3. **M Step**: Re-estimate the parameters for the next iteration from the Eqs 2.8, 2.9 and 2.12 using the current value of posterior: $z_{ij}^{(t)}$.

4. Evaluate the log-likelihood $L(\Theta, \pi | X)$ from Eq. 2.5 and check for convergence of either the log-likelihood or the parameter values. If the convergence criterion is not satisfied, return to step 2.

### 2.7 Foreground Segmentation based on Conventional Gaussian Mixture Model

In conventional GMM, the values of a particular pixel over time is termed as “pixel process”. Thus, the pixel process is a set that consists of scalar gray values for gray scale images, or vector of colour values for colour images. At time $t$, the history of $i^{th}$ pixel at position $(u,v)$ consists of the set

$$\{x_{1,i}, ..., x_{t,i}\} \text{ with } x_{t,i} = F_t(u,v),\quad (2.13)$$

where, time $t \in [1, t]$. The dynamic nature of the pixel process needs an adaptive mixture model for effective representation. The recent history of a pixel can be modeled as a mixture of $K$ Gaussians, as

$$f(x_{t,i} \mid \pi_{t,i}, \Theta_{t,i}) = \sum_{j=1}^{K} \pi_{t,i,j} \Phi(x_{t,i} \mid \Theta_{t,i,j}) \quad (2.14)$$
Here, \( \pi_{t,i} = \{\pi_{t,i,j}; j = (1, 2, ..., K) \) is the set of prior distributions of probabilities or simply, weights at time \( t \), and \( \Phi(\cdot) \) is the Gaussian probability density function with \( \Theta_{t,i,j} = \{\mu_{t,i,j}, \Sigma_{t,i,j}\}; j = (1, 2, ..., K) \). The distribution at time \( t \) is expressed as follows:

\[
\Phi(x_{t,i} | \Theta_{t,i,j}) = \frac{1}{\sqrt{2\pi D}} \exp \left\{ -\frac{\Delta^2}{2} \right\}
\]

(2.15)

where, \( \Delta = \sqrt{(x_{t,i} - \mu_{t,i,j})^T \Sigma_{t,i,j}^{-1} (x_{t,i} - \mu_{t,i,j})} \) represents the Mahalanobis distance. For reduction in computation, covariance matrix \( \Sigma_{t,i,j} \) is assumed to be of a diagonal form: \( \sigma^2_{t,i,j} I_{D \times D} \) with \( I_{D \times D} \) denoting the identity matrix, and \( \sigma_{t,i,j} \) denoting the Standard Deviation (SD) of \( j^{th} \) Gaussian. This implicitly assumes independence among the components along different dimensions, i.e., the colour channels. It also assumes every channel to have the same variance. These assumptions, although not completely correct, avoid costly matrix inversion at the cost of slight decrease in accuracy. Another important factor to notice here is the subscript of \( t \) and \( i \) with every parameter. That signifies that the distributions are per-instant as well as per-pixel unlike in image segmentation, where the parameters do not depend on a single pixel’s position or on the time-step, and are global. Of course, the constraints on priors still hold as follows:

\[
0 \leq \pi_{t,i,j} \leq 1 \quad \text{and} \quad \sum_{j=1}^{K} \pi_{t,i,j} = 1
\]

(2.16)

At this point, the understanding of the vastness needs to be apprehended. Unlike image segmentation, the pixels from a single image-frame do not represent the data to be modeled by the GMM. Instead, the GMM models a single pixel’s history of values over time. Thus, image segmentation requires one GMM consisting of \( K \) Gaussian distributions, whereas, foreground segmentation requires \( N \) GMM each consisting of \( K \) Gaussian distributions, and each GMM modeling a single pixel’s pixel process. As the conventional EM algorithm is iterative and iterates over each parameter value of a GMM, solution of foreground segmentation requires simultaneous processing on \( N \)
GMM using EM algorithm. This is infeasible and cannot provide even approximate real-time performance. Thus, other ways of optimizing the parameter values are required.

To update the parameters for each per-pixel distribution, an online recursive filter based GMM was proposed by Stauffer and Grimson [4, 5]. Following this model, every new pixel is compared against the $K$ Gaussian means. A match is found if the new pixel value $x_{t,i}$ is within a multiple of standard deviation from the mean. Mathematically, it can be written as

$$x_{t,i} \in \Phi(x_{t,i} | \mu_{t,i,j}, \Sigma_{t,i,j}) \text{ if } |x_{t,i} - \mu_{t,i,j}| < T \sigma_{t,i,j},$$

(2.17)

where, $T$ is a constant multiplier of standard deviation, normally lying between 2.5 – 3.5. For the matched distribution(s) (there may be more than one matched distribution), $\pi_{t,i,j}$, $\mu_{t,i,j}$ and $\sigma_{t,i,j}$ are updated according to the recursive formulations as

$$\pi_{t,i,j} = (1 - \alpha)w_{t-1,i,j} + \alpha;$$
$$\mu_{t,i,j} = (1 - \rho)\mu_{t-1,i,j} + \rho x_{t,i};$$
$$\sigma_{t,i,j}^2 = (1 - \rho)\sigma_{t-1,i,j}^2 + \rho (x_{t,i} - \mu_{t,i,j})^T(x_{t,i} - \mu_{t,i,j}),$$

(2.18)

where, $\alpha$ and $\rho$ are the learning rate and learning factor respectively. These parameters can be tuned for optimal performance depending on the application. For unmatched distributions, $\mu_{t,i,j}$ and $\sigma_{t,i,j}$ remain same, while the prior weight is reduced by a factor of $(1 - \alpha)$. If none of the distributions match the current pixel value, the distribution with lowest weight is replaced by a distribution with an initial low weight, $x_{t,i}$ as mean and a high variance. Next, the distributions are ordered by the descending values of $\pi/\sigma$ to determine the background, as the background supposed to be consisting of distribution(s) with highest weight(s) and lowest variance(s). The first $B$ distributions are chosen as the background for which the following holds

$$B = \arg \min_b \left( \sum_{j=1}^{b} \pi_{t,i,j} > Th \right),$$

(2.19)
Table 2.1 – Notational simplifications for foreground and video segmentation

<table>
<thead>
<tr>
<th>Original Notation</th>
<th>Reduced Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{t,i}$</td>
<td>$x_t$</td>
</tr>
<tr>
<td>$\mu_{t,i,j}$</td>
<td>$\mu_{t,j}$</td>
</tr>
<tr>
<td>$\sigma_{t,i,j}$</td>
<td>$\sigma_{t,j}$</td>
</tr>
<tr>
<td>$\Sigma_{t,i,j}$</td>
<td>$\Sigma_{t,j}$</td>
</tr>
<tr>
<td>$\pi_{t,i,j}$</td>
<td>$\pi_{t,j}$</td>
</tr>
<tr>
<td>$\Theta_{t,i}$</td>
<td>$\Theta_t$</td>
</tr>
<tr>
<td>$\Phi(x_t \mid \Theta_{t,i,j})$</td>
<td>$\Phi(x_t \mid \Theta_{t,j})$</td>
</tr>
<tr>
<td>$f(x_t \mid \pi_{t,i}, \Theta_{t,i})$</td>
<td>$f(x_t \mid \pi_{t}, \Theta_{t})$</td>
</tr>
<tr>
<td>$BG_{t,i}$</td>
<td>$BG_t$</td>
</tr>
</tbody>
</table>

where, $Th$ is a threshold that determines the minimum amount of data constituting the background. If a single distribution is chosen, the mean of the distribution would represent the background intensity value. Otherwise, an average of $B$ means $\{\mu_{t,i,b}\}$ weighted according to their prior weights $\{\pi_{t,i,b}\}$, would represent the background intensity $BG_{t,i}$ as shown below:

$$BG_{t,i} = \frac{1}{B} \sum_b \pi_{t,i,b} \mu_{t,i,b}. \quad (2.20)$$

The complete process is simple and feasible for real-time video processing systems. As it does not use the EM algorithm, the accuracy of segmentation is not comparable. However, it has shown reasonably good performance in case of foreground segmentation.

**Notational simplification:** Due to the per-pixel distribution, the symbols contain large subscripts and look clumsy. As can be observed from the equations in this section, we do not actually use a collective function for the whole image-frame that needs summation or grouping over $i$ denoting particular pixel positions. Thus, for
Chapters 4, 5 and 6, the notation would not use $i$. Consequently, the symbols are reduced as shown in Table 2.1. However, Chapter 3 deals with image segmentation, and would use $i$ as subscript.
Chapter 3

Bilateral Filter Based Mixture Model For Image Segmentation

One of the main drawbacks of conventional GMM is that the prior distribution $\pi_j$ does not depend on the pixel index $i$ and thus, on the spatial relationships between the labels of neighbouring pixels. Thus, the segmentation is extremely noise-prone and illumination dependent. To overcome this disadvantage, mixture models with MRF have been employed for pixel labeling, as already discussed in Chapter 2. The distinct difference is that the prior distribution $\pi_{ij}$ varies for every pixel $x_i$ corresponding to each label $\Omega_j$ and depends on the neighbouring pixels and the corresponding parameters. The disadvantages of the MRF based methods lie in lacking robustness against high amount of noise and increase in computational cost. Research has been done to extend the models [102] where an MRF models the joint distribution of the priors of each pixel, instead of the joint distribution of the pixel labels.

In this work, a new MRF based mixture model is proposed. The model is made based on the following considerations - firstly, the model is very simple compared to the other MRF based models. The structure of the model has been reduced to simple filtering in probability domain. Secondly, the spatial information has been successfully incorporated in the model with the use of bilateral filtering and the EM-algorithm can be directly applied to compute the parameters of the method.

The research work has been organized in following sections. In section 3.1, the background of the current work is briefly discussed. The proposed method is described in section 3.2. Section 3.3 provides brief understanding of the Bilateral filtering used for the work. Section 3.4 includes the experimental results and finally, the work is
3.1 Mixture Model based on Markov Random Field

Although the following discussion is similar to the one presented in Section 2.6, the particular interesting part is the per-pixel prior distribution $\pi_{ij}$. As before, $x_i$ denotes an observation at the $i^{th}$ pixel of an image. The neighbourhood of the $i^{th}$ pixel is presented by $N_i$. The target is to associate each $x_i$ with a label in $(1, 2, \ldots, K)$. For this classification, standard GMM assumes that each observation $x_i$ is independent of the label $\Omega_j$. The density function $f(x_i | \Pi, \Theta)$ at an observation $x_i$ is given by:

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

(3.1)

where, $\Pi = \{\pi_{ij}; i = (1, 2, \ldots, N), j = (1, 2, \ldots, K)\}$ is the set of prior distributions of probabilities where $\pi_{ij}$ denotes the probability that pixel $x_i$ is in label $\Omega_j$ and satisfies the constraints:

$$0 \leq \pi_{ij} \leq 1 \text{ and } \sum_{j=1}^{K} \pi_{ij} = 1$$

(3.2)

As stated before, the difference between Eq. 3.1 and Eq. 2.1 is the use of a per-pixel prior $\pi_{ij}$. As the set of prior distributions have two dimensions along $N$ and $K$, it is represented by the matrix $\Pi$ in comparison to the vector $\pi$ in Section 2.6. Also, $\Phi(x_i | \Theta_j)$ is a component of the gaussian mixture. Each component can be written in the form:

$$\Phi(x_i | \Theta_j) = \frac{|\Sigma_j|^{-1/2}}{(2\pi)^{D/2}} \exp \left\{ - \frac{\Delta^2}{2} \right\}$$

(3.3)

where $\Delta^2 = (x_i - \mu_j)^T \Sigma_j^{-1} (x_i - \mu_j)$ is the squared Mahalanobis distance and $\Theta_j = \{\mu_j, \Sigma_j\}; j = (1, 2, \ldots, K)$. The $D$-dimensional vector $\mu_j$ is the mean, the $D \times D$ matrix $\Sigma_j$ is the covariance, and $|\Sigma_j|$ denotes the determinant of $\Sigma_j$. From Eq.(3.1), the joint conditional density of the data set $X = (x_1, x_2, \ldots, x_N)$ can be written as:

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

(3.4)
This modeling has a fundamental problem. Since the observation $x_i$ is considered to be independent given the pixel label, the spatial correlation between the neighbouring pixels is not taken into account. In natural images, the neighbouring pixels are highly correlated if they belong to same object. If the correlation is not used, the segmentation can be very sensitive to noise, varying illumination and other environmental factors such as wind, rain or camera movements. MRF was introduced for segmentation in order to use this spatial information and has the following form:

$$p(\Pi) = Z^{-1} \exp \left\{ -\frac{1}{T} U(\Pi) \right\}$$

(3.5)

where, $Z$ is a normalizing constant, $T$ is a temperature constant set to 1 ($T = 1$), and $U(\Pi)$ is the smoothing prior. The posterior probability density function given by Bayes rules can be written as:

$$p(\Pi, \Theta \mid X) \propto p(X \mid \Pi, \Theta) p(\Pi)$$

(3.6)

By incorporating 3.4 and 3.5 into 3.6, the log-likelihood of 3.6 can be derived as:

$$L(\Pi, \Theta \mid X) = \log p(\Pi, \Theta \mid X)
= \sum_{i=1}^{N} \log \left\{ \sum_{j=1}^{K} \pi_{ij} \Phi(x_i \mid \Theta_j) \right\} - \log Z - \frac{1}{T} U(\Pi)$$

(3.7)

Depending on the type of energy $U(\Pi)$ selected in Eq.(3.7), we can have different kinds of models. Different researchers have used different expressions for this energy function to successfully incorporate local information into the approach. But, this incorporation increases the complexity of the method and may not provide robustness against noise. Also, in order to maximize the log-likelihood function with respect to parameters $\Pi$ and $\Theta$, an iterative EM algorithm needs to be applied. Due to the complexity of the log-likelihood function and the constraint in Eq.(3.2) to be satisfied, the M step of the EM algorithm cannot be directly applied to the prior distribution $\pi_{ij}$. Thus, the methods tend to become complex to solve the constrained optimization problem.
3.2 The Proposed Method

The proposed method is based on the fact that the energy function $U(\Pi)$ incorporates the spatial relationship among neighbouring pixels and is a smoothing function that reduces the classification ambiguity between neighbouring pixels. The method has been introduced keeping in mind that it should reduce the misclassification noise and in process, should not increase the computational complexity of the GMM.

In keeping with the above, we can refer to the following assumption. If the posterior probability of the $i^{th}$ pixel for the $j^{th}$ label is termed as $z_{ij}$, then the set $z_j = \{ z_{ij} \}; i = (1, 2, ..., N)$ for $j = (1, 2, ..., K)$ represents a posterior probability map which is smoothed using the energy function $U(\Pi)$ based on the neighbouring relationship among the $z_{ij}$ values. This assumption leads us to use image processing filters for smoothing this posterior probability map. Let, $\tilde{z}_j$ represent the filtered posterior probability map after applying a filter to $z_j$. If the elements of $\tilde{z}_j$ are termed as $\tilde{z}_{ij}$, then we use the following approach to incorporate the spatial information into the smoothing prior $U(\Pi)$ as follows:

$$U(\Pi) = - \sum_{i=1}^{N} \sum_{j=1}^{K} \tilde{z}_{ij}^{(t)} \log \pi_{ij}^{(t+1)}$$  \hspace{1cm} (3.8)

where, $t$ indicates the iteration step. An important concern when applying a smoothing filter to $z_j$ is the edges where probability changes suddenly. This leads to application of an edge-preserving filter, which is discussed in detail in section 3.3. Considering a smoothed $\tilde{z}_j$, the MRF distribution $p(\Pi)$ in Eq.(3.5) is given by:

$$p(\Pi) = Z^{-1} \exp \left\{ \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{K} \tilde{z}_{ij}^{(t)} \log \pi_{ij}^{(t+1)} \right\}$$  \hspace{1cm} (3.9)

Given the MRF distribution $p(\Pi)$, the log-likelihood function in Eq.(3.7) is written...
in the form:

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

$$+ \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \log \pi_{ij}^{(t+1)}$$

(3.10)

Applying the complete data condition, maximization of $L(\Pi, \Theta \mid X)$ will lead to an increase in the value of the objective function $J(\Pi, \Theta \mid X)$ given by:

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

$$- \log Z + \frac{1}{T} \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \log \pi_{ij}^{(t+1)}$$

(3.11)

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

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

(3.12)

The next objective is to optimize the parameter set $\{\Pi, \Theta\}$ in order to maximize the objective function $J(\Pi, \Theta \mid X)$ in Eq.(3.11). For simplicity, $Z$ and $T$ in Eq.(3.11) are set equal to one ($Z = 1, T = 1$). From Eq.(3.11) and using Eq.(3.3), the objective function can be rewritten as:

$$J(\Pi, \Theta \mid X) =$$

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

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

$$+ \sum_{i=1}^{N} \sum_{j=1}^{K} z_{ij}^{(t)} \log \pi_{ij}^{(t+1)}$$

(3.13)

To maximize this function, the EM algorithm is applied where the derivative of $J(\Pi, \Theta \mid X)$ is taken with respect each parameter in the parameter set $\{\Pi, \Theta\}$.
and equating it to zero. The solution to \( \partial J / \partial \mu^{(t+1)j} = 0, \partial J / \partial \Sigma_j^{-1(t+1)} = 0 \) would provide the minimizer of \( \mu_j \) and \( \Sigma_j \) respectively, at the \((t+1)\) step. It can be proven using simple vector differentiation, the minimizer values are:

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

(3.14)

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

(3.15)

For the prior distribution \( \pi_{ij}^{(t+1)} \), the solution to \( \partial J / \partial \pi_{ij}^{(t+1)} = 0 \) must also satisfy the constraints in Eq.(3.2). To enforce the constraint, the Lagranges multiplier \( \lambda_i \) for each data point is used to get the following equation:

\[
\frac{\partial}{\partial \pi_{ij}^{(t+1)}} \left[ J - \sum_{i=1}^{N} \lambda_i \left( \sum_{j=1}^{K} \pi_{ij}^{(t+1)} - 1 \right) \right]
\]

(3.16)

Eq.(3.16) can be solved using the constraint \( \sum_{j=1}^{K} \pi_{ij}^{(t+1)} = 1 \) to yield the following solution:

\[
\pi_{ij}^{(t+1)} = \frac{z_{ij}^{(t)} + \tilde{z}_{ij}^{(t)}}{\sum_{k=1}^{K} (z_{ik}^{(t)} + \tilde{z}_{ik}^{(t)})}
\]

(3.17)

Thus, using Eq.(3.14), (3.15) and (3.17), the optimum parameter values can be obtained that minimize \( J \) and hence, \( L \).

### 3.3 Bilateral Filtering

In Sec.3.2, it was mentioned that the smoothed posterior probability map \( \tilde{z}_j \) is obtained using some image processing filter on the posterior probability map \( z_j \). A smoothing filter removes noise but at the same time blurs the image so that the edge information in the image is reduced. In segmentation, edges carry high importance and the borderline between two distinctly segmented regions is decided by how strong the edges are. Also, the edges in \( z_j \) correspond to edges in the image because, in general, an edge signifies two clusters and hence, two different probabilities. This
leads us to apply a filter that can preserve the edge information in high extent while smoothing the map (Note: In $z_j$, an edge actually corresponds to a sudden change in probability). Bilateral filtering, in simple terms, is an \textit{edge-preserving smoothing} filtering technique. Here, each pixel value (probability value in this case) is replaced by a weighted average of intensity values from neighbouring pixels based on Gaussian distributions that are based on both the Euclidean distance and the range of intensity values of the neighbouring pixels. Due to the combined distance based smoothing and intensity range based smoothing approach, the filter achieves the desired edge preservation.

When there is a non-edge region, the neighbouring pixels have similar intensity and thus, bilateral filter acts as a standard smoothing filter that averages the noisy pixels with neighbouring pixels. But, at the edges where there is a sudden change in intensity, part of the neighbourhood have dark intensity and the rest are bright. In this case, due to a normalizing function, the center pixel value is replaced by the averaging values of the pixels in its vicinity. Thus, if the pixel belongs to dark region, its value will most likely be replaced by averaging the dark pixel values in its neighbourhood. Similar reasoning applies for bright pixels. For mathematical basis of Bilateral filtering, the readers are referred to [182].

\section*{3.4 Experimental Results}

The proposed algorithm has been tested on the images from the Berkeley Image Segmentation Data Set (BSDS500). The algorithm has been extensively compared with K-means, FCM [58], conventional GMM, SMM [103] and SVFMM [101] algorithms which are some of the popular and leading methods for segmentation. The methods were run until convergence. Also, comparison has been done on the BSDS500 region benchmarks with the best image segmentation algorithms available, using the measures PRI and VoI. The experimentation has been divided into two categories -
Table 3.1 – Performance of the proposed image segmentation method with varying level of noise and varying spatial variance

<table>
<thead>
<tr>
<th>Filter Sigma</th>
<th>Noise Sigma</th>
<th>Noisy MCR</th>
<th>MCR (Proposed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0.03</td>
<td>9.66</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>0.07</td>
<td>22.56</td>
<td>1.79</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>27.48</td>
<td>3.06</td>
</tr>
<tr>
<td></td>
<td>0.03</td>
<td>9.66</td>
<td>1.11</td>
</tr>
<tr>
<td>5</td>
<td>0.07</td>
<td>22.56</td>
<td>3.81</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>27.48</td>
<td>8.04</td>
</tr>
<tr>
<td></td>
<td>0.03</td>
<td>9.66</td>
<td>2.56</td>
</tr>
<tr>
<td>9</td>
<td>0.07</td>
<td>22.56</td>
<td>13.6</td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>27.48</td>
<td>21.59</td>
</tr>
</tbody>
</table>

(A) with a synthetic image for varying levels of noise and (B) with real world colour images. All the methods were run on a PC with Intel Core 2 Duo CPU of 2 GHz with 2 GB of RAM. For synthetic image, in order to quantitatively compare the results, misclassification ratio (MCR) has been used. The definition of MCR is provided in Section 1.4.1. In the experiments, all the methods have been initialized with K-means.

3.4.1 Segmentation of Synthetic Image

The algorithms were compared with a number of synthetic images with varying level of noise. In this work, results are shown with a single synthetic image for three levels of noise and effect of changing the spatial standard deviation value (sigma) of the Bilateral filter. The synthetic image has been corrupted with Gaussian noise with zero mean and varying variance value. One set of result is shown in Fig. 3.1 for a single noise level (0 mean, 0.1 variance) and for a constant spatial sigma 3. For
Table 3.2 – Comparison of performance of the proposed image segmentation method with other methods for real-world colour images, in terms of PRI

<table>
<thead>
<tr>
<th>Images</th>
<th>FCM</th>
<th>GMM</th>
<th>SMM</th>
<th>SVFMM</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mountains</td>
<td>0.889</td>
<td>0.888</td>
<td>0.889</td>
<td>0.887</td>
<td>0.891</td>
</tr>
<tr>
<td>Horse</td>
<td>0.750</td>
<td>0.782</td>
<td>0.810</td>
<td>0.777</td>
<td>0.818</td>
</tr>
<tr>
<td>Lamb</td>
<td>0.580</td>
<td>0.844</td>
<td>0.750</td>
<td>0.785</td>
<td>0.856</td>
</tr>
<tr>
<td>Bird</td>
<td>0.732</td>
<td>0.738</td>
<td>0.797</td>
<td>0.733</td>
<td>0.808</td>
</tr>
</tbody>
</table>

varying level of noise, the MCR values for the proposed method are compared for varying spatial sigma values in Table 3.1.

From Fig. 3.1, it is visible that the performance of the proposed method is less affected by the noise. The parameters of the filter also controls the robustness of the method against varying noise level. The change in performance due to change in parameters and change in noise level can be observed from Table 3.1.

Finally, Fig. 3.2 demonstrates the effectiveness of Bilateral filter over other commonly used low pass filters. The original image is corrupted with a Gaussian noise of 0.03 variance. The outputs from Median, averaging and Gaussian filter show that output segments do not overlap properly with the ground-truth, while the Bilateral filter produces accurate output very close to the ground-truth.

3.4.2 Segmentation of Real World Colour Images

In this section, four real world colour images are used from Berkeley dataset for comparing different methods. As a metric for comparison, Probabilistic Rand Index (PRI) has been used. A discussion on PRI has already been provided in Section 1.4.1. The images are shown in Fig. 3.3 and the quantitative results are provided in Table 3.2.

From the figures, the effect of noise is noticeable. The figures 3.3(i), 3.3(k), 3.3(p),
3.3(q), 3.3(v) and 3.3(w) show how affected the segmentations are with the noisy pixels. The proposed method, on the other hand, is quite robust to this noise level and successfully segment the images into separate regions. The quantitative results are shown in Table 3.2. As can be seen, the proposed method has the highest PRI values for the segmented images.

Finally, a global benchmarking has been done using the region benchmarking software available as part of the BSDS500. The results are tabulated in Table 3.3. The BSDS500 benchmark demonstrates the results for some of the leading methods in literature including gPb-owt-ucm [183], Mean-Shift [51], NCuts [84] and Graph-

### Table 3.3 – Region benchmarking with the best image segmentation methods on BSDS500

<table>
<thead>
<tr>
<th>Methods</th>
<th>PRI</th>
<th>VoI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ODS</td>
<td>OIS</td>
</tr>
<tr>
<td>Human</td>
<td>0.88</td>
<td>0.88</td>
</tr>
<tr>
<td>gPb-owt-ucm</td>
<td>0.83</td>
<td>0.86</td>
</tr>
<tr>
<td>Mean-Shift</td>
<td>0.79</td>
<td>0.81</td>
</tr>
<tr>
<td>MRF-Bilateral (Proposed)</td>
<td><strong>0.764</strong></td>
<td><strong>0.791</strong></td>
</tr>
<tr>
<td>MRF-Averaging</td>
<td>0.724</td>
<td>0.755</td>
</tr>
<tr>
<td>MRF-Gaussian</td>
<td>0.733</td>
<td>0.733</td>
</tr>
<tr>
<td>GMM</td>
<td>0.756</td>
<td>0.756</td>
</tr>
<tr>
<td>SMM</td>
<td>0.744</td>
<td>0.744</td>
</tr>
<tr>
<td>SVFMM</td>
<td>0.735</td>
<td>0.735</td>
</tr>
<tr>
<td>FCM</td>
<td>0.728</td>
<td>0.728</td>
</tr>
<tr>
<td>NCuts</td>
<td>0.746</td>
<td>0.764</td>
</tr>
<tr>
<td>GraphCuts</td>
<td>0.725</td>
<td>0.764</td>
</tr>
</tbody>
</table>
Cuts [85]. We have also listed results for GMM, SMM, SVFMM, FCM along with results of the MRF based framework with other low pass filters. According to the benchmarking algorithm [184], since some of the methods produce hierarchical region trees, obtaining a single segmentation as output involves a choice of scale. Two cases are considered: 1) a fixed threshold for all images considered, calibrated to provide optimal performance on the training set: Optimal Dataset Scale (ODS) and 2) the optimal threshold is selected by an oracle on a per-image basis: Optimal Image Scale (OIS). Obviously, OIS is per-image basis and provides better segmentation. Hence, the results are improved for OIS. As can be seen, the proposed method provides better results compared to a number of leading methodologies.

3.5 Summary

In this Chapter, an enhanced GMM has been presented for image segmentation. The model uses simple bilateral filtering based MRF to include spatial relationship among neighbouring pixels. Also, it has been kept fairly easy to manipulate the parameters of the technique and use the EM algorithm to compute the optimum values for the parameters of the mixture model.
Figure 3.1 – Synthetic image segmentation: (a) original image, (b) image corrupted with noise, (c) K-means, (d) GMM, (e) SMM, (f) SVFMM, (g) FCM and (h) proposed
Figure 3.2 – Comparison with other low pass filters: (a) Original image, (b) corrupted with Gaussian noise (0 mean, 0.03 variance), (c) Median filter output, (d) averaging filter output, (e) Gaussian filter output, (f) Bilateral filter output
Figure 3.3 – Colour image segmentation: (first row) - original image, (second row) - FCM, (third row) - GMM, (fourth row) - SMM, (fifth row) - SVFMM and (sixth row) - proposed
Chapter 4

A Multiresolution based Gaussian Mixture Model for Foreground Segmentation

This work proposes a novel multiresolution based mixture model approach in regards to the demands for a simpler, real-time and accurate approach. The related works and main contributions of the current work are presented in Section 4.1. Section 4.2 is divided into a number of subsections to justify the use of multiresolution, discuss the proposed method, and extend it towards a generic multiresolution based approach. Section 4.3 provides several types of experimental results, comparisons with some of the state-of-the-arts methods and a scope for general applicability of multiresolution features for other background modeling techniques. Finally, the chapter is concluded in Section 4.4.

4.1 A Literature Review on Wavelet based Foreground Segmentation

Use of Wavelets for foreground segmentation is not new. Wavelet based change detections [108, 109] have been proposed a long time ago. In recent years, a number of Wavelet based approaches [185, 186] and a Hadamard transform based approach [187], are proposed. These approaches fall in the category of change detection based approaches. Thus, they are highly susceptible to noise and can only provide an ap-
proximate edge image of the foreground. To improve the detection by increasing the fraction of detected edges, extensions have also been proposed [109, 188]. However, the fundamental idea of edge grouping and post-processing with morphological image filling operation is not a legitimate option to extract foreground as it often recovers part of background, and still suffers from high susceptibility to noise.

In recent years, Wavelet, Hadamard and Walsh transforms have been used for modeling the background [189, 190, 191, 192, 193, 194, 134]. However, [189],[190],[191] and [192] provide too few examples on dynamic backgrounds, and perform morphological operation on the outputs. Mendizabal et. al. [193] proposed a region-based GMM approach with Wavelets. The background consists of only one mode and the method is tested only on a few datasets and has not been compared with other state-of-the-arts approaches. Jalal et. al. [194] propose a background modeling framework using complex Wavelets. The method has a complex procedure and involves postprocessing. Sigma-delta has been improved using multiresolution in [134]. However, the quality is limited by the performance standard of Sigma-delta.

Keeping in view the above discussion, this work proposes a novel mixture model based approach in regards to the demands for a simpler, real-time and accurate approach. The model uses multiresolution coefficients for modeling the background instead of raw image data. This implicitly incorporates the spatial relationship between pixels in the mixture model without noticeable increase in complexity. The results are also compared with several approaches on a number of publicly available databases for verification.

4.2 Proposed Method

The proposed method is discussed in detail in this section. The method can be broadly classified into two distinct subsections - Wavelet based Gaussian Mixture Model (WavGMM) and generic Multiresolution based Gaussian Mixture Model (MRGMM)
due to the inherent difference of the deployment platform. The subsections are divided accordingly. However, before going into the details of the method, the readers may be interested in understanding the motivation behind using multiresolution features for clustering video sequences and the reason behind the claimed improvement that are depicted in the experimental section afterwards. Thus, this section is divided as follows. Subsection 4.2.1 explains where the current methods fall short and the motivation behind the use of multiresolution. Next, WavGMM is discussed in Subsection 4.2.2. This subsection describes both fixed and variable clustering based WavGMM, and is followed by an objective validation of the performance improvement by WavGMM, discussed in Subsection 4.2.3. Finally, an extension for other multiresolution methods has been detailed in Subsection 4.2.4.

4.2.1 Why Use Multiresolution: An Intuitive Deduction

The conventional approach discussed in Section 2.7 lacks the followings. Firstly, it assumes the intensity channels of a video frame to be independent, and considers a diagonal covariance matrix for the density functions as described in Section 2.7, to reduce the computational burden of matrix inversion. Secondly, the models do not take into account the inherent spatial relationship between neighbouring pixels. The updates of the parameter values are simply based on the new pixel value as shown in Eqs. 2.18. Models based on CRF [150, 153] take this into account, but with a major toll in computational speed and implementation complexity. Thirdly, the spatial relationship often embeds important features in images such as edges and contours. These image features are never exploited in the model based approaches to avoid huge complexity. Lastly, a manual intervention is needed to decide on the number of clusters $K$ used in the mixture model. Several researchers have proposed ideas to automatically determine the number of clusters. But, this improvement is obtained at the cost of reduced accuracy.

Based on these points, the proposed work is an effort to utilize image features and
neighbourhood information for GMM based segmentation by incorporating multiresolution features in machine learning. When an image is decomposed into multiresolution subbands, useful features in different scales can be obtained. For example, Wavelet transform decomposes an image into approximate, horizontal, vertical and diagonal subbands. While the approximate subband contains the low frequency information, the directional high frequency information is obtained by the other three subbands. This useful information has shown to provide an increased amount of detected edges in [108, 109] indicating their usefulness in change detection. The proposed method uses the subband information to model the background by GMM. Each video frame is decomposed into subbands using a multiresolution transform. These subbands are used as the temporal data for the GMM. A modified recursive filter model has been developed to construct the background in the multiresolution domain. Finally, the background is reconstructed in spatial domain using this multiresolution data. The GMM is made flexible to use different number of clusters for individual pixel processes, thus reducing the need for manual intervention in initialization.

This approach provides the following advantages. Firstly, the subbands can be considered relatively independent [195, 196, 197]. For example, the horizontal edge information is independent from the vertical edge information. This assumption is a better approximation compared to the assumption of independent colour channels. Secondly, spatial frequency decomposition inherently contains the spatial relationship information in the subbands. Thus, this relationship is somewhat exploited without an alarming increase in complexity. Actually, the subband sizes are smaller compared to the original image, leading to an increase in the speed of computation. Thus, the increase in complexity is partially compensated as well. This claim is experimentally proven in Section 4.3.3. Thirdly, as already discussed, use of multiresolution provides a scope for using inherent image features for the modeling. Lastly, in this proposed work, a variable number of clusters have been used to automate the process in comparison to a fixed number of predefined clusters for conventional GMM. The approach
of variable clustering is partially influenced from SAGMM [32], and modified to suit
the multiresolution model.

### 4.2.2 Wavelet based Gaussian Mixture Model

The Multiresolution (MR) decomposition decomposes an image into several subbands. The subbands represent different spatial frequency components of the image (e.g., Wavelets) and in some cases, the orientation information (e.g. Curvelets, Contourlets). The increased amount of information can assist in recovering the minute changes from frame to frame in a temporal sequence, thus yielding a better segmentation. Before experimentally validating the aforementioned statement, we discuss the approach of MR decomposition based GMM. In general, an MR operation \( \text{mlres}(F_t) \) on an grayscale image-frame \( F_t \) decomposes the image into a set of \( L \) subbands \( \{S_{t,l} : l \in [1, L]\} \) and the reverse operation \( \text{imlres}(\{S_{t,l}\}) \) reconstructs the image \( F_t \) from the subbands. The difference between Wavelets and the other multiresolution methods is that the Wavelets subbands are equal in size at each level of decomposition and the decomposition is dyadic in nature. Thus, for WavGMM, the following procedure is used:

1. Decompose current frame \( F_t \) of size \( P \times Q \) into 4 subbands: \( \{S_{t,l} : l \in [1, 4]\} \) each of size \( P/2 \times Q/2 \).

2. Reshape the subbands to form column vectors: \( \{v_{t,l} : l \in [1, 4]\} \) each of size \( PQ/4 \times 1 \).

3. Construct the data matrix \( X_t \) of size \( PQ/4 \times 4 \) consisting of the 4 subband vectors as the columns.

4. Use the conventional GMM approach on \( X_t \) for segmentation to find the means \( \{M_{t,k} : k \in [1, K]\} \) of \( K \) Gaussian distributions that make up the mixture model. Here, \( M \) is used instead of \( \mu \) (as in Section 2.7) in order to show that
Figure 4.1 – The comparison of means acquired from K-means, GMM and WavGMM: First row represents the K-means cluster frames, while second and third row represent the corresponding GMM and WavGMM cluster frames respectively.
the mean $\mathbf{M}$ is represented as a matrix of dimension $PQ/4 \times 4$, unlike $\mathbf{\mu}$ which is a vector of dimension $D$.

5. Estimate the background $\mathbf{B}_t$ of size $PQ/4 \times 4$ using the weighted mean approach of the conventional GMM (explained in Eq. 2.20).

6. Extract the four subband vectors: $\{\mathbf{v}_{t,l} : l \in [1, 4]\}$ each of size $PQ/4 \times 1$, from the background $\mathbf{B}_t$.

7. Reshape the subband vectors to form the subband matrices: $\{\mathbf{S}_{t,l} : l \in [1, 4]\}$ each of size $P/2 \times Q/2$.

8. Finally, reconstruct the background in the spatial domain: $\mathbf{B G}_t = \text{imlres}(\{\mathbf{S}_{t,l}\})$

The procedure is simple and provides the modification necessary to incorporate the Wavelet subbands in the segmentation process. The situation does not change when variable number of clusters is added as an extension as the modification is only part of the conventional GMM approach (Step 4 of the above procedure). The following procedure provides a brief description of the extension in the conventional GMM algorithm to incorporate variable number of clusters:

1. Keep a fixed maximum number of allowed clusters: $K_M$ for each data item in $\mathbf{X}_t$.

2. Start the process with an initial number of clusters: $K = K_0$ for each data item in $\mathbf{X}_t$.

3. Match every data value of $\mathbf{X}_t$ with the help of Eq. 2.17 to each cluster mean.

4. If match is found, the cluster parameters can be updated as in Eq. 2.18.

5. If none of the cluster means match the current data value, then a new cluster is created with the initial parameter values as mentioned in [32], provided $K < K_M$. 

78
Figure 4.2 – An objective validation of WavGMM performance: (a) Plot of the per pixel error Euclidean distance for LEC K-means-GMM and LEC K-means-WavGMM pairs, (b) Pixel process histogram overlapped with cluster means for K-means, GMM and WavGMM for pixel (70, 60), (c) and (d) represent the 3D mesh plots of error Euclidean distance for LEC K-means-GMM and LEC K-means-WavGMM pairs over the image frame, respectively.

6. If $K = K_M$, the cluster with the smallest weight is replaced with a new cluster initialized as in previous step.

The value of $K_0$ is taken as 1 for Wavelet based Gaussian Mixture Model with Variable Clustering (WavGMM_VC). As will be clear from the experiments section, variable number of clusters reduces the effectiveness of WavGMM to some extent, but the use of Wavelets provides a certain trade-off.
4.2.3 An Objective Validation

After the discussion on the motivation for using multiresolution and method to incorporate the subbands in GMM, the next question that comes to mind: does this incorporation really improve the results? To verify the same, an interesting validation method has been developed. From the discussion on the pixel processes, it is evident that, if the complete pixel process is available before the clustering operation, there would be no need for a recursive formulation. A simple K-means clustering approach can be used to accurately find the cluster centers, which represent the background and foreground. Thus, it would be an interesting experiment to compare the cluster centers acquired by a recursive GMM algorithm to those acquired using K-means.

As the recursive filter methodology was originally developed based on the K-means approach [4], this comparison can directly demonstrate the decline in accuracy due to the error in cluster center locations produced by recursive updates. Thus, a procedure is used to test the deviation of the errors when clusters from K-means are compared with the clusters from GMM and WavGMM, respectively and also, to compare these two errors.

For the experiment, the “viptraffic” video sequence of 120 frames (each frame of size 120 × 160) is used. After reshaping each frame as a vector of \( D \) elements (\( D = 120 \times 160 = 19200 \)), the data size for the video becomes \( N \times D \) where, \( N = 120 \). Assuming relative independence between pixels, K-means clustering is used to cluster the data in \( K \) clusters. \( K = 3 \) has been chosen to simplify the clustering and comparison operations, and will be clarified shortly. Similarly, GMM and WavGMM are also used to recursively cluster the data until the last frame to acquire the final 3 clusters. For all pixel processes, the corresponding clusters (represented by the respective means) are estimated and reshaped again to original size 120 × 160 for visual inspection. Thus, corresponding to the clusters obtained using K-means, GMM and WavGMM respectively, 9 images are formed and presented in Fig. 4.1. The figure raises a question - how to understand which cluster from GMM or WavGMM
correspond to a cluster from K-means? Unfortunately, due to the randomness of the K-means algorithm, there is no straightforward way to find the correspondence. Hence, a “least error correspondence” (LEC) method has been developed and discussed next.

The cluster values acquired using GMM or WavGMM deviate from the K-means cluster values. The deviation can be calculated by simple Euclidean distance measure. The LEC method is developed to find this deviation and the cluster correspondence between K-means and GMM or WavGMM. The distance measure $DM$ is defined as follows:

$$DM = \sum_{i=1}^{D} \left( \sum_{j=1}^{K} (kd_{i,j} - rd_{i,j})^2 \right), \quad (4.1)$$

where, $kd_{i,j}$ and $rd_{i,j}$ represent the K-means and recursive filtering based cluster value for $i^{th}$ pixel process and $j^{th}$ cluster, respectively. The number of clusters being 3, there can be 6 combinations of correspondence between K-means clusters and recursive filter based clusters. This is why, there can be 6 DM values each for K-means-GMM comparison and K-means-WavGMM comparison. Only 3 clusters were used to reduce the number of combinations.

**Proposition 1** Least Error Correspondence (LEC): If there are two clustering methods for clustering a data set of size $N \times D$ into $K$ clusters, there can be $K!$ ways the clusters from the second method can correspond to the first method. Out of these $K!$ combinations, the correct corresponding clusters are the ones for which $DM$ has the least value. The correct corresponding pair is termed “LEC pair”.

The proposition yields sufficiently accurate results provided the clustering methods use similar conditioning for clustering and can successfully cluster the data set. Using the LEC method, out of the 6 combinations, the corresponding clusters for K-means-GMM and K-means-WavGMM are found and shown in Fig. 4.1. It is clear from the figure that, the clusters of WavGMM are very close to that of K-means, though two of the clusters of WavGMM almost overlap with each other. For a better estimation,
we plot the per pixel error Euclidean distance for the LEC pairs, in Fig. 4.2(a) which shows that the error values for most of the pixels are lower for K-means-WavGMM compared to K-means-GMM. That means, WavGMM clusters are closer to the K-means clusters compared to the GMM clusters. Fig. 4.2(b) shows the pixel process of a sample pixel (70,60) which also verifies the previous figures. The WavGMM means are closer to those of K-means, and as before, two of the WavGMM means overlap with each other. GMM means are scattered farther.

Finally, we plotted the 3D mesh of the DM values in the shape of the original video frames, for the LEC K-means-GMM pair and K-means-WavGMM pair in Fig. 4.2(c) and (d) respectively. This brings forth an interesting fact. Before going into the details, it is important to know that the floor of the 3D meshes correspond to the figures in Fig. 4.1. Looking closely, it is evident that the high error levels for K-means-GMM pair are scattered all over the image surface while, it is only concentrated on the highway for K-means-WavGMM. The grass on the sideways have relatively low values of DM. This is because, the grass corresponds to relatively static background and the highway is randomly changed between background and foreground. Thus, WavGMM deviated from the correct means only in the dynamic areas whereas, GMM deviated irrespective of type of areas. WavGMM has been able to lower the values of DM significantly compared to GMM, indicating a better performance.

4.2.4 Extension for Other Multiresolution Methods

Multiresolution methods that also bring out the orientation information within the images, are hard to apply in GMM based foreground segmentation due to the unequal sizes of the subbands. Thus, instead of combining all the subbands as columns of a single data matrix $X_t$, each subband is treated as a separate data vector and clustered into $K$ clusters using GMM. This signifies that each subband has its own set of mixture models for each data value with distinct set of parameters. The modification is described in the following algorithm:
• Decompose current frame $F_t$ of size $P \times Q$ into $L$ subbands: $\{S_{t,l} : l \in [1, L]\} = \text{mlres}(F_t)$ with sizes $P_l \times Q_l$ where $l \in [1, L]$.

• For each subband $S_{t,l}$, do the followings:

  1. Reshape the subband $S_{t,l}$ to form the data vector $x_{t,l}$ of size $P_lQ_l \times 1$.

  2. Use the conventional GMM approach on $x_{t,l}$ for segmentation to find the means $\{\mu_{t,l,k} : k \in [1, K]\}$ of $K$ Gaussian distributions that make up the mixture model.

  3. Estimate the background vector $b_{t,l}$ of size $P_lQ_l \times 1$ using the weighted mean approach of the conventional GMM, in $n^{th}$ subband domain.

  4. Reshape the subband vectors $b_{t,l}$ to form the background subband matrices $B_{t,l}$ of size $P_l \times Q_l$.

• Finally, use the background subbands to reconstruct the background in the spatial domain: $BG_t = \text{imlres}(\{B_{t,l}\})$

Though the repeated decomposition and reconstruction at each frame would slow down the execution, the processing of subbands smaller than the original image also increase the execution speed and provide a balancing effect as discussed in next section. Experimentation has been done to compare the execution speeds of the methods to show the effect of the aforesaid modifications. Also worth mentioning that, Wavelets can be applied through this framework. But, separately treating different Wavelet subbands of equal size would meaninglessly increase the execution load. Thus, this part is not covered in the experiments.

The general framework is termed as MRGMM. For the comparison, two well known multiresolution methods are used other than Wavelets: Curvelets [198] and Contourlets [197] with 3 levels and 16 orientations of decomposition. Interested readers are encouraged to go through the references given for these multiresolution meth-

83
ods to know more about them. To simplify the operations and distinctly identify them, a flowchart representation of the procedure is presented in Fig. 4.3.

### 4.2.5 Complexity Analysis

The complexity analysis can be easily done by breaking up the two major parts of the algorithm: 1) MR-decomposition and reconstruction, 2) processing of GMM. The part of MR-decomposition and reconstruction is well-explored and have been discussed in many literatures. The approximate computational complexity of 2D Discrete Wavelet Transform (DWT) is $O(N \log(N))$, where, $N$ represents the number of data points, or in the case of image, the number of pixels. In case of more complex MR like Contourlets and Curvelets, the complexity goes up. However, analysis of any higher range is not required at this moment. The reason is explained next.

The second part using GMM requires more explanation. The three main variables controlling the complexity are: 1) the number of pixels $N$, 2) the number of clusters $K$ less or equal to $K_M$, and 3) the number of subbands $L$ after MR-decomposition. Considering the general framework, for each subband $l \in [1, L]$, the conventional GMM algorithm consists of the following steps:

1. For each pixel, find the matching cluster using Eq.2.17. The approximate complexity is: $K_M N$.

2. For each pixel, update the matching cluster. This step requires no iteration over clusters and has a complexity: $O(N)$.

3. For each pixel, sort the clusters according to the ratio of $\pi/\sigma$ resulting in the highest complexity level of $K_M^2 N$ or more efficiently, $K_M \log(K_M) N$.

4. For each pixel, identify the $B$ clusters representing the background. This part also leads to a complexity of $K_M N$. 

84
Figure 4.3 – A graphical representation of WavGMM and MRGMM
Based on the above analysis, for each subband, the maximum complexity arise in the third step: $K_M^2N$ (deliberately using the less efficient sorting) resulting in a total complexity of $LK_M^2N$. However, considering the facts $K_M \ll N$ and $L \ll N$, the complexity is $O(N)$. Combining the complexity of the two major parts, the complexity of the algorithm is $O(N \log(N)) + O(N) \approx O(N \log(N))$. Thus, the complexity of the algorithm is mainly controlled by the MR-decomposition and reconstruction step. This is the main reason for not going in details of the computational complexity for more complex MR methods.

In practice, the decomposition yields lower size of subbands. It increases the execution speed of the GMM by a large amount partly compensating the MR-decomposition and reconstruction part. In case of WavGMM and WavGMM_VC, this actually increases the execution speed as demonstrated in Section 4.3.3. However, for Contourlet based Gaussian Mixture Model (ContGMM) and Curvelet based Gaussian Mixture Model (CurveGMM), the speed somewhat reduces.

### 4.3 Experimental Results

The experiments section has been divided in several subsections to demonstrate the propositions and their advantages. The experimentations have been carried out on the CAVIAR datasets, the CMS sequence with ground-truth [33], the SZTAKI and ATON surveillance benchmark set [15, 16, 17] with ground-truth, the BMC datasets [34], the CDW datasets [12] as well as the “Car” sequence from DynTex dynamic textures datasets [14] and the “Postman” sequence (not part of any database), to demonstrate the methods in specific situations. CAVIAR and Postman datasets do not have binary foreground ground-truths. Thus, quantitative evaluation has been conducted on CMS, SZTAKI, Car, BMC and CDW datasets. Due to limited space, only a number of images from some of the datasets are shown. The databases used in the experiments are already described in Table 1.1. Each row provides one database with
Figure 4.4 – Qualitative evaluations on CAVIAR video sequence: (a) and (j) represent Fight_Runaway frames 125 and 180 respectively; (b)-(i) and (k)-(r) represent the outputs for frame 125 and 180 respectively: (b),(k) GMM, (c),(l) EGMM, (d),(m) CRFGMM, (e),(n) SAGMM, (f),(o) WavGMM, (g),(p) WavGMM_VC, (h),(q) ContGMM and (i),(r) CurveGMM
its specialty key (defined in Section 1.2) and a brief description.

Comparative studies have been carried out with some well-known methods like conventional GMM, EGMM proposed by Lee et al. [147], CRFGMM [150] and also with SAGMM [32] which is one of the recently developed methods. However, the shadow removal part of the SAGMM is not used in the implementation as it is beyond the scope of this work.

As mentioned in Section 4.2.4, two MR methods used to demonstrate the potential of MRGMM. An important difference between MRGMM and the compared methods is that MRGMM works with grayscale frames because MR methods use gray scale information. But, any colour video sequence can be converted to gray scale sequence and used for segmentation. Thus, this limitation does not affect the segmentation process. Every method compared, has been coded in MATLAB and run on a desktop with 3 GHz AMD Phenom II X6 Processor. Except for the comparison of execution speed and for CRFGMM which uses 3 clusters, a maximum of 5 clusters are used for all experiments.

It is also important to mention that the experiments do not consist of any pre-processing/post-processing filtration or morphological operations for noise removal. The effects of filtering or morphological operations are of different extent for different methods. Thus, it would not be possible to use same platform for comparison any more. It is also worth mentioning that morphological operations done after foreground extraction do not necessarily yield correct segmentation as already pointed out in Section 1.2; sometimes background also becomes part of the foreground due to overfilling. Thus, only the outputs of segmentation methods are demonstrated in the experiments.

The experimental results are divided into subsections based on the types of experiments performed. Subsection 4.3.1 consists of a qualitative comparison of the above mentioned methods. Quantitative evaluations are provided in Subsection 4.3.2. An execution speed comparison has been conducted in Subsection 4.3.3. Finally, the gen-
4.3.1 Experiment I: Qualitative Analysis

For qualitative analysis, a number of frames are shown from different video sequences to demonstrate the performance of the proposed methods in different situations. The outputs from all the methods are displayed together for comparison. Fig. 4.4(a)...
and (j) show two of the original video frames (numbered 125 and 180) from the Fight_Runaway sequence from CAVIAR datasets. The current sequence starts with a man standing in the middle of the frame while a second man standing a little to his top left. After a considerable amount of time, the first person moves towards the bottom of the frame and slowly gets out of the frame. After some time, the second person starts moving towards the left of the frame while the first person comes back into the frame (frame 125). At the end, the first person reaches his original position while the second person moves a considerable distance (frame 180). The outputs for the frames are shown in Fig. 4.4. Due to the long motionlessness of the first person, a “ghost” is left on the output for frame 125. Looking carefully at the results, it is clear that the output from EGMM (Fig. 4.4(c)) and SAGMM (Fig. 4.4(e)) suffer considerably from this ghost effect while the others yield quite better results. There is a ghost for the second person also but it is expected because the second person has just started his movement. The effect of noise are low for CRFGMM (Fig. 4.4(d)) and SAGMM (Fig. 4.4(e)) as well as for WavGMM_VC, ContGMM and CurveGMM (Fig. 4.4(g),(h) and (i) respectively). But for CRFGMM, the detection rate is low while SAGMM has been highly affected by the ghost effect. The resistance to noise and reduction in ghost effect for the proposed methods are prominent from frame 180. From the Fig. 4.4(k)-(r), it is clear that most of the outputs contain no ghosts. The outputs of the last row are qualitatively much better compared to the 3rd row results. The effect of noise is very low for WavGMM, WavGMM_VC, ContGMM and CurveGMM (Fig. 4.4(o),(p),(q) and (r) respectively). This is also because the noise and approximate parts of the image have been divided into separate bands and clustered independently. Thus, if the approximate part has a larger effect on the clusters, the relatively low noisy part would cancel out rendering the methods more robust against noisy outliers. Also, as can be seen from the figure, the proposed methods do not have any trouble detecting multiple motions. It is worth mentioning that, as WavGMM_VC is partly related to the SAGMM method, the readers would
A radially moving object moves parallel to the camera axis towards or far from the camera. Such a kind of motion is hard to detect for a background suppressing method because the moving object would always occupy some part of the background throughout its movement. These occupied pixels may be similar in intensity value and easy to get confused with background. In a natural sequence, this is even harder with the background noises. For the experiment, a sequence named “Postman” of 200 frames has been used where a postman comes down from his van, radially comes towards the front of the camera, drops a parcel and radially leaves towards his van. Frame 125 and 180 captures two time frames when the postman is leaving towards his van. The capturing camera is a low quality surveillance camera supposedly put on top of the door. Thus, the output is very noisy with unwanted tree movements in the background. Fig. 4.5(a) and (j) show frame 125 and 180. To demonstrate the results properly, detected foregrounds are shown for both frames respectively.

The outputs (Fig. 4.5) for frame 125 demonstrate the effects of a radial motion which occurred before. From a glance at the results in Fig. 4.5(b)-(i), it is clear that the noisy background and unwanted tree movements have least affected the proposed
Figure 4.7 – Qualitative evaluations on SZTAKI video sequences: (a) and (k) represent two frames from Seam and Senoon sequences, respectively. (b) and (l) provide the ground-truth segmentation; corresponding outputs are: (c),(m) GMM, (d),(n) EGMM, (e),(o) CRFGMM, (f),(p) SAGMM, (g),(q) WavGMM, (h),(r) WavGMM_VC, (i),(s) ContGMM and (j),(t) CurveGMM
Table 4.1 – Quantitative evaluations of MRGMM on datasets: CMS, SZTAKI

<table>
<thead>
<tr>
<th>Methods</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FPR</td>
<td>ACC</td>
<td>JC</td>
<td>MCC</td>
<td>FPR</td>
<td>ACC</td>
<td>JC</td>
<td>MCC</td>
<td>FPR</td>
<td>ACC</td>
<td>JC</td>
<td>MCC</td>
</tr>
<tr>
<td>GMM</td>
<td>12.153</td>
<td>87.953</td>
<td>10.657</td>
<td>74.877</td>
<td>22.682</td>
<td>79.505</td>
<td>41.151</td>
<td>53.209</td>
<td>5.291</td>
<td>94.554</td>
<td>19.912</td>
<td>70.971</td>
</tr>
<tr>
<td>EGMM</td>
<td>9.822</td>
<td>89.938</td>
<td>11.643</td>
<td>75.439</td>
<td>25.385</td>
<td>76.960</td>
<td>38.382</td>
<td>49.774</td>
<td>6.228</td>
<td>93.312</td>
<td>18.501</td>
<td>68.445</td>
</tr>
<tr>
<td>CRFGMM</td>
<td><strong>1.637</strong></td>
<td><strong>97.806</strong></td>
<td><strong>18.681</strong></td>
<td><strong>80.340</strong></td>
<td>18.201</td>
<td>80.239</td>
<td>31.577</td>
<td>43.375</td>
<td>3.720</td>
<td>95.558</td>
<td>19.336</td>
<td>68.255</td>
</tr>
<tr>
<td>SAGMM</td>
<td>5.948</td>
<td>93.840</td>
<td>15.956</td>
<td>79.833</td>
<td>8.018</td>
<td>91.426</td>
<td>58.673</td>
<td>69.792</td>
<td>1.544</td>
<td>97.796</td>
<td>27.688</td>
<td>77.526</td>
</tr>
<tr>
<td>WavGMM</td>
<td>8.450</td>
<td>91.264</td>
<td>15.371</td>
<td>79.389</td>
<td>18.214</td>
<td>83.063</td>
<td>45.915</td>
<td>57.695</td>
<td>3.591</td>
<td>95.812</td>
<td>26.873</td>
<td>77.243</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>3.102</td>
<td>96.382</td>
<td>18.116</td>
<td>81.598</td>
<td><strong>4.525</strong></td>
<td><strong>93.443</strong></td>
<td><strong>62.667</strong></td>
<td><strong>73.066</strong></td>
<td><strong>0.753</strong></td>
<td><strong>98.257</strong></td>
<td><strong>28.992</strong></td>
<td><strong>78.564</strong></td>
</tr>
<tr>
<td>ContGMM</td>
<td>3.059</td>
<td>96.498</td>
<td>19.328</td>
<td>82.785</td>
<td>9.954</td>
<td>90.077</td>
<td>54.152</td>
<td>66.125</td>
<td>3.807</td>
<td>95.759</td>
<td>20.325</td>
<td>70.803</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>2.840</td>
<td>96.713</td>
<td><strong>19.982</strong></td>
<td><strong>83.361</strong></td>
<td>9.699</td>
<td>90.293</td>
<td>54.736</td>
<td>66.632</td>
<td>3.678</td>
<td>95.880</td>
<td>20.663</td>
<td>71.127</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FPR</td>
<td>ACC</td>
<td>JC</td>
<td>MCC</td>
<td>FPR</td>
<td>ACC</td>
<td>JC</td>
<td>MCC</td>
<td>FPR</td>
<td>ACC</td>
<td>JC</td>
<td>MCC</td>
</tr>
<tr>
<td>WavGMM</td>
<td>2.872</td>
<td>96.818</td>
<td>36.130</td>
<td>56.584</td>
<td>2.304</td>
<td>96.589</td>
<td>27.038</td>
<td>62.923</td>
<td>4.742</td>
<td>93.614</td>
<td>30.597</td>
<td>44.166</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td><strong>2.384</strong></td>
<td><strong>97.074</strong></td>
<td><strong>34.994</strong></td>
<td><strong>54.214</strong></td>
<td><strong>1.860</strong></td>
<td><strong>96.761</strong></td>
<td><strong>26.560</strong></td>
<td><strong>61.161</strong></td>
<td><strong>4.575</strong></td>
<td><strong>93.544</strong></td>
<td><strong>28.279</strong></td>
<td><strong>40.705</strong></td>
</tr>
<tr>
<td>ContGMM</td>
<td>2.710</td>
<td>96.925</td>
<td>36.411</td>
<td>56.301</td>
<td>2.165</td>
<td>96.620</td>
<td>26.680</td>
<td>61.831</td>
<td>5.428</td>
<td>93.072</td>
<td>29.985</td>
<td>43.991</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>2.598</td>
<td>97.037</td>
<td><strong>37.724</strong></td>
<td><strong>57.532</strong></td>
<td>2.024</td>
<td>96.743</td>
<td><strong>27.994</strong></td>
<td><strong>63.112</strong></td>
<td>5.134</td>
<td>93.329</td>
<td><strong>30.636</strong></td>
<td><strong>44.686</strong></td>
</tr>
</tbody>
</table>
methods. Even as WavGMM_VC produce similar results to SAGMM, the output is comparatively less noisy. The outputs for frame 180 are shown in Fig. 4.5 (k)-(r). As before, the results are least affected by background motions and noises, for proposed methods. Also, the detection results are very accurate for ContGMM and CurveGMM.

Finally, we provide a few frames from the CMS video sequence, and the SZ-TAKI datasets. As these datasets contain ground-truth, they are mainly used in the quantitative analysis section. The outputs of CMS contain noise due to the continuous camera movements. A frame with outputs and ground-truth is shown in Fig. 4.6. From the figure, it is visible that GMM and EGMM cannot suppress the noise. CRFGMM suppresses some noise but is affected by the slow moving foreground. SAGMM has a better output, while ContGMM and CurveGMM provide the best resistance against noise. SZTAKI and ATON surveillance benchmark set provides several types of surveillance videos that incorporate the challenges of background noise, slow foreground, along with a high amount of illumination variation. The surveillance dataset consists of five video sequences. Two frames from SZTAKI Seam and Senoon sequences, with outputs and ground-truth are provided in Fig. 4.7. The illumination variation and noises in the background due to motion in grass are prevalent in these sequences. These noises severely affect the outputs from GMM, EGMM and CRFGMM. SAGMM provides comparatively better output, but still suffers from the noise. The proposed methods demonstrate resistance against the noises and illumination variation. Specifically, ContGMM and CurveGMM outputs are close to the ground-truth. Again, the quantitative results are provided in Subsection 4.3.2.

4.3.2 Experiment II: Quantitative Analysis

Qualitative results are very informative and easy to visualize. But, often they cannot provide the scope to distinguish subtle differences between outputs. Also, sometimes the qualitative results may seem better but are not. A quantitative analysis is more
Figure 4.8 – Qualitative evaluations on Car video sequence: (a)-(i) and (j)-(r) represent the detected foreground and background for frame 100, respectively: (a) Ground-truth, (j) Original video frame, (b),(k) GMM, (c),(l) EGMM, (d),(m) CRFGMM, (e),(n) SAGMM, (f),(o) WavGMM, (g),(p) WavGMM_VC, (h),(q) ContGMM and (i),(r) CurveGMM
authentic, leaves less room for confusion and strengthens the claims. But, a quanti-
titative analysis for foreground segmentation needs the ground-truth segmentation
results for the video sequence. Here, we provide the test results on CMS, SZTAKI,
BMC and CDW datasets. Also, we use the 692 frame long video sequence “Car” from
Dyntex. The sequence is too congested for the GMM model to successfully construct
the background. A manual binary ground-truth data for frames numbered 11 to 100,
has been carefully constructed to verify the performance of the methods compared.
Comparisons have been carried out using ACC, JC and MCC as they are considered
to be among the best measures [32]. FPR is also used for average results to highlight
the misclassification. The definitions for these measures are provided in Section 1.4.2.

CMS contain 500 frames with ground-truth. Its cumbersome to show the met-
rics for each frame. Also, SZTAKI sequences do not have ground-truth for each
frame. Thus, for both databases, we provide an average of the metrics over all
the frames containing ground-truth. Table 4.1 provides the metric values for all
datasets. With reference to the table, for CMS, CRFGMM provides least misclassi-
ification (FPR) and best ACC while CurveGMM is best for JC and MCC. Also, the
ACC values for CurveGMM, ContGMM and WavGMM_VC are very close to that of
CRFGMM. Regarding the SZTAKI sequences, we find that the proposed methods
consistently provide better results compared to the other methods. The results show
that WavGMM_VC provide better results in cases of illumination variations and noisy
backgrounds.

For the “Car” sequence, the average values of ACC, JC and MCC are provided in
Table 4.2. The rows are sorted according to the descending order of the metric values.
As evident from the table, the metric values consistently follow the same pattern. A
sample frame (frame 100) from the video sequence is also shown with corresponding
foreground and background output in Fig.4.8. There exist some “ghost” of previous
cars in some of the results. As the sequence is very congested, the background road
cannot be well constructed due to the high amount of foreground present. It can
Table 4.2 – Average metric values of MRGMM and other methods for the Car sequence

<table>
<thead>
<tr>
<th>Method</th>
<th>ACC</th>
<th>JC</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>WavGMM</td>
<td>74.186</td>
<td>56.569</td>
<td>56.991</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>73.845</td>
<td>55.237</td>
<td>55.681</td>
</tr>
<tr>
<td>ContGMM</td>
<td>73.031</td>
<td>54.292</td>
<td>54.281</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>72.652</td>
<td>53.385</td>
<td>53.240</td>
</tr>
<tr>
<td>CRFGMM</td>
<td>72.574</td>
<td>53.330</td>
<td>53.149</td>
</tr>
<tr>
<td>SAGMM</td>
<td>70.381</td>
<td>52.214</td>
<td>50.876</td>
</tr>
<tr>
<td>GMM</td>
<td>67.740</td>
<td>50.506</td>
<td>48.296</td>
</tr>
<tr>
<td>EGMM</td>
<td>60.036</td>
<td>44.164</td>
<td>38.170</td>
</tr>
</tbody>
</table>

It can be seen from the background figures that WavGMM, ContGMM and CurveGMM can still construct backgrounds (4.8(o), (q) and (r)) that are close to accurate. This results in their better performance.

As already discussed in Section 1.4.2, the authors provide evaluation procedures for BMC and CDW datasets in the respective papers ([34] and [12], respectively). We have used the softwares available in respective websites, to evaluate our results with the ground-truth provided. As both databases consist of a number of different datasets, it would be too cumbersome to show individual dataset results. Instead, for BMC, we show the average values of the metrics over the 9 real video datasets (Table 4.3). For CDW, we used the ranking procedure described in [12], and provide the average ranking for the “types of datasets”: Baseline (BL), Camera Jitter (CJ), Dynamic Background (DBG), Intermittent Object (IO), Shadow (SH), Thermal (TH) and an Overall score (OL) (Table 4.4). Each of these “types” contain 4-6 datasets. Finally, we provide the average ranks across datasets, of the methods for both databases using the ranking procedure for CDW (Table 4.5). As can be seen from Table 4.5, CurveGMM consistently provide the best results while WavGMM is
Table 4.3 – BMC: Average metric values of MRGMM and other methods over all datasets

<table>
<thead>
<tr>
<th>Method</th>
<th>RC</th>
<th>PR</th>
<th>F</th>
<th>PSNR</th>
<th>D</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>0.751</td>
<td>0.583</td>
<td>0.661</td>
<td>18.996</td>
<td>0.048</td>
<td>0.594</td>
</tr>
<tr>
<td>EGMM</td>
<td>0.759</td>
<td>0.604</td>
<td>0.676</td>
<td>22.542</td>
<td>0.035</td>
<td>0.599</td>
</tr>
<tr>
<td>CRFGMM</td>
<td>0.715</td>
<td>0.607</td>
<td>0.653</td>
<td>24.226</td>
<td>0.027</td>
<td>0.707</td>
</tr>
<tr>
<td>SAGMM</td>
<td>0.741</td>
<td>0.666</td>
<td>0.702</td>
<td>27.738</td>
<td>0.028</td>
<td>0.728</td>
</tr>
<tr>
<td>WavGMM</td>
<td>0.757</td>
<td>0.672</td>
<td>0.712</td>
<td>27.160</td>
<td>0.029</td>
<td>0.759</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>0.719</td>
<td>0.689</td>
<td>0.704</td>
<td>31.023</td>
<td>0.019</td>
<td>0.868</td>
</tr>
<tr>
<td>ContGMM</td>
<td>0.759</td>
<td>0.662</td>
<td>0.707</td>
<td>26.940</td>
<td>0.029</td>
<td>0.707</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>0.761</td>
<td>0.668</td>
<td>0.711</td>
<td>27.461</td>
<td>0.028</td>
<td>0.731</td>
</tr>
</tbody>
</table>

comparable to CurveGMM for CDW while falls behind for BMC. Of course, a single method cannot provide the best results for each type of video. This is also evident from Table 4.3 and 4.4 results.

4.3.3 A Comparison on Execution Time

For execution time comparison, all methods are evaluated on the platform mentioned earlier. The “viptraffic” sequence sample video (source: MATLAB) has been used for this experiment. The video sequence (each frame of size $160 \times 120$ pixels) of 120 frames, is a relatively easy sequence and all the methods are run with 3 clusters. The execution times are tabulated in Table 4.6 sorted in ascending order.

As can be seen from the table, WavGMM and WavGMM_VC are fast compared to other methods, the reason being that the Wavelet decomposition produces subbands of half size which require less time for processing. The important point to mention here is that the implementation of the multiresolution transform largely controls the execution time. This explains the relatively large difference between the execution
### Table 4.4 – CDW: Average ranking of MRGMM and other methods for each dataset

<table>
<thead>
<tr>
<th>Datasets</th>
<th>BL</th>
<th>CJ</th>
<th>DBG</th>
<th>IO</th>
<th>SH</th>
<th>TH</th>
<th>OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>6.29</td>
<td>5.71</td>
<td>5.86</td>
<td>4.86</td>
<td>3.86</td>
<td>3.29</td>
<td>4.86</td>
</tr>
<tr>
<td>EGMM</td>
<td>5.29</td>
<td>4.71</td>
<td>5</td>
<td>3.57</td>
<td>6.29</td>
<td>7.71</td>
<td>6.14</td>
</tr>
<tr>
<td>CRFGMM</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>6.57</td>
<td>7.43</td>
<td>6.14</td>
<td>6.86</td>
</tr>
<tr>
<td>SAGMM</td>
<td>5.86</td>
<td>3</td>
<td>4.86</td>
<td>5.29</td>
<td>5.29</td>
<td>4.57</td>
<td>4.86</td>
</tr>
<tr>
<td>WavGMM</td>
<td>1</td>
<td>5.14</td>
<td>3</td>
<td>2.71</td>
<td>1.71</td>
<td>2.57</td>
<td>2.71</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>3.43</td>
<td>5.57</td>
<td>5.14</td>
<td>4.57</td>
<td>4.43</td>
<td>3.86</td>
<td>4.14</td>
</tr>
<tr>
<td>ContGMM</td>
<td>3.57</td>
<td>3.14</td>
<td>3.43</td>
<td>4.43</td>
<td>4.86</td>
<td>4.57</td>
<td>4</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>2.57</td>
<td>2.71</td>
<td>2.71</td>
<td>4</td>
<td>2.14</td>
<td>3.29</td>
<td>2.43</td>
</tr>
</tbody>
</table>

### Table 4.5 – Average ranking of MRGMM and other methods over all datasets for BMC and CDW

<table>
<thead>
<tr>
<th>Method</th>
<th>BMC</th>
<th>CDW</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>6</td>
<td>5.29</td>
</tr>
<tr>
<td>EGMM</td>
<td>4.67</td>
<td>5.43</td>
</tr>
<tr>
<td>CRFGMM</td>
<td>4.33</td>
<td>7.86</td>
</tr>
<tr>
<td>SAGMM</td>
<td>5</td>
<td>5.14</td>
</tr>
<tr>
<td>WavGMM</td>
<td>4</td>
<td>1.86</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>2.67</td>
<td>4.57</td>
</tr>
<tr>
<td>ContGMM</td>
<td>4.67</td>
<td>4.00</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>2.33</td>
<td>1.86</td>
</tr>
</tbody>
</table>
Table 4.6 – Comparison of MRGMM and other methods in terms of Computational speed

<table>
<thead>
<tr>
<th>Method</th>
<th>Total time (seconds)</th>
<th>Frames per second</th>
</tr>
</thead>
<tbody>
<tr>
<td>WavGMM</td>
<td>2.28</td>
<td>50-54</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>2.41</td>
<td>48-51</td>
</tr>
<tr>
<td>EGMM[147]</td>
<td>4.50</td>
<td>25-30</td>
</tr>
<tr>
<td>SAGMM[32]</td>
<td>5.87</td>
<td>19-22</td>
</tr>
<tr>
<td>ContGMM</td>
<td>7.59</td>
<td>14-16</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>35.90</td>
<td>3-5</td>
</tr>
<tr>
<td>CRFGMM[150]</td>
<td>41.2</td>
<td>2-4</td>
</tr>
</tbody>
</table>

speed of ContGMM and CurveGMM. Finally, the variation in frame rate is kept due to the fact that the amount of foreground has an effect on clustering complexity.

4.3.4 A Discussion on General Applicability of Multiresolution for Background Modeling

Based on the discussion in Section 4.2.3 and 4.3, an overall improvement of GMM using multiresolution features can be observed. Section 4.1 also described a number of methods on background modeling using multiresolution features. However, two questions may be raised: 1) How beneficial the integration of multiresolution to GMM can be? 2) Is there any scope of improvement, if multiresolution features are used with any other background modeling methods?

In order to answer the first question, we need to judge the scopes and scenarios where MRGMM may be applied better, compared to conventional GMM. MRGMM differs from conventional method in terms of applicability as the multiresolution mod-
ule needs to be implemented alongside the conventional GMM module. However, in practice, there are a number of Wavelet, Curvelet and Contourlet implementations available in real-time computer languages like C and C++. Use of any such a library would reduce the implementation complexity by a considerable amount. Next, in order to judge whether such an extension has any worth, we can consider the outcomes of the experiments. Without any postprocessing, considerably good results are obtainable by use of multiresolution. An application using background modeling, normally uses morphological transformations and connected component analysis as means of post-processing, and these processing methods are severely application dependent. Thus, a single postprocessing may not suffice to a collection of applications. Also, as background subtraction has its use as an intermediate processing step for a number of applications in computer vision, the accuracy and postprocessing time play important roles in the overall quality of the application. By using multiresolution with a simple increment in implementation complexity, a huge benefit can be gained in terms of reduction of postprocessing and increased accuracy. Also, the frame processing time for GMM is reduced by half, decreasing the overall processing time. Thus, the benefit of multiresolution is in three folds: 1) Increased accuracy, 2) Reduction in postprocessing, and 3) Reduction in overall execution time; the advantages can definitely cover the expense of incremental implementation complexity.

Finally, the improvements depend on the complexity and type of video sequences. Thus, the improvement may not be similar for all video sequences. But, parameters for conventional GMM are also tuned for specific applications. In similar way, multiresolution based GMM parameters such as the level of decomposition and learning rate, can also be tuned for specific applications like highway surveillance, convenience store CCTV etc.

To answer the second question, we picked two popular nonparametric methods in literature: Codebook based model (Codebook) [142], and Kernel Density Estimate (KDE) [144], and applied multiresolution based decomposition on both of
them. The OpenCV implementation of Codebook and Background Subtraction Library (BGS) implementation of KDE have been used for the experiments. The experiments are limited to Wavelet based decomposition only. The frames are decomposed into Wavelet subbands, and combined as described in Section 4.2. The subband images are used instead of the original image for Wavelet based methods. The outputs are rescaled to original scale for comparisons. For comparisons, SZTAKI and ATON datasets are used. The output DR, FPR, ACC, JC and MCC are tabulated in Tables 4.7 and 4.8 (values are rounded up to 2nd decimal places to reduce table length). As seen from the table, in general, outputs from Wavelet based methods are improved compared to the original ones. The improvements are mainly due to reduction in noises and increase in detected foregrounds.

Although straightforward generalizations like the ones proposed are easier, they may not always be beneficial for any method. Even the improvements for Codebook and KDE are not similar. WavCodebook has shown high improvements over Codebook where, WavKDE is similar in performance to KDE in some of the sequences. Thus, an extensive study on the effects of integrating multiresolution to most of the state-of-the-arts method would be a very interesting idea. However, it is an enormous effort as every method has specific type of implementation and inclusion of multiresolution may not be as straightforward. Thus, this study has been left out of this work due to its limited scope. The purpose of this section is to address the concerns related to the necessity of the proposed improvements, and to suggest a general applicability of multiresolution for background modeling that may be beneficial to future researches.

4.4 Summary

This work proposed a novel modification to Gaussian mixture model for foreground segmentation by incorporating multiresolution decomposition. The work has three
Table 4.7 – Comparison of Codebook and KDE with their Wavelet based implementations (I)

<table>
<thead>
<tr>
<th>Methods</th>
<th>DR</th>
<th>FPR</th>
<th>ACC</th>
<th>JC</th>
<th>MCC</th>
<th>DR</th>
<th>FPR</th>
<th>ACC</th>
<th>JC</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SZTAKI Highway</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Codebook</td>
<td>13.16</td>
<td>1.28</td>
<td>86.57</td>
<td>12.28</td>
<td>23.19</td>
<td>68.29</td>
<td>4.04</td>
<td>93.41</td>
<td>18.38</td>
<td>66.08</td>
</tr>
<tr>
<td>WavCodebook</td>
<td>41.40</td>
<td>1.93</td>
<td>89.88</td>
<td>36.99</td>
<td>51.61</td>
<td>79.01</td>
<td>5.00</td>
<td>93.36</td>
<td>27.49</td>
<td>71.94</td>
</tr>
<tr>
<td>KDE</td>
<td>59.46</td>
<td>1.82</td>
<td>93.17</td>
<td>53.14</td>
<td>66.41</td>
<td>64.83</td>
<td>2.09</td>
<td>95.28</td>
<td>18.11</td>
<td>66.96</td>
</tr>
<tr>
<td>WavKDE</td>
<td>96.90</td>
<td>8.79</td>
<td>92.15</td>
<td>61.34</td>
<td>73.71</td>
<td>77.92</td>
<td>4.09</td>
<td>94.29</td>
<td>25.68</td>
<td>72.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SZTAKI Laboratory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Codebook</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WavCodebook</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KDE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WavKDE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>DR</th>
<th>FPR</th>
<th>ACC</th>
<th>JC</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SZTAKI Seam</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Codebook</td>
<td>37.16</td>
<td>1.50</td>
<td>97.01</td>
<td>16.01</td>
<td>31.56</td>
</tr>
<tr>
<td>WavCodebook</td>
<td>65.48</td>
<td>1.16</td>
<td>97.92</td>
<td>33.14</td>
<td>54.45</td>
</tr>
<tr>
<td>KDE</td>
<td>75.92</td>
<td>3.66</td>
<td>95.97</td>
<td>29.01</td>
<td>48.08</td>
</tr>
<tr>
<td>WavKDE</td>
<td>97.19</td>
<td>5.43</td>
<td>94.72</td>
<td>29.27</td>
<td>52.60</td>
</tr>
</tbody>
</table>
Table 4.8 – Comparison of Codebook and KDE with their Wavelet based implementations (II)

<table>
<thead>
<tr>
<th>Methods</th>
<th>DR</th>
<th>FPR</th>
<th>ACC</th>
<th>JC</th>
<th>MCC</th>
<th>DR</th>
<th>FPR</th>
<th>ACC</th>
<th>JC</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SZTAKI Senoon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SZTAKI Sepm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Codebook</td>
<td>73.20</td>
<td>2.71</td>
<td>96.64</td>
<td>24.09</td>
<td>57.05</td>
<td>62.84</td>
<td>5.35</td>
<td>92.86</td>
<td>29.24</td>
<td>43.94</td>
</tr>
<tr>
<td>WavCodebook</td>
<td>88.33</td>
<td>3.09</td>
<td>96.85</td>
<td>29.95</td>
<td>64.89</td>
<td>79.49</td>
<td>6.63</td>
<td>92.51</td>
<td>33.84</td>
<td>51.10</td>
</tr>
<tr>
<td>KDE</td>
<td>84.75</td>
<td>3.84</td>
<td>95.61</td>
<td>22.63</td>
<td>57.34</td>
<td>68.07</td>
<td>3.98</td>
<td>94.55</td>
<td>36.45</td>
<td>51.88</td>
</tr>
<tr>
<td>WavKDE</td>
<td>97.88</td>
<td>5.29</td>
<td>94.58</td>
<td>21.78</td>
<td>59.56</td>
<td>86.06</td>
<td>10.21</td>
<td>89.54</td>
<td>28.25</td>
<td>45.99</td>
</tr>
</tbody>
</table>
distinct contributions. Firstly, Wavelet based Gaussian Mixture Model has been proposed and discussed. Secondly, incorporation of variable number of clusters in the proposed approach has also been demonstrated. Thirdly, a generic framework has been constructed to incorporate any multiresolution method into GMM. Extensive experimentations have been conducted to support the proposals. Finally, a discussion has been made on the benefits of the proposed approach, and the general applicability of multiresolution towards background modeling.
Chapter 5

Gaussian Mixture Model with Advanced Distance Measure based on Support Weights and Histogram of Gradients for Foreground Segmentation

In answer to the need for higher accuracy while keeping a relatively low implementation complexity, the proposed approach is an effort to incorporate spatial dependencies between adjacent pixels into classification process with simple changes in distance measure. The contributions of the proposed approaches are: Use of support weights and histogram of gradients to formulate an effective distance measure to evaluate cluster distances, and use of background layer to properly segment the foreground while using variable number of clusters. In order to easily designate the method, it is termed as Advanced Distance Measure based Gaussian Mixture Model (ADMGMM). It has a variation with no iteration for parameter updates, called Advanced Distance Measure based GMM with no Iterations (ADMGMM_NI) and discussed later in the chapter. Extensive experiments are carried out to validate the improvement in accuracy with both ADMGMM and ADMGMM_NI.

The rest of the chapter is divided as follows: Section 5.1 is divided into a three subsections to discuss the proposed distance measure, followed by the proposed approach and a complexity analysis. Section 5.2 provides an intuitive explanation of the algorithm followed by several types of experimental results and comparisons with some of the state-of-the-art methods. Finally, the chapter is concluded in Section 5.3.
5.1 Proposed Method

Referring to Section 2.7, conventional GMM has a number of problems:

1. it does not take into account, the spatial relationship between the neighbouring pixels. Pixels belonging to one object often resemble similar characteristics such as colour, intensity and edge orientations. This can actually help to reduce the misclassification problem.

2. the distance formulation in GMM is only based on Euclidean distance between colour vectors. This distance often fails as different value-pairs in separate channels may yield same distance. Thus, a foreground pixel may be confused with a background pixel having similar colour distances, even if they have different colours.

3. the distance thresholding in equation 2.17 is done using variance level. A busy foreground may raise the variance level and not get detected at all. This problem also persists in Sigma-Delta based techniques.

4. conventional GMM uses a threshold on the \( \pi / \sigma \) ratio to determine the background modes (Eq. 2.19). However, a dynamic background may have a lot of unwanted motions that cannot properly separate background from foreground. Eventually, the dynamic background gets detected or it increases the variance level so much that the foreground is less detected (from previous problem). Thus, a better separation of background and foreground is required.

In view of the above, ADMGMM has been developed. The method handles problems 1 and 2 using two different features to yield the distance value: Adaptive Support Weights (ASW), and HOG. Problem 3 has been handled by implicitly using variance inside the SW distance while using constant thresholds for matching. This part is more elaborated in Section 5.1.2. Finally, problem 4 is answered using a background-layer based concept used with variable number of clusters.
Before going into the details of the algorithm, a discussion on ASW and HOG for distance measurement is necessary. Hence, the current section is divided as follows: Section 5.1.1 discusses the distance measure using ASW and HOG, followed by Section 5.1.2 describing the algorithm in detail. Finally, an algorithm complexity analysis has been conducted in Section 5.1.3.

5.1.1 Distance Measure

Support weights have already been used for disparity measurements [200, 201]. The Gestalt principles of similarity and proximity are used to compute the support weights involving a neighbourhood of any pixel. The support weight between two neighbour pixels \( x \) and \( y \) is written as:

\[
sw(x,y) = kf_s(\Delta c_{xy})f_p(\Delta g_{xy}). \tag{5.1}
\]

Here, \( f_s(\cdot) \) and \( f_p(\cdot) \) represent the strength of grouping by similarity and proximity, respectively. \( \Delta c_{xy} \) and \( \Delta g_{xy} \) denote the distances in colour and spatial domain, between pixels \( x \) and \( y \). \( k \) is a proportionality constant. In colour space, \( \Delta c_{xy} \) is expressed as:

\[
\Delta c_{xy} = \sqrt{(R_x - R_y)^2 + (G_x - G_y)^2 + (B_x - B_y)^2}, \tag{5.2}
\]

where, \((R_x, G_x, B_x)\) are the three colour components of pixel \( x \). In the implementation, RGB colour-space has been used for colour videos. For gray-scale videos, equation 5.2 is transformed to simple absolute difference.

The strength of grouping by similarity is expressed as:

\[
f_s(\Delta c_{xy}) = \exp\left(-\frac{\Delta c_{xy}}{\gamma_c}\right), \tag{5.3}
\]

where, \( \gamma_c \) is a constant. Similarly, the strength of grouping by proximity has the expression:

\[
f_p(\Delta g_{xy}) = \exp\left(-\frac{\Delta g_{xy}}{\gamma_p}\right), \tag{5.4}
\]
where, $\gamma_p$ is a constant; $\Delta g_{xy}$ can be computed using simple Euclidean distance measure between $x(u_x, v_x)$ and $y(u_y, v_y)$ as follows:

$$\Delta g_{xy} = \sqrt{(u_x - u_y)^2 + (v_x - v_y)^2},$$

(5.5)

where, $(u, v)$ represent the coordinates of a pixel. Combining equations (5.3) and (5.4), equation (5.1) becomes:

$$sw(x, y) = k \exp \left( -\frac{\Delta c_{xy} + \Delta g_{xy}}{\gamma_c} \right).$$

(5.6)

The support weights have the characteristics to provide more significance to pixels closer or similar in colour to the center pixel. When comparing a pixel value to a cluster mean, erroneous matching may result if simple Euclidean distance is used. Also, pixels in similar colour region may yield a large distance due to variation in one channel. To minimize the effects of such pixels, the distance is computed between two pixels by combining the support weights in support windows around the pixels. A support window contains neighbouring pixels around a center pixel.

To get the support weight based distance between data point $x_t$ and cluster mean $\mu_{t,j}$ corresponding to the image coordinates of $x_t$, support windows around $x_t$ and $\mu_{t,j}$ are considered. For the computation, $j^{th}$ cluster means are formed into an image (examples of clusters as images are shown in Fig. 5.2). In this image, $\mu_{t,j}$ has a neighbourhood of cluster means belonging to $j^{th}$ cluster. Let, $N_{t,x}$ and $N_{t,\mu,j}$ denote the neighbourhood of the support windows (of length $N_{SW}$) around $x_t$ and $\mu_{t,j}$, respectively. The distance between $x_t$ and $\mu_{t,j}$ can be expressed in terms of support weights, as:

$$D_{SW}(x_t, \mu_{t,j}) = \sum_{(y \in N_{t,x}), (m_{t,j} \in N_{t,\mu,j})} \frac{sw(x_t, y_t)sw(\mu_{t,j}, m_{t,j})e(y_t, m_{t,j})}{\sigma_{t,y,j}},$$

(5.7)

where, $y_t$ and $m_{t,j}$ represent neighbour pixels of $x_t$ and $\mu_{t,j}$, respectively. $e(y_t, m_{t,j})$ represents the colour based distance value as in equation 5.2. $\sigma_{t,y,j}$ represents the
SD of \( j \)th cluster at position of \( y_{t,j} \), or more specifically, the SD of mean \( \mu_{t,j} \). The explanation of the division by SD is provided afterwards. \( D_{SW} \) is normalized by dividing it with \( \sqrt{3 \times 255^2} \), which is the maximum value it can take.

\( D_{SW} \) is one of the best measures for distance as described in [200]. However, it may still provide inaccurate results where the regions to be matched are of uniform textures, or less contrast. For these regions, HOG measure is more useful. HOG depends on the orientation of gradient at each pixel in a region. To compute HOG, following steps are performed:

- The image is convolved with filters \([-1, 0, 1]\) and \([-1, 0, 1]^T\) to yield the filtered edge images \( F_x \) and \( F_y \) along x and y-directions, respectively.
- The gradient norm \( F_g \) and orientation \( \Xi_g \) are found in the following way:
  \[
  F_g = \sqrt{F_x^2 + F_y^2}; \quad \Xi_g = \arctan\left(\frac{F_y}{F_x}\right). \tag{5.8}
  \]
- A histogram is formed over the support window \( N_{t,x} \) for each pixel by quantizing the orientation values in \( N_{\Xi} \) bins.

Thus, for each pixel, a histogram \( h_{t,x} \) of length \( N_{\Xi} \) is obtained. Similarly, a histogram \( h_{t,\mu,j} \) is computed for each cluster mean. The HOG distance between \( x_t \) and \( \mu_{t,j} \) is obtained as:

\[
D_{HoG}(x_t, \mu_{t,j}) = \frac{1}{N_{\Xi}} \sum_{\xi \in N_{\Xi}} (h_{t,x}(\xi) \neq h_{t,\mu,j}(\xi)). \tag{5.9}
\]

\( h_{t,x}(\xi) \) denotes the value of histogram \( h_{t,x} \) for the bin center angle \( \xi \). The distance measure \( D_{t,x,j} \) between \( x_t \) and \( j \)th cluster mean, is obtained as follows:

\[
D_{t,x,j} = \eta D_{SW}(x_t, \mu_{t,j}) + (1 - \eta) D_{HoG}(x_t, \mu_{t,j}). \tag{5.10}
\]

Here, \( \eta > 0 \) is a constant with value less than 1. As both \( D_{SW} \) and \( D_{HoG} \) are normalized, \( D_{t,x,j} \) lies between 0 and 1. \( \eta \) is generally kept above 0.5 to give more preference to ASW. More preference to \( D_{HoG} \) would increase the effect of unnecessary textures in clustering.
5.1.2 Algorithm

In order to incorporate the new distance measure, the algorithm is modified from conventional GMM. Also, a variable clustering based scheme has been used as mentioned before. A maximum of $K_M$ clusters are allowed. Initially, the algorithm begins with a single cluster i.e. $K = 1$. The mean is assigned as the first frame in the video sequence. The initial SD and weight are kept as $\sigma_{init}$ and $\pi_{init}$, respectively. The ASW and HOG for clusters are simply computed based on the first frame values. When the algorithm encounters a new frame, the new pixel value $x_t$ is compared against each of the means $\mu_{t,j}$ and the distance $D_{t,x,j}$ is computed for each cluster. Cluster $jm$ is selected with minimum distance $D_{t,x,jm}$. Considering each pixel’s minimum distance from certain cluster, a minimum distance matrix $D_t$ (of image size) is obtained. $D_t$ is filtered with a mean filter to reduce outliers in distance values. Thus, a new filtered minimum distance is found:

$$D_t = f(D_t),$$

where, $f(\cdot)$ denotes the filter operation, and $D_t$ is the filtered minimum distance matrix. Filtered distance for pixel $x_t$ at position $(u, v)$ can be represented as $D_{t,x,jm}$, which is the element of $D_t$ at position $(u, v)$.

A match is found if the distance is within a threshold $T$. In line with Eq. 2.17, mathematically it can be expressed as:

$$x_t \in \Phi(x_t | \mu_{t,jm}, \Sigma_{t,jm}) \text{ if } D_{t,x,jm} < T,$$

(5.12)

Considering the third problem described at the beginning of this section, a thresholding by variance wouldn’t always provide an expected result. Instead, the variance is accumulated in the support weight calculation as depicted in equation 5.7. This allows us to use a fixed threshold for matching. Experiments have been provided to validate the use of the fixed threshold.

For matched distributions, the parameters of the clusters are updated according to equation 2.18. Here, a change has been proposed to expedite the process of updating
parameters. The ASW and HOG values for each cluster can be computed at each time-step. But, this increases the computational burden on the algorithm. Thus, the cluster ASW and HOG values are updated along with the other parameters. The update process is described below:

\[
sw(\mu_{t+1,jm}, m_{t+1,jm}) = (1 - \alpha)sw(\mu_{t,jm}, m_{t,jm}) + \alpha sw(x_t, y_t);
\]

\[
h_{t+1,\mu,j}(\xi) = (1 - \alpha)h_{t,\mu,j}(\xi) + \alpha h_{t,x}(\xi).
\]

With this recursive update procedure, large number of support weight computations that directly affect the execution speed, can be avoided. Without the recursive procedure, the results improve but with additional drop in execution speed. The outputs using iteration (ADMGMM) and no iteration (ADMGMM_NI) are both compared in Section 5.2.3 with a comparison of execution speed in Section 5.2.4.

For the unmatched distributions, no parameter update is necessary except for the cluster weights which are reduced by a factor of \((1 - \alpha)\). If none of the distributions match the current pixel value, there are two options. If the number of clusters \(K\) has already reached maximum number of allowed clusters \(K_M\), the cluster \(jl\) with lowest weight is replaced by a new cluster with the following parameters:

\[
\mu_{t+1,jl} = x_t, \sigma_{t+1,jl} = \sigma_{init}, \pi_{t+1,jl} = \pi_{init};
\]

ASW and HOG values of the current pixel are used for the cluster as well. If \(K\) has not reached \(K_M\), a new cluster can simply be added with same initial values.

To compute the foreground, a concept of background layer is used. The clusters belonging to background can be computed using conventional method (equation 2.19). The clusters are sorted according to the descending values of \(\pi_{t+1,j}/\sigma_{t+1,j}\), followed by marking the first \(B\) distributions as part of the background while the rest of the distributions to be part of the foreground. In order to compute foreground, \(D_{t,x,jm}\) is thresholded by \(T\) to get the initial binary foreground mask. Next, the pixels, matched to a cluster not part of the first \(B\) distributions, are also added to the foreground.
Mathematically, it can be expressed as:

$$fg = \text{NOT} \ ((D_{t,x,jm} < T) \ \text{AND} \ (jm \leq B));$$

Or, $$fg = (D_{t,x,jm} \geq T) \ \text{OR} \ (jm > B).$$

To simplify the procedure, the algorithm steps are sketched in Fig. 5.1 and summarized below:

Pre-process the frame to compute its ASW and HOG:

1. Compute the ASW based distance $$D_{SW}$$: equation 5.7
2. Compute the HOG based distance $$D_{HoG}$$: equation 5.9

For each pixel $$x_t$$ in $$[1, N]$$:

1. For each cluster $$j$$ in $$[1, K]$$, compute $$D_{t,x,j}$$: equation 5.10
2. Find $$D_{t,x,jm}$$ and $$\overline{D_{t,x,jm}}$$: equation 5.11
3. Threshold distance: $$d = (D_{t,x,jm} < T)$$ equation 5.12
4. If $$d$$ equals 1, update $$jm$$th cluster: equation 2.18, 5.13; no update for unmatched clusters except for cluster weights
5. If $$d$$ equals 0, and $$K$$ equals $$K_M$$, replace cluster with lowest weight, by a cluster with initial parameter values
6. If $$d$$ equals 0, and $$K$$ is less than $$K_M$$, add a new cluster with initial parameter values
7. Sort the clusters according to the sorting criteria
8. Label the $$B$$ background clusters
9. Compute the foreground: equation 5.14
Figure 5.1 – A graphical representation of ADMGMM
Figure 5.2 – Comparison on mode selection of GMM and ADMGMM: First and second row contain the clusters for GMM and ADMGMM in decreasing order of $\pi_{t,j}/\sigma_{t,j}$, respectively. In third row, the original image is followed by the ground-truth, the results for GMM and ADMGMM, respectively.

Figure 5.3 – Qualitative evaluation of performance improvement using background layer: For each row, the columns contain the original image, the ground-truth, GMM output, output of ADMGMM without and with background layer based detection, respectively.
5.1.3 Complexity Analysis

The algorithm follows a number of steps. The complexity of each step depends on three main variables: 1) the number of pixels \( N \), 2) the number of clusters \( K \) less or equal to \( K_M \), and 3) the number of neighbours \( N_{SW} \) for each pixel. To conduct the analysis, we can refer to the algorithm steps mentioned at the end of Section 5.1.2. According to the rule, we need to analyze the complexity of each step and assign the maximum step complexity as the complexity of the algorithm. The pre-processing step 1 iterates over each pixel and its neighbours. Hence, the complexity is \( N_{SW}N \).

Pre-processing step 2 consists of an image filtering operation with complexity in \( O(N) \) and a histogram formation with complexity \( N\xi N \). Thus, the combined pre-processing complexity is \( O(N) \) considering \( N_{SW} > N\xi \) and \( N_{SW} \) remaining constant for different video frame sizes.

Next, for each pixel, we deduce the complexity. Computation of \( D_{t,x,j} \) has a complexity of \( K_MN_{SW} \). Steps 3, 4, 6 and 9 have constant complexity as they contain no iteration. Steps 2, 5, and 8 have a complexity of \( K_M \). Finally, step 7 has \( K_M^2 \) (or, more efficiently \( K_M \log(K_M) \)) complexity. As in practice, \( K_M \ll N_{SW} \), combined complexity of steps [1-9] for each pixel, is \( K_MN_{SW} \). Thus, the total complexity is \( K_MN_{SW}N \). With constant \( K_M \) and \( N_{SW} \) for any video, the complexity of algorithm is \( O(N) \). For conventional GMM, step 1 has only \( K_M \) iterations leading to a complexity of \( K_MN \). Thus, although mathematically both complexities may be represented as \( O(N) \), in practical situations, the factor of \( N_{SW} \) reduces the execution speed. Also, for ADMGMM_NI, step 4 would have an additional iteration of \( N_{SW} \) due to feature computations instead of updates. Thus, another \( N_{SW}N \) factor gets added to the complexity, and further lowers the execution rate by 50-60%. An estimation of execution speed has been provided in Section 5.2.4.
Figure 5.4 – Qualitative evaluations on CAVIAR video sequence: Two groups of images corresponding to the Fight_OneManDown sequence from CAVIAR datasets. Each group of images contain: (a) Original frame, (b) GMM output, (c) EGMM output, (d) CRFGMM output, (e) SAGMM output, and (f) ADMGMM output.
Figure 5.5 – Qualitative evaluations on CMS and ATON video sequences: Two image groups from Carnegie Mellon dataset and ATON Highway sequence, respectively. Each group of images contain: (a) Original frame, (b) GMM output, (c) EGMM output, (d) CRFGMM output, (e) SAGMM output, and (f) ADMGMM output.
5.2 Experiments

This section is divided in four major subsections. Section 5.2.1 provides an intuitive explanation of the algorithm and the enhancements proposed to solve the problems of conventional GMM. Section 5.2.2 provides qualitative results for experiments on datasets from CAVIAR database, Carnegie Mellon database, SZTAKI & Aton surveillance benchmark sets, the BMC datasets and the CDW datasets. CAVIAR datasets do not contain proper ground-truth for the detected foreground. Hence, Section 5.2.3 conducts quantitative experiments on datasets from Carnegie Mellon, SZTAKI & ATON, BMC and CDW databases. Finally, Section 5.2.4 reports a comparison on execution time.

For comparison, a number of well-known methods are used. Apart from conventional GMM, EGMM [147], CRF based GMM (CRFGMM) [150] and SAGMM [32] have been used as representatives of GMM based techniques. SAGMM is one of the recently developed methods. However, the shadow removal part of SAGMM is not used for the comparison as it is beyond the scope of ADMGMM. Several recent techniques are also considered for the quantitative analysis: VISIONSYS [123], FASOM [154], GMG [138] and Type-2 Fuzzy GMM and Markov Random Field based method with Uncertain Mean (T2FMRF_Um) and Type-2 Fuzzy GMM and Markov Random Field based method with Uncertain Variance (T2FMRF_Uv) [202]. For FASOM, GMG, T2FMRF_Um and T2FMRF_Uv, the implementations from Background Subtraction Library (BGSLIB) [199] are used. GMM, EGMM, CRFGMM, SAGMM and VISIONSYS are implemented in MATLAB using authors’ given parameter values. The codes run on a machine with 3.4 GHz Intel Core i7-3770 processor. Except for CRFGMM, which uses only 3 clusters, all the GMM based methods are run with maximum 5 clusters. No post-processing operation has been conducted on any of the outputs to keep a fair comparison. The support weight computation reduces the computational speed of the conventional GMM by a considerably high amount. But,
the ADMGMM compensates with a higher accuracy. ADMGMM_NI provides higher accuracy at the price of larger reduction in speed. Thus, we recommend the iteration in equation 5.13 as a trade-off between accuracy and execution speed. This part has been elaborated in Section 5.2.3.

The parameters for the method have been fixed for each experiment. The following values have been used: $K_M = 5; T = 0.5; \alpha = 0.01; \rho = 0.01; Th = 0.6; \eta = 0.6; N_\xi = 25; \gamma_c = \gamma_p = 15$. Choice of $K_M$ depends on the complexity of video. $K_M > 4$ generally suffice as 3-4 clusters can properly model the background while the remaining cluster(s) can model the foreground. $T$ is used to threshold the distance measure which ranges from 0 to 1. Keeping $T$ too close to the minima (0) may reject inliers with relatively high values, and keeping too close to maxima (1) may include unnecessary outliers. Hence, $T$ is chosen as 0.5 to keep equal distance from the extremas.

The rest of the constant values are based on previous literature. The neighbourhood window sizes for both ASW and HOG are kept at $5 \times 5$. Thus, $N_{SW} = 25$ which is a reasonably large neighbourhood.

### 5.2.1 An Intuitive Explanation

The proposed algorithm differs from the conventional method in a number of ways. It includes a different distance measure to properly estimate the distance of current frame value from each cluster. A proper estimation can successfully separate a video sequence in correct number of background and foreground clusters. Moreover, the foreground extraction depends both on the distance value as well as the type of cluster. This approach helps to include pixels which are close to foreground clusters with low distance values. Here, we provide a few examples to show the advantages offered by the algorithm. In Fig. 5.2, a Wallflower video sequence [6] - WavingTrees, has been shown with the ground-truth for the frame, the result for GMM and ADMGMM. The wallflower sequence shows a tree violently but periodically shaking in the background. After some time, a man comes and stands in front of the tree. This particular sequence
Figure 5.6 – Qualitative evaluations on ATON and SZTAKI video sequences: Two image groups from ATON Laboratory and SZTAKI Senoon sequences, respectively. Each group of images contain: (a) Original frame, (b) GMM output, (c) EGMM output, (d) CRFGMM output, (e) SAGMM output, and (f) ADMGMM output.

contains a multimodal background for the tree movement, and is appropriate to show how the distance measure affects the detection. Both of the methods compared, use 5 clusters. The clusters are also displayed in the figure. The clusters are aligned from left to right with the descending values of $\pi_{t,j}/\sigma_{t,j}$. Compared to the conventional GMM, ADMGMM has distinctively separated foreground and background clusters. Due to the better distance measure, the foreground is less mixed with background and provides better foreground detection. Also, it is visible that the foreground gradually affects the background layer for ADMGMM, while it randomly affects the layers for GMM.
To show the effect of background layer based foreground detection, two more sequences from Wallflower are used. Results for sequences Bootstrap (containing a number of moving and non-moving people) and ForegroundAperture (a flickering monitor as part of the background, is suddenly covered by a man in foreground) are shown in Fig. 5.3, compared with the ground-truth. The results including the pixels not falling into the background layer, are more complete and close to the ground-truth. The results for GMM and ADMGMM without including background layer based detection, contain low and erroneous detection.

5.2.2 Qualitative Analysis

The datasets shown in the work are already listed in Table 1.1. For the qualitative studies, a few video frames with output results are demonstrated in figures 5.4-5.6.

Fig. 5.4 shows two frames and corresponding outputs from CAVIAR fight, OneManDown sequence. The first frame shows a man moving from his starting position to get out of the scene. Thus, a “ghost” of the man persists in his original position. Also, due to unnecessary movements in the background and noisy data from surveillance, the output is very noisy for GMM, EGMM and SAGMM. CRFGMM is comparatively better. ADMGMM has performed much better and the amount of ghost is small. In the second frame, after the first man leaves, the second man starts to move away. Most of the compared methods contain ghosts for both persons. ADMGMM contains least noise and amount of ghost.

Fig. 5.5 provides one frame each from Carnegie Mellon dataset and ATON Highway dataset. Carnegie Mellon dataset has been captured with a vertically shaking camera. Thus, noise is incorporated in the outputs of all the compared methods. ADMGMM can better handle multiple clusters created due to the camera movements, with least amount of noise. SZTAKI Highway dataset contains a long highway sequence with illumination variations. This variation is captured in the outputs of GMM, EGMM and CRFGMM. SAGMM and ADMGMM, using variable number of
clusters, can perform better.

Finally, Fig. 5.6 depicts frames from ATON Laboratory dataset and SZTAKI Senoon dataset. In the Laboratory sequence, variations in foreground movements are used. People moving parallel to the camera axis have also been presented. The current shows one person moving parallel to the camera frame while, the other one parallel to camera axis. Illumination variation is also part of the video. SAGMM and ADMGMM can properly classify the foreground motion with less noise. For the Senoon dataset, a long sequence (1501 frames) of people entering and leaving an University through a glass door, is presented. Senoon corresponds to the time of noon. A particular scene is shown in the figure, where three people are standing and talking with each other while, a person is coming towards them. The scene represents a time-frame after the door was closed. GMM, EGMM and CRFGMM are affected by the glass door operation, the people standing for a long time as well as the noises due to grass movements and surveillance camera. While SAGMM still loses the foreground, ADMGMM can properly detect them.

5.2.3 Quantitative Analysis

Quantitative analysis has been carried out in accordance with Chapter 4. The analysis is kept similar to the ones presented in Section 4.3.2 to show the improvements over MRGMM as well as other state-of-the-art methods. For the first analysis, FPR, ACC, JC and MCC have been used as measures. Their definitions are provided in Section 1.4.2.

SZTAKI & ATON datasets do not have ground-truth for all the frames. Also, it is cumbersome to show results for individual frames. Instead, an average of the metric values for each dataset has been provided in Tables 5.1 and 5.2. ADMGMM without the iteration of equation 5.13 (ADMGMM\_NI) has also been tested and the results are provided. According to the results, T2FMRF\_UM has consistently performed well for FPR. However, its performance drops for other metrics. This signifies that
Table 5.1 – Quantitative evaluations of ADMGMM on datasets (I): CMS, SZTAKI

<table>
<thead>
<tr>
<th>Algorithms</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FPR</td>
<td>ACC</td>
<td>JC</td>
<td>MCC</td>
<td>FPR</td>
<td>ACC</td>
<td>JC</td>
<td>MCC</td>
<td>FPR</td>
<td>ACC</td>
<td>JC</td>
<td>MCC</td>
</tr>
<tr>
<td>EGMM</td>
<td>9.822</td>
<td>89.938</td>
<td>11.643</td>
<td>75.439</td>
<td>25.385</td>
<td>76.960</td>
<td>38.382</td>
<td>49.774</td>
<td>5.972</td>
<td>92.284</td>
<td>17.686</td>
<td>65.300</td>
</tr>
<tr>
<td>CRFGMM</td>
<td>1.637</td>
<td>97.806</td>
<td>18.681</td>
<td>80.400</td>
<td>18.201</td>
<td>80.239</td>
<td>31.577</td>
<td>43.375</td>
<td>3.370</td>
<td>94.662</td>
<td>19.159</td>
<td>66.601</td>
</tr>
<tr>
<td>SAGMM</td>
<td>5.948</td>
<td>93.840</td>
<td>15.956</td>
<td>79.833</td>
<td>8.018</td>
<td>91.426</td>
<td>58.673</td>
<td>69.792</td>
<td>1.104</td>
<td>96.967</td>
<td>26.665</td>
<td>76.226</td>
</tr>
<tr>
<td>FASOM</td>
<td>0.959</td>
<td>98.822</td>
<td>32.190</td>
<td>91.560</td>
<td>1.753</td>
<td>90.130</td>
<td>32.511</td>
<td>57.908</td>
<td>3.303</td>
<td>95.054</td>
<td>30.994</td>
<td>74.729</td>
</tr>
<tr>
<td>GMG</td>
<td>5.939</td>
<td>94.057</td>
<td>24.720</td>
<td>85.135</td>
<td>12.698</td>
<td>88.256</td>
<td>51.530</td>
<td>70.465</td>
<td>3.582</td>
<td>95.613</td>
<td>28.519</td>
<td>76.065</td>
</tr>
<tr>
<td>T2FMRF,UM</td>
<td>0.826</td>
<td>97.253</td>
<td>35.626</td>
<td>73.901</td>
<td>0.001</td>
<td>86.230</td>
<td>0.058</td>
<td>67.401</td>
<td>0.697</td>
<td>94.939</td>
<td>37.779</td>
<td>50.892</td>
</tr>
<tr>
<td>T2FMRF,UV</td>
<td>5.851</td>
<td>92.669</td>
<td>5.587</td>
<td>65.806</td>
<td>2.724</td>
<td>86.279</td>
<td>13.099</td>
<td>23.108</td>
<td>3.753</td>
<td>93.240</td>
<td>5.416</td>
<td>53.114</td>
</tr>
<tr>
<td>WavGMM</td>
<td>8.450</td>
<td>91.264</td>
<td>15.371</td>
<td>79.389</td>
<td>18.214</td>
<td>83.063</td>
<td>45.915</td>
<td>57.695</td>
<td>3.591</td>
<td>95.812</td>
<td>26.873</td>
<td>77.243</td>
</tr>
<tr>
<td>WavGMM,VC</td>
<td>3.102</td>
<td>96.382</td>
<td>18.116</td>
<td>81.598</td>
<td>4.525</td>
<td>93.443</td>
<td>62.667</td>
<td>73.066</td>
<td>0.753</td>
<td>98.257</td>
<td>28.992</td>
<td>78.564</td>
</tr>
<tr>
<td>ContGMM</td>
<td>3.059</td>
<td>96.498</td>
<td>19.328</td>
<td>82.785</td>
<td>9.954</td>
<td>90.077</td>
<td>54.152</td>
<td>66.125</td>
<td>3.807</td>
<td>95.759</td>
<td>20.325</td>
<td>70.803</td>
</tr>
<tr>
<td>ADMGMM</td>
<td>0.420</td>
<td>98.916</td>
<td>28.730</td>
<td>89.833</td>
<td>4.643</td>
<td>92.506</td>
<td>56.038</td>
<td>67.104</td>
<td>1.124</td>
<td>97.140</td>
<td>35.715</td>
<td>77.361</td>
</tr>
<tr>
<td>ADMGMM,N</td>
<td>0.067</td>
<td>99.233</td>
<td>62.037</td>
<td>91.309</td>
<td>2.970</td>
<td>94.062</td>
<td>61.697</td>
<td>72.905</td>
<td>0.564</td>
<td>97.152</td>
<td>46.435</td>
<td>75.7898</td>
</tr>
</tbody>
</table>
### Table 5.2 – Quantitative evaluations of ADMGMM on datasets (II): CMS, SZTAKI

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>FPR</th>
<th>ACC</th>
<th>JC</th>
<th>MCC</th>
<th>FPR</th>
<th>ACC</th>
<th>JC</th>
<th>MCC</th>
<th>FPR</th>
<th>ACC</th>
<th>JC</th>
<th>MCC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SZTAKI Seam</td>
<td></td>
<td></td>
<td></td>
<td>SZTAKI Senoon</td>
<td></td>
<td></td>
<td></td>
<td>SZTAKI Sepm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2FMRF UM</td>
<td>0.067</td>
<td>97.280</td>
<td>0.312</td>
<td>7.696</td>
<td><strong>0.439</strong></td>
<td>96.767</td>
<td>4.942</td>
<td>32.732</td>
<td><strong>0.454</strong></td>
<td>94.480</td>
<td>1.735</td>
<td>6.044</td>
</tr>
<tr>
<td>WavGMM</td>
<td>2.872</td>
<td>96.818</td>
<td>36.130</td>
<td>56.584</td>
<td>2.304</td>
<td>96.589</td>
<td>27.038</td>
<td>62.923</td>
<td>4.742</td>
<td>93.614</td>
<td>30.597</td>
<td>44.166</td>
</tr>
<tr>
<td>WavGMM LVC</td>
<td>2.384</td>
<td>97.074</td>
<td>34.994</td>
<td>54.214</td>
<td>1.860</td>
<td>96.761</td>
<td>26.560</td>
<td>61.161</td>
<td>4.575</td>
<td>93.544</td>
<td>28.279</td>
<td>40.705</td>
</tr>
<tr>
<td>ContGMM</td>
<td>2.710</td>
<td>96.925</td>
<td>36.411</td>
<td>56.301</td>
<td>2.165</td>
<td>96.620</td>
<td>26.680</td>
<td>61.831</td>
<td>5.428</td>
<td>93.072</td>
<td>29.985</td>
<td>43.991</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>2.598</td>
<td>97.037</td>
<td>37.724</td>
<td>57.532</td>
<td>2.024</td>
<td>96.743</td>
<td>27.994</td>
<td>63.112</td>
<td>5.134</td>
<td>93.329</td>
<td>30.636</td>
<td>44.686</td>
</tr>
<tr>
<td>ADMGMM</td>
<td>2.342</td>
<td><strong>97.346</strong></td>
<td><strong>42.473</strong></td>
<td><strong>62.134</strong></td>
<td>1.831</td>
<td>96.886</td>
<td><strong>33.543</strong></td>
<td><strong>67.109</strong></td>
<td>3.711</td>
<td><strong>94.607</strong></td>
<td><strong>34.074</strong></td>
<td><strong>47.841</strong></td>
</tr>
<tr>
<td>ADMGMM NI</td>
<td>1.226</td>
<td><strong>98.037</strong></td>
<td><strong>44.662</strong></td>
<td><strong>62.412</strong></td>
<td>1.504</td>
<td>97.082</td>
<td><strong>40.584</strong></td>
<td><strong>66.968</strong></td>
<td>1.482</td>
<td><strong>95.777</strong></td>
<td>30.530</td>
<td>44.030</td>
</tr>
</tbody>
</table>
Table 5.3 – BMC: Average metric values of ADMGMM and other methods over all datasets

<table>
<thead>
<tr>
<th>Method</th>
<th>RC</th>
<th>PR</th>
<th>F</th>
<th>PSNR</th>
<th>D</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>0.751</td>
<td>0.583</td>
<td>0.661</td>
<td>18.996</td>
<td>0.048</td>
<td>0.594</td>
</tr>
<tr>
<td>EGMM</td>
<td>0.759</td>
<td>0.604</td>
<td>0.676</td>
<td>22.542</td>
<td>0.035</td>
<td>0.599</td>
</tr>
<tr>
<td>CRFGMM</td>
<td>0.715</td>
<td>0.607</td>
<td>0.653</td>
<td>24.226</td>
<td>0.027</td>
<td>0.707</td>
</tr>
<tr>
<td>SAGMM</td>
<td>0.741</td>
<td>0.666</td>
<td>0.702</td>
<td>27.738</td>
<td>0.028</td>
<td>0.728</td>
</tr>
<tr>
<td>WavGMM</td>
<td>0.757</td>
<td>0.672</td>
<td>0.712</td>
<td>27.160</td>
<td>0.029</td>
<td>0.759</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>0.719</td>
<td>0.689</td>
<td>0.704</td>
<td>31.023</td>
<td>0.019</td>
<td>0.868</td>
</tr>
<tr>
<td>ContGMM</td>
<td>0.759</td>
<td>0.662</td>
<td>0.707</td>
<td>26.940</td>
<td>0.029</td>
<td>0.707</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>0.761</td>
<td>0.668</td>
<td>0.711</td>
<td>27.461</td>
<td>0.028</td>
<td>0.731</td>
</tr>
<tr>
<td>ADMGMM</td>
<td>0.801</td>
<td>0.712</td>
<td>0.749</td>
<td>37.714</td>
<td>0.009</td>
<td>0.939</td>
</tr>
<tr>
<td>ADMGMM_NI</td>
<td>0.799</td>
<td><strong>0.747</strong></td>
<td><strong>0.768</strong></td>
<td><strong>40.847</strong></td>
<td><strong>0.006</strong></td>
<td><strong>0.963</strong></td>
</tr>
</tbody>
</table>

Table 5.4 – CDW: Average ranking of ADMGMM and other methods for each dataset

<table>
<thead>
<tr>
<th>Datasets</th>
<th>BL</th>
<th>CJ</th>
<th>DBG</th>
<th>IO</th>
<th>SH</th>
<th>TH</th>
<th>OL</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>8.43</td>
<td>5.71</td>
<td>6.43</td>
<td>7.57</td>
<td>5.71</td>
<td>5.29</td>
<td>7.00</td>
</tr>
<tr>
<td>EGMM</td>
<td>7.29</td>
<td>5.57</td>
<td>7.43</td>
<td>7.00</td>
<td>7.86</td>
<td>9.29</td>
<td>8.43</td>
</tr>
<tr>
<td>CRFGMM</td>
<td>9.86</td>
<td>8.00</td>
<td>8.14</td>
<td>9.00</td>
<td>9.29</td>
<td>8.14</td>
<td>8.86</td>
</tr>
<tr>
<td>SAGMM</td>
<td>7.86</td>
<td>3.71</td>
<td>4.86</td>
<td>7.29</td>
<td>7.43</td>
<td>7.00</td>
<td>7.00</td>
</tr>
<tr>
<td>WavGMM</td>
<td>3.00</td>
<td>5.86</td>
<td>3.57</td>
<td>3.86</td>
<td>3.86</td>
<td>5.86</td>
<td>4.86</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>5.57</td>
<td>6.14</td>
<td>5.29</td>
<td>6.00</td>
<td>6.86</td>
<td>6.57</td>
<td>6.29</td>
</tr>
<tr>
<td>ContGMM</td>
<td>5.57</td>
<td>3.71</td>
<td>3.71</td>
<td>5.43</td>
<td>7.00</td>
<td>4.71</td>
<td>4.43</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>4.43</td>
<td><strong>3.29</strong></td>
<td><strong>3.29</strong></td>
<td>5.29</td>
<td>4.00</td>
<td>3.71</td>
<td>3.00</td>
</tr>
<tr>
<td>ADMGMM</td>
<td><strong>1.14</strong></td>
<td>6.14</td>
<td>6.71</td>
<td><strong>2.43</strong></td>
<td><strong>1.00</strong></td>
<td><strong>1.14</strong></td>
<td><strong>1.86</strong></td>
</tr>
<tr>
<td>ADMGMM_NI</td>
<td>1.86</td>
<td>6.86</td>
<td>5.57</td>
<td><strong>1.14</strong></td>
<td>2.00</td>
<td>3.29</td>
<td>3.29</td>
</tr>
</tbody>
</table>

126
the method detects less false positives but fails to detect true positives as well. Its performance severely drops for the Seam, Senoon and Sepm sequences. VISIONSYS performed well for Highway data sequence. The algorithm keeps a running average and median based background and uses subtraction to detect foreground. This provides a unimodal background modeling which works well for relatively static backgrounds with low amount of motion. However, its performance drops for CMS and ATON sequences where, background consists of shaky camera movements and waving grass, respectively. GMG also fails in these cases due to its weakness against rapidly changing pixel values. SAGMM is a consistent method with respect to all datasets due to its adaptive nature. However, it still suffers from the inherent problems of GMM. FASOM proves to be a consistent method providing relatively high metric values for most of the datasets. However, it becomes evident from the results that ADMGMM performs as one of the top methods in most of the datasets, due to its accurate distance measure and background modeling. Of course, no single method proved to be best for all datasets. But, ADMGMM shows more consistency compared to others. ADMGMM\_NI has outperformed ADMGMM in a number of datasets and closely followed it for the others. However, the main disadvantage of ADMGMM\_NI methods is the reduction in speed due to repeated ASW and HOG computation (discussed in Section 5.1.3 and 5.2.4). The reduction in speed is partly compensated in ADMGMM using the iteration in equation 5.13. Another problem with the features used is that they are highly reactive to the illumination changes. This is more prominent for ADMGMM\_NI where, the ASW and HOG features are calculated on each frame. Due to increase in shadows and illumination variations, ADMGMM\_NI features get affected for Sepm sequence.

Next, analysis on BMC and CDW databases are conducted similar to the ones presented in Section 4.3.2. Thus, for BMC, we show the average values of the metrics over the 9 real video datasets (Table 5.3) and for CDW, we used the ranking procedure described in [12], and provide the average ranking for the “types of datasets”: Baseline
Camera Jitter (CJ), Dynamic Background (DBG), Intermittent Object (IO), Shadow (SH), Thermal (TH) and an Overall score (OL) (Table 5.4). Finally, we provide the average ranks across datasets, of the methods for both databases using the ranking procedure for CDW (Table 5.5).

Table 5.3 shows that the average metric values are much better for ADMGMM and ADMGMM\_NI. Specifically, ADMGMM\_NI provides the best quality of results according to most of the measures. For CDW, according to Table 5.4, ADMGMM has the lowest rank closely followed by ADMGMM\_NI. Finally, Table 5.5 shows that the previous results follow the overall ranking. For BMC, ADMGMM\_NI has the best rank followed by ADMGMM. For CDW, ADMGMM has the best rank closely followed by ADMGMM\_NI. Of course, just like in Section 4.3.2, a single method cannot provide best results for each dataset. The apparent lower performance of ADMGMM\_NI for CDW as compared to ADMGMM can be attributed to the same reason for which ADMGMM\_NI relatively failed for Sepm sequence. The method is highly susceptible to illumination variations. BMC has a high number of simulated video sequences whereas CDW has only natural video sequences. Illumination variations and noises are much higher for natural sequences due to the unconstrained random nature of the environment. Hence, although the metric values for ADMGMM\_NI are still much better compared to the other methods, the method failed for a few sequences: specifically for CJ and DBG where the illumination variations and noises are too high.

5.2.4 A Comparison on Execution Time

GMM, EGMM, CRFGMM, SAGMM, VISIONSYS and the proposed methods (WavGMM, ADMGMM and ADMGMM\_NI) are implemented in MATLAB while the methods from BGSLIB have their implementations in C++. An unbiased comparison of execution speed needs all the methods to be in the same platform. Re-implementing each method in a single platform is a huge work. Instead, a comparison is presented in
Table 5.5 – Average ranking of ADMGMM and other methods over all datasets for BMC and CDW

<table>
<thead>
<tr>
<th>Method</th>
<th>BMC</th>
<th>CDW</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>7.14</td>
<td>6.71</td>
</tr>
<tr>
<td>EGMM</td>
<td>6.93</td>
<td>7.86</td>
</tr>
<tr>
<td>CRFGMM</td>
<td>6.07</td>
<td>9.86</td>
</tr>
<tr>
<td>SAGMM</td>
<td>7.00</td>
<td>6.57</td>
</tr>
<tr>
<td>WavGMM</td>
<td>5.78</td>
<td>3.57</td>
</tr>
<tr>
<td>WavGMM_VC</td>
<td>4.64</td>
<td>6.00</td>
</tr>
<tr>
<td>ContGMM</td>
<td>6.93</td>
<td>4.57</td>
</tr>
<tr>
<td>CurveGMM</td>
<td>4.50</td>
<td>3.57</td>
</tr>
<tr>
<td>ADMGMM</td>
<td>3.86</td>
<td><strong>2.71</strong></td>
</tr>
<tr>
<td>ADMGMM_NI</td>
<td><strong>2.71</strong></td>
<td>3.14</td>
</tr>
</tbody>
</table>

C++ keeping in mind that a practical implementation needs a fast programming language like C or C++. Also, most of the algorithms including the conventional GMM are implemented in BGSLIB. Thus, the execution time comparison is provided with the GMM, as well as the methods from BGSLIB. EGMM, CRFGMM, SAGMM and VISIONSYS have been left out of the comparison. However, EGMM and SAGMM have similar methodologies and execution complexity as GMM. CRFGMM is a slow method due to repeated neighbourhood computations, and VISIONSYS is a faster method compared to GMM due to simple mean and median filtering. For implementation of WavGMM, the C++ Wavelet2d libraries (http://wavelet2d.sourceforge.net/) are used. For the comparison of other methods, the “viptraffic” sequence (each frame of size 160 × 120 pixels) of 120 frames (source: MATLAB) has been used for this experiment. The execution times are tabulated in Table 5.6.

Referring to the table, ADMGMM is much slower in comparison to all the other algorithms mainly due to the support weight calculations. However, currently, there
<table>
<thead>
<tr>
<th>Method</th>
<th>Total time (seconds)</th>
<th>Frames per second</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>0.382</td>
<td>314</td>
</tr>
<tr>
<td>FASOM</td>
<td>0.402</td>
<td>298</td>
</tr>
<tr>
<td>GMG</td>
<td>0.325</td>
<td>369</td>
</tr>
<tr>
<td>T2FMRF, UM</td>
<td>0.495</td>
<td>242</td>
</tr>
<tr>
<td>T2FMRF, UV</td>
<td>0.606</td>
<td>198</td>
</tr>
<tr>
<td>WavGMM</td>
<td>0.394</td>
<td>304</td>
</tr>
<tr>
<td>ADMGMM</td>
<td>3.401</td>
<td>35</td>
</tr>
<tr>
<td>ADMGMM, NI</td>
<td>8.189</td>
<td>14.65</td>
</tr>
</tbody>
</table>

are a number of faster implementations of support weights [203, 204] that can be used to increase the speed considerably. As mentioned before, the iterative approach is much faster compared to the non-iterative approach. Also, square root and exponential operations are costly operations. In the calculation of support weights, square roots are used to find Euclidean distance and exponentials are used to provide monotonicity and scaling. They can be avoided by using absolute distance and optimizing the constant values for specific applications. By avoiding these operations, we have managed to improve the performance to over 100 frames per second for the same video sequence on the same hardware setup. Finally, our C++ implementation of the proposed algorithm has not been trimmed for minimum execution speed. A better implementation may improve the speed as well.
5.3 Summary

A novel method has been proposed to incorporate support weights and histogram of gradients, into a distance measure for foreground segmentation based detection of moving objects using Gaussian mixture model. The concept of background layer has also been used to properly identify foreground regions. The method demonstrates consistently good performance for several video sequences.
Chapter 6

Streaming Spatio-Temporal Video Segmentation Using Gaussian Mixture Model

Motivated from the current challenges, this work proposes a novel online video segmentation algorithm based on GMM. GMM has been successfully used for image segmentation[101] and foreground segmentation [4], though the methodologies are completely different. For image segmentation, EM algorithm is used. In contrary, EM is not applicable for foreground segmentation due to a high amount of processing involved. Thus, a recursive filter based algorithm is followed. In this work, a hybrid methodology is proposed to segment the images based on the distance from Gaussian means, but use a recursive filter for updating the parameters of the mixture model. The method is proposed keeping in mind the main challenges. Thus, it produces coherent segmentation for a long video sequence; it has an automatic cluster selection methodology and is less parameter dependent; due to the frame based segmentation, it has very low memory requirements, and has a high scalability.

The rest of the chapter is divided as follows. The proposed algorithm is discussed in Section 6.1. Several experiments and comparisons with other state-of-the-art methods are conducted in Section 6.2. Finally, the chapter is concluded in Section 6.3.
6.1 Proposed Method

GMM is well-explored in the domains of image segmentation and foreground segmentation. However, its use for spatio-temporal segmentation has been limited due to high complexity of EM algorithm [180]. Spatio-temporal segmentation of a video sequence would segment each frame of the video into distinctly separate regions based on colour or other image cues, and these regions must be temporally coherent. In the proposed approach, the initial spatial segmentation has been done using K-means as it is faster than GMM. GMM is used to propagate the clusters through each frame with cluster addition or removal as required, to maintain the temporal consistency. Unlike in the previous chapters, as the number of clusters $K$ changes for each time-frame, the subscript $t$ has been used to represent the time.

The algorithm begins with an initial number of clusters $K_t; t = 1$. The first frame is segmented by K-means to yield $K_t$ number of means $\{\mu_{t,j} : j \in [1, K_t], t = 1\}$. The GMM is initialized with means $[\mu_{t,1}, \mu_{t,K_t}]$. The SD $\{\sigma_{t,j} : j \in [1, K_t], t = 1\}$ are computed from the set of pixels allocated to each Gaussian. After initialization, the algorithm processes each frame. The processing has three distinct steps as explained below.

1) Cluster Assignment: At first, the distances between each pixel $x_t$ and the cluster means $\mu_{t,j}$ are computed using a function called cMeasure(·) (discussed afterwards). This distance value is termed as $D_{t,x,j}$. Let, $D_{t,x,j}$ is minimum for $jm$th cluster. Thus, $D_{t,x,jm}$ is compared against $\sigma_{t,jm}$ to find whether $x_t$ is an inlier to $jm$th cluster, as follows:

$$x_t \in \Phi(x_t \mid \mu_{t,jm}, \sigma_{t,jm}) \text{ if } D_{t,x,jm} \leq T\sigma_{t,jm}.$$  \hspace{1cm} (6.1)

Here, $jm$th Gaussian probability density function is represented by $\Phi(x_t \mid \mu_{t,jm}, \sigma_{t,jm})$, and $T$ is a constant multiple. $D_{t,x,jm} > T\sigma_{t,jm}$ denotes an outlier to $jm$th cluster. Let, $I_{t,j}$ and $O_{t,j}$ denote the set of inliers and outliers to $j$th cluster with $NI_{t,j}$ and
$N\mathbf{O}_{t,j}$ number of pixels, respectively. If $N\mathbf{I}_{t,j} > 0$ i.e. $j^{th}$ cluster has inliers, $\sigma_{t,j}$ is recursively updated as follows:

$$\sigma_{j,t+1} = \alpha \sigma_{t,j} + (1 - \alpha) \sqrt{V_{t,N\mathbf{I}_{j}}}$$

(6.2)

where, $V_{t,N\mathbf{I}_{j}}$ represents the variance of the $N\mathbf{I}_{t,j}$ inliers of $j^{th}$ cluster, and $\alpha (< 1)$ is the learning rate (typically 0.01 [4]). Eq. 6.2 assures that the SD of each Gaussian updates with new pixel assignment, as this updated SD will form a threshold for next pixel assignment. $\mu_{t,j}$ is not updated to reduce any flickering effect in subsequent frames. Finally, the weight or contribution $\pi_{t+1,j}$ of $\Phi(x_t | \mu_{t,jm}, \sigma_{t,jm})$ in GMM can easily be found out as: $\pi_{t+1,j} = N\mathbf{I}_{t,j} / \sum_{k=1}^{K_t} N\mathbf{I}_{t,k}$.

2) Cluster Creation: The outliers indicate the advent of new information and necessitate creation of new clusters. This raises two questions: i) how many new clusters would be required? ii) what would be the mean and SD of the new clusters? To answer question i, a cluster similarity based algorithm is developed. The algorithm is based on the fact that the outliers in $\mathbf{O}_{t,j}$ have minimum distance to $\mu_{t,j}$ compared to their distances from any other existing cluster mean (Eq. 6.1). Evidently, if a new cluster $j'n$ with mean $\mu_{t,jn}$ is created to assimilate the outliers in set $\mathbf{O}_{t,j}$, the distance of $\mu_{t,jn}$ from $\mu_{t,j}$ should also be minimum in comparison to its distance from any other existing cluster mean. We refer to cluster $j$ and $j'n$ as “mother” and “daughter” cluster, respectively. Considering this fact, if $N\mathbf{O}_{t,j} > 0$, $j'n^{th}$ daughter cluster needs to be created, situating close to $j^{th}$ mother cluster and assimilating outliers from set $\mathbf{O}_{t,j}$. Thus, if $K_{n_t}(\leq K_t)$ mother clusters have nonzero number of outliers, $K_{n_t}$ daughter clusters need to be created. In answer to question ii, we find the mean $\mu_{\mathbf{O}_{t,j}}$ and SD $\sigma_{\mathbf{O}_{t,j}}$ of the outliers in set $\mathbf{O}_{t,j}$ representing the parameters of $j'n^{th}$ daughter cluster. This also assures that each outlier would fall in one of the daughter clusters. Finally, at the end of this step, we have $K_t + K_{n_t}$ clusters.

3) Cluster Removal: Due to cues likes motion, illumination changes, $N\mathbf{I}_{t,j}$ changes with each frame, and may reduce to zero as well. With unnecessary clusters, the com-
Figure 6.1 – Automatic cluster formation, propagation and removal: Results for frames 45, 234 and 400 from “Atonement”. Each column contains a segmented frame, histogram of corresponding original frame, and GMM. Clusters formed between frame pairs (1-45), (45-234) and (234-400) are shown in Red, Green and Blue, respectively. GMM is shown in black.
putational burden increases without any change in segmentation result. To prevent that, a cluster removal approach is proposed. For this step, a predefined threshold $Th$ is used. Distances $D_{\mu_{t,i,j}} = \| \mu_{i,j} - \mu_{t,j} \|_2$ between each pair of clusters $(i, j) \in [1, K_t]$, are computed. Thus, a total of $\binom{K_t}{2}$ number of distances are computed. If $D_{\mu_{t,i,j}} < Th$, clusters $i$ and $j$ are close to each other. If further, $N_{I_{t,i}} > N_{I_{t,j}}$, cluster $j$ is removed and inliers in $I_{t,j}$ are assigned to cluster $i$. Similarly, if $N_{I_{t,j}} = 0$, cluster $j$ is immediately removed. Finally, some challenging scenes may need arbitrarily high number of clusters with each cluster having a low number of assigned pixels. Thus, a maximum cluster limit $K_M$ can be enforced. If $K_t + K_{n_t} > K_M$ for step 2, then $K_{n_t}$ is reduced to $K_M - K_t$. The mother clusters are sorted by decreasing values of $N_O_{t,j}$ and first $K_M - K_t$ clusters are chosen to create daughter clusters. The rest of the outliers are assigned to mother clusters instead. $K_M$ would depend on the computational capability of the hardware used.

At this stage, a discussion on $D_{t,x,j}$ is important. Function $cMeasure(\cdot)$ can be user-defined. For this work, effect of two distance measures are shown: 1) Euclidean distance, 2) Neighbourhood based distance. The methods based on 1 and 2 are termed “EuclidDist” and “NeighbourDist”, respectively. For NeighbourDist, a neighbourhood $N_{t,x}$ of pixels around $x_t$ is used and Euclidean distance between each pixel in $N_{t,x}$ to $\mu_{t,j}$ is computed. $D_{t,x,j}$ is found by averaging these distances over $N_{t,x}$. Mathematically, it can be expressed as follows:

$$D_{t,x,j} = \frac{1}{N_x} \sum_{(y_t \in N_{t,x})} \| y_t - \mu_{t,j} \|; \quad (6.3)$$

where, $N_x$ represents the number of pixels in $N_{t,x}$. Section 6.2 details the results for both methods. Finally, we provide a pictorial depiction of the algorithm in Fig. 6.2 and outline the three steps for each frame, after initialization:

1. Step 1: Assign each pixel $x_t$ to a cluster as an inlier or outlier based on Eq. 6.1. Update $\sigma_{t,j}$ based on Eq. 6.2. Assign $K_{n_t} = 0$. 

136
2. Step 2: For each cluster \( j \in [1, K_t] \), if \( N_{O,t,j} > 0 \), assign \( Kn_t = Kn_t + 1 \) (subject to \( K_M \)). Finally, find the \( Kn_t \) daughter cluster means and SDs as \( \mu_{O,t,j} \) and \( \sigma_{O,t,j} \), respectively.

3. Step 3: Compute distance \( D_{\mu_{t,i,j}} \) between each pair of clusters \( i \) and \( j \). If \( (D_{\mu_{t,i,j}} < Th \ AND \ N_{I,t,i} > N_{I,t,j}) \ OR \ (N_{I,t,j} = 0) \), cluster \( j \) is removed.

The steps are demonstrated in Figure 6.1 using 3 frames (45, 234 and 400) from “Atonement” sequence [24]. Atonement is chosen as it is a long video of 605 frames, and changes of cluster assignment can be properly observed. The image histograms and GMMs corresponding to each frame using EuclidDist, are positioned below the frames. As explained in the figure, some clusters propagate to a frame from earlier frames while some are generated or removed. A cluster (green) was created before frame 234, but was removed before frame 400. Again, a new cluster (blue) was generated between frames 234 and 400. Also, as SD for each cluster changes at each frame, the shape of the Gaussian changes accordingly. Finally, the GMM for frame \( t \) (shown in black) is found by multiplying \( j^{th} \) Gaussian with \( \pi_{t,j} \) and summing over all Gaussians. As observed, the GMM can approximately follow the histograms and has similar peaks even without EM. The close approximation denotes the accuracy of the algorithm. The next section details the complexity of the algorithm.

### 6.1.1 Complexity Analysis

Apart from the complexity of K-means iteration at the beginning, the complexity can be easily deduced by independently considering the steps of the algorithm listed before. Step 1 denotes cluster assignment, followed by cluster creation in step 2 and cluster merging in step 3.

1. Step 1: For each pixel, find the matching cluster based on Eq. 6.1. The approximate complexity is: \( K_M N \) where \( N \) represents the number of pixels in each frame.
Figure 6.2 – A graphical representation of the proposed video segmentation algorithm
Figure 6.3 – Qualitative evaluations on Label Propagation datasets: Temporal consistency results for datasets Garden (frames 12 & 75 in rows 1 & 2, respectively) and Ice (frames 12 & 80 in rows 3 & 4, respectively). The columns, from the left to right, represent the original image, followed by the ground-truth segmentation, the results for GBH, SWA, EuclidDist and NeighbourDist, respectively.
2. Step 2: For each cluster, assign the inliers, outliers and daughter clusters, if required. This step has an approximate complexity of: $K_M$ and is negligible.

3. Step 3: This step computes distance between each pair of clusters and removes a number of clusters based on some conditions. The approximate complexity is: $K_M^2$ which is again negligible in comparison to $K_M N$.

As the K-means is done only at the beginning, we do not consider it while analyzing the run-time complexity of the algorithm. Ignoring the initializations at the beginning, the complexity of step 1 is $K_M N$, or $O(N)$. As the other steps have constant complexity with respect to the number of data points or pixels, the complexity of the algorithm is governed by step 1 and hence, it is $O(N)$.

6.2 Experimental Results

The proposed algorithm is tested on the label propagation database from Chen [1] and video sequences used in [24]. Qualitative and quantitative comparisons are done with GBH and SWA, as they represent the best methods in the literature. However, as both are supervoxel based methods while the proposed approach is not, a supervoxel based quantitative analysis may not be appropriate. On the other hand, three metrics are available in literature for segmentation error analysis as already discussion in Section 1.4.1: 1) PRI - measures the likelihood of a pixel pair being grouped consistently in two segmentations, 2) VoI - computes the amount of information in one result not part of the other one, and 3) GCE - measures the extent to which one segmentation is a refinement of the other one. When compared to ground-truth, a higher value for PRI and lower values for VoI and GCE denote better results. As these measures do not require the number of clusters to be same as ground-truth, they are appropriate for a quantitative study. Each video dataset has several frames and corresponding ground-truths. The metrics are computed for each frame and averaged
over the number of frames for each video dataset. For comparison, the LIBSVX [23] implementations of GBH and SWA have been used. Both methods have a heavy dependency on parameters, and have been run with different levels of hierarchy. Based on the tutorial presented in LIBSVX, GBH and SWA have been run from levels 0 to 20 and 7 to 12 respectively. Using the quantitative approach, the best results for GBH and SWA are yielded with levels 20 and 12, respectively. Increasing hierarchy beyond these values may increase the PRI, but reduces overall score. For the proposed methods, initial $K_t$ is kept at 5. This does not influence the segmentation as $K_t$ varies for each frame. Also, the multiplier $T$ does not have a high influence on the segmentation and a value of 2.5 is used [4]. The only and most important parameter is the distance threshold $Th$. This controls the merging of clusters and has an immediate effect on segmentation. With a trial of runs for values 1 to 30 on each dataset, $Th = 25$ and $Th = 20$ are found to be good choices for EuclidDist and NeighbourDist, respectively and have been used for all experiments. The qualitative and quantitative discussions are provided in Sections 6.2.1 and 6.2.2, respectively.

### 6.2.1 Qualitative Results

Qualitative results are shown in Figure 6.3. Two frames far apart in time are used for sequences “Garden” and “Ice” to evaluate the temporal consistency over the sequence. As displayed, both proposed methods have a consistency of segmentation over the interval of frames. The consistency problem can be evident from the results of GBH and SWA. GBH and SWA have different segmented regions for the floor in two frames of Ice as compared to EuclidDist and NeighbourDist. Also, the segmentation results for EuclidDist and NeighbourDist do not have too much boundary overlap errors and the objects can be properly separated. Specifically, in the Garden sequence, the tree has overlapping segmentation and cannot be properly distinguished for SWA. EuclidDist and NeighbourDist, on the other hand, provide distinct boundaries. This is due to proper segmentation of the outliers. The proposed algorithm detects outliers in
each frame and can increase or decrease number of clusters for a suitable segmentation.

### 6.2.2 Quantitative Results

Quantitative results using GCE, PRI and VoI are provided in Table 6.1. The performances of EuclidDist and NeighbourDist may not be the best for all sequences, but are fairly competitive to the performances of GBH and SWA. Specifically, their performances are outstanding for Ice sequence, outperforming GBH and SWA by a large margin for PRI and VoI. Also, the proposed methods have advantage of real-time processing. For SWA, if a dataset over 60 frames is processed at once, the processing time is more than 5 hours (on a machine with 3.4 GHz Intel Core i7-3770 processor and 16 GB RAM) due to high memory usage. GBH has lower processing times [24], but not suitable for a real-time operation. EuclidDist and NeighbourDist can process more than 30 and 20 frames per second on the same machine, respectively, rendering their suitability for real-time processing with a competitive quality.

### 6.3 Summary

A video segmentation approach based on GMM has been proposed. The main contributions of the work are: (a) automatic determination of varying number of clusters over the frames, (b) maintaining temporal coherence and scalability and (c) linear time complexity. The applicability of the method is increased by the incorporation of a user-defined distance measure to determine the befitting clusters over each frame. The method can be considered fairly automatic as it depends mostly on the distance threshold for quality of segmentation. Using two types of suggested distance measures, the method performs competitively with two best methods in literature. The future work, therefore, includes the evaluation of the complete potential of the proposed method on the quality of segmentation by exploring other distance measures.
Table 6.1 – Average metric values of GBH, SWA and the proposed video segmentation method for 8 datasets from Label Propagation database[1]

<table>
<thead>
<tr>
<th>Datasets</th>
<th>GCE</th>
<th>PRI</th>
<th>VoI</th>
<th>GCE</th>
<th>PRI</th>
<th>VoI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GBH</td>
<td></td>
<td></td>
<td>SWA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bus</td>
<td>0.24</td>
<td>0.79</td>
<td>5.57</td>
<td>0.08</td>
<td>0.78</td>
<td>10.04</td>
</tr>
<tr>
<td>Container</td>
<td>0.13</td>
<td>0.70</td>
<td>3.79</td>
<td>0.05</td>
<td>0.68</td>
<td>6.36</td>
</tr>
<tr>
<td>Garden</td>
<td>0.21</td>
<td>0.75</td>
<td>4.70</td>
<td>0.08</td>
<td>0.73</td>
<td>10.93</td>
</tr>
<tr>
<td>Ice</td>
<td>0.09</td>
<td>0.52</td>
<td>4.15</td>
<td>0.03</td>
<td>0.44</td>
<td>6.82</td>
</tr>
<tr>
<td>Paris</td>
<td>0.12</td>
<td>0.32</td>
<td>5.59</td>
<td>0.07</td>
<td>0.31</td>
<td>9.67</td>
</tr>
<tr>
<td>Salesman</td>
<td>0.25</td>
<td>0.51</td>
<td>3.83</td>
<td>0.06</td>
<td>0.47</td>
<td>10.29</td>
</tr>
<tr>
<td>Soccer</td>
<td>0.11</td>
<td>0.77</td>
<td>4.33</td>
<td>0.08</td>
<td>0.76</td>
<td>6.98</td>
</tr>
<tr>
<td>Stefan</td>
<td>0.11</td>
<td>0.76</td>
<td>4.17</td>
<td>0.03</td>
<td>0.73</td>
<td>9.96</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>EuclidDist</th>
<th>NeighbourDist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus</td>
<td>0.26</td>
<td>0.39</td>
</tr>
<tr>
<td>Container</td>
<td>0.13</td>
<td>0.20</td>
</tr>
<tr>
<td>Garden</td>
<td>0.13</td>
<td>0.28</td>
</tr>
<tr>
<td>Ice</td>
<td>0.07</td>
<td>0.10</td>
</tr>
<tr>
<td>Paris</td>
<td>0.08</td>
<td>0.18</td>
</tr>
<tr>
<td>Salesman</td>
<td>0.31</td>
<td>0.18</td>
</tr>
<tr>
<td>Soccer</td>
<td>0.35</td>
<td>0.34</td>
</tr>
<tr>
<td>Stefan</td>
<td>0.11</td>
<td>0.25</td>
</tr>
</tbody>
</table>
Chapter 7

Conclusions

The dissertation is an effort to provide a detailed overview of segmentation in images and videos, followed by proposing a number of enhancements to conventional GMM towards segmentation. Segmentation is often considered as one of the most popular yet challenging research area in computer vision. In literature, image and video segmentation has been separately approached due to the fundamental differences in processing. In spite of the separate treatment, image and video segmentation share a set of common traits and hence, are connected. Through this work, we have tried to exhibit this connection with a thorough discussion on image, foreground and video segmentation, followed by establishing the common traits and the gradual progress from images, through foreground towards video segmentation. In Chapter 1, the fundamental aspects of each type of segmentation has been discussed followed by a discussion on the various methods and databases to validate the quality of segmentation. Chapter 2 is dedicated to providing an approximate categorization of each type of segmentation. Both categorization and validation are important towards understanding and practice of a concept. The dissertation imparts even more importance to categorization and validation as segmentation serves as a backbone or baseline to many other research areas in computer vision. Chapter 2 also indicates the foundation of the dissertation being the enhancements of GMM towards segmentation through clarification of its common impact in image and foreground segmentation and its possible use for spatio-temporal video segmentation. The Chapter concludes with a discussion on GMM for image and foreground segmentation.

With the foundation ready, the dissertation details the proposed enhancements of
GMM towards image and foreground segmentations in Chapters 3, 4 and 5, respectively; later, it introduces an efficient procedure to apply GMM for spatio-temporal video segmentation, in Chapter 6. Each of the proposed works consists of some contributions and limitations, as summarized next.

7.1 Contributions and Limitations

The contributions of the proposed work can be divided in three broad categories: image segmentation, foreground segmentation and video segmentation. Each of them has some uniqueness and some common traits, the main common trait being the use of GMM. The categories and their salient points are discussed in the next three subsections followed by a summary.

7.1.1 Image Segmentation

An efficient enhancement of conventional GMM towards image segmentation is proposed in Chapter 3 using Bilateral filtering to denoise the posterior probability map of the GMM based on MRF. The proposition brings out the following points:

1. The prevalent works on enhancement of conventional GMM based on MRF increase the implementation complexity and the direct implementation of EM algorithm becomes difficult. The proposed work tries to keep the implementation simple while maintaining the provision for EM.

2. The main reason behind enhancing conventional GMM is its inability to utilize the relationships among neighbouring pixels for segmentation. For natural images, the neighbouring pixels belonging to a common object often share strong spatial bonding and need to be classified based on this bonding. A filtering on the posterior probability map inherently utilizes this bonding and helps reduce noise by removing outliers from segmented regions.
3. In general, quality of segmentation is often reduced by overlapping regions over common boundaries. Bilateral filter provides a mean filtering while preserving the boundaries of regions. This partially cures the problem of erroneous segmentation at boundaries.

4. The main limitation of the proposed approach arises due to the use of Bilateral filter which has a slow computational speed, as compared to average or Gaussian filtering.

7.1.2 Foreground Segmentation

Two enhancements are proposed for conventional GMM towards foreground segmentation in Chapters 4 and 5. Chapter 4 proposes to incorporate multiresolution decomposition of the image data before processing by GMM and reconstruction afterwards. The following aspects are noticeable:

1. An intuitive study has been made to investigate the effectiveness of the proposition. The study reveals that the proposed enhancement provides a better approximation for a volumetric segmentation of the entire video using K-means, as compared to the segmentation provided by conventional GMM.

2. A generic framework is proposed to incorporate other multiresolution methods with proper reconstruction procedure, in the enhancement. To demonstrate the applicability of the framework, two recent multiresolution methods are used: Curvelets and Contourlets.

3. The conventional GMM always uses fixed number of clusters. The framework is modeled to use variable number of clusters so that the enhanced model becomes completely independent of parameters.

4. A short study has been carried out to demonstrate the effectiveness of the framework when used with any other foreground segmentation method. Two popular
methods are chosen for the study: KDE and Codebook. As each method has its particularity of implementation and effectiveness, a straight-forward use of the framework may not bring forth the same amount of improvement. The same has been verified from the study as enhancement on Codebook has shown more improvements as compared to that on KDE.

5. MRGMM consists of a limitation of unimodality. As the background is found by weighted averaging of all modes belonging to background, background consists of a single mode. Thus, a background of high dynamic nature may not be sufficiently modeled.

Keeping a note to the limitation of MRGMM towards segmentation, Chapter 5 uses a completely different approach to enhance the conventional GMM by proposing an advanced distance measure. The proposition highlights the followings:

1. As mentioned earlier, a dynamic background cannot be properly represented by a single mode. Absence of multimodality was a limitation for MRGMM. The proposed advanced distance measure does not affect the use of multiple modes. Thus, the limitation of MRGMM is successfully addressed.

2. The advanced distance is based on ASW and HOG. ASW provides unique representation for each pixel based on its spatial neighbourhood, while HOG provides more advantages when the neighbourhood has a lot of random textures and less continuity of colours. Using both of these features, the advanced distance measure can uniquely classify each pixel to an appropriate mode.

3. The variance of a congested foreground increases gradually with successive frames, due to the continuous variation of values in the pixel process. In such cases, the threshold of distance by variance may not suffice. The proposed distance measure incorporates variance in the measure itself, and uses fixed threshold as a solution.
4. The concept of background layer is used to properly separate foreground from the background part of the modes of the GMM.

5. Computation of ASW and HOG is more time-consuming compared to the distance calculation for the conventional GMM. This may create a bottleneck for a hardware realization of the proposed method.

### 7.1.3 Video Segmentation

GMM has been little explored for spatio-temporal video segmentation as already pointed out in Section 2.3. Noticing the success of GMM for image and foreground segmentation, an effort has been spent to extend it towards video segmentation. In Chapter 6, based on the methodology for applying GMM towards image and foreground segmentation, a hybrid method is proposed for video segmentation. The method provides streaming, real-time, automatic spatio-temporal video segmentation combining the advantages of both image and foreground segmentation based on GMM. The proposition has a number of important parts to consider:

1. The condition of streaming segmentation is to segment each frame on-the-fly as they are loaded into memory. This part is incorporated in the proposed approach through propagation of GMM clusters through frames, updating them when necessary, and modifying the number of clusters if required. This also reduces the load on the processing machine as the total video is not required to be present and processed as a whole.

2. The computational complexity of cluster creation, update and removal is dependent on the maximum number of clusters allowed. Thus, the method can be easily optimized for a machine based on its processing power, and can be executed in real-time.

3. The cluster processing is made totally automatic and the initial number of
clusters is mostly irrelevant to the overall performance of the method. Thus, the proposed method is automatic.

4. The main challenge of spatio-temporal video segmentation is to keep a temporal consistency of segmentation over a large number of video frames. Existing methods, which can approximately achieve this goal, suffer from heavy memory and processing power requirements. The proposed method provides a consistent segmentation while achieving a real-time performance due to its hybrid methodology of mode propagation influenced from foreground segmentation.

5. The only manual factor controlling the quality of segmentation is the threshold used for merging clusters. The effort to choose an appropriate threshold may hinder the applicability.

7.1.4 General Summary

The dissertation has been targeted towards enhancing conventional GMM for segmentation. The effort was spent to encompass both image and video segmentation in the process. However, as there was no straight-forward way to extend GMM for image segmentation towards videos, research has been carried out towards enhancing GMM for foreground segmentation. Finally, a hybrid methodology is proposed for video segmentation influenced from image and foreground segmentation. A number of important challenges for each type of segmentation have been addressed, while a few remained unaddressed. A section is provided next discussing some probable future directions for research.

7.2 Scope for Future Work

In coherence with previous section, this section is divided into three subsections each addressing a type of segmentation. Sections 7.2.1, 7.2.2 and 7.2.3 are dedicated
towards image, foreground and video segmentation, respectively.

### 7.2.1 Image Segmentation

A limitation for the proposed work has been pointed out in Section 7.1.1. In reference to this limitation, a number of possible improvements and extensions can be carried out as described next:

1. There have been a number of techniques for faster processing of a Bilateral filter. Such an approach can increase the execution speed of the proposed method by several times.

2. A study on the effect of different filtering operations can be done. The study may bring out a competitive or better candidate for filtering as well.

### 7.2.2 Foreground Segmentation

Both of the proposed methods have a number of possible future extensions. The extensions may help remove the limitations as well as improve the quality of segmentation.

1. A possible future work for MRGMM would be to observe the effects of different levels of multiresolution decomposition on the quality of foreground segmentation.

2. The advanced distance measure proposed in the second approach for foreground segmentation relies on two features: ASW and HOG. Both of these features are dependent on intensity variations and thus, get affected by illumination variations. Future work may be spent to make them more robust against illumination variations.

3. As speed is an issue while computing the ASW, a number of faster implementations can be used as indicated in Section 5.2.4.
7.2.3 Video Segmentation

The hybrid methodology leaves a lot of scope for future works as discussed next:

1. Segmentation has been carried out with only color values. Motion cues or other unique image features may be exploited to improve the quality of segmentation.

2. Two distance measures have been proposed. A search for more advanced distance measure may result in improvement as well.

3. As already pointed out in Section 7.1.3, automatic threshold computation is required to generalize the applicability. A feedback on segmentation quality during execution may help vary the threshold automatically based on the scene content.
References


Appendix A

IEEE Permission to Reprint

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of University of Windsor’s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.
Vita Auctoris

NAME : Dibyendu Mukherjee
BIRTH YEAR : 1983
BIRTH PLACE : INDIA
EDUCATION
2014 : Doctor of Philosophy
       Electrical and Computer Engineering
       University of Windsor, Windsor, Ontario, Canada
2010 : Masters of Applied Science
       Electrical and Computer Engineering
       University of Windsor, Windsor, Ontario, Canada
2006 : Bachelors of Engineering
       Electronics and Telecommunications Engineering
       Bengal Engineering and Science University, Howrah, India