An Efficient Algorithm for Background Subtraction in Videos

by

NS Wajiha Munir

Supervisor

Col. Adil Masood Siddiqui, PhD

A thesis submitted to the faculty of Electrical Engineering Department, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS

in

Electrical Engineering

DECEMBER 2019

SUPERVISOR CERTIFICATE

Certified that final copy of MS thesis written by Ms. **NS Wajiha Munir**, Registration No.**00000172722** of Military College of Signals has been vetted by undersigned, found complete in all respect as per NUST Statues/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: ____________________________________

Name of Supervisor: **Col. Adil Masood Siddiqui, PhD**

Date: ____

Signature (HoD):______________________________

Date: ______________________________________

Signature (Dean):_____________________________

Date: _______________________________________

ABSTRACT

Video processing is a substantially important branch of Image Processing and Computer Vision focused on extracting information from real scene videos. Among several other video processing techniques, Background Subtraction has attained great importance as a developing research area, during the past few years. It is a widely used technique particularly in surveillance videos, object tracking and detection, traffic or crowd monitoring etc. The goal of Background Subtraction is to segment the moving foreground part from the stationary background for a given scene, in order to make the post-processing tasks efficient and relatively easier.

In this research, we propose a background subtraction technique that aims at progressively fitting a particular subspace for the background that is obtained from L1-Low rank matrix factorization (LRMF) using cyclic weighted median (CWM) and a certain distribution of mixture of Gaussian (MoG) for the foreground. Expectation maximization (EM) algorithm is applied to optimize the Gaussian mixture model (GMM). The effectiveness of the proposed method is augmented by using subsampling technique to execute on an average more than 250 frames per second while maintaining a good performance in accuracy.

The performance of the proposed method is evaluated by comparing it with other state-of-the-art methods and it was concluded that the proposed method performs well in terms of F-measure and computational complexity.

DEDICATION

Dedicated to
MY BELOVED PARENTS, RESPECTED TEAHERS, FAMILY AND FRIENDS
for their guidance, encouragement, incredible support and love
that always lightened my ways.

ACKNOWLEDGEMENTS

I am thankful to Allah Almighty (all praise belongs to Him) for giving me the strength and ability to achieve my objectives and successfully complete this research work.

I am highly grateful to my supervisor, Col. Dr. Adil Masood Siddiqui for guiding and supporting me throughout my research work. His timely encouragement has led me to complete this research successfully. He not only imparted significant knowledge to me but also contributed extensively in my work. I am highly obliged to my Guidance and Evaluation Committee (GEC) members as well, that included Col. Dr. Imran Tauqir and Lec. Maryam Rasool for dedicating their precious time for my research work and always guiding me through it.

Finally I would like to pay my gratitude to all those people without whom I could never have accomplished my aim including my family, especially my parents and my teachers. I am grateful to them for their constant prayers and persistent support throughout this journey. May Allah bless them all with eternal happiness.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

Principal Component Analysis	PCA
Low Rank Matrix Factorization	LRMF
Cyclic Weighted Median	CWM
Gaussian Mixture Model	GMM
Expectation Maximization	EM
Maximum Likelihood	ML

Chapter 1

INTRODUCTION

Computer vision is an interdisciplinary field that mainly focuses on making the computers capable of acquiring a good understanding from images as well as videos. Being a scientific discipline, it deals with the theory behind the artificial systems that acquire data from images. Several forms of image information can be there, for instance video sequences, images obtained from numerous cameras or the information obtained from a medical scanner, which will be a multi-dimensional data. Computer vision finds its application as a technological discipline in the theories and models that are exploited to develop computer vision systems. Computer Vision has several sub-domains including video tracking, motion estimation, object recognition, scene reconstruction, image restoration and 3D pose estimation, indexing and learning.

Due to the enormous number of benefits achieved by the innovations in these disciplines, automated video based surveillance has emerged as a peculiar area of research. In the recent years, it has acquired immense attention because of the growing security threats in public places such as airports, railway stations, large shopping malls etc.

In computer vision, detection of objects in the video surveillance system is an open research area that has drawn the attention of numerous researchers, over the past few decades,. Various security tools can be used to monitor the moving objects thus providing a solution for video surveillance. Recently, automated video surveillance system requirement has increased tremendously owing to several kind of object tracking and detection in abnormal activities such as terrorism activities etc.

1.1. Motion Detection in Video Surveillance

In numerous applications of computer vision, for instance video surveillance, human computer interaction, low-rate coding of videos etc., change detection which is basically a low level vision task is utilized as an initial phase. For a given video sequence, the main aim is to recognize the arrangement of pixels in every frame which are fairly distinct compared to the previous frames. Subject to the usage domain, the calculation for such distinction of pixels may be different.

In video surveillance domain, motion detection is an essential part for the separation of moving foreground objects from the stationary background. Once the moving foreground is effectively segmented, it makes the subsequent tasks such as object tracking, object classification, activity recognition in videos etc. relatively easy. Various different methods can be used to perform motion segmentation , the most common of which are background subtraction, temporal differencing and optical flow. Among these aforementioned methods, background subtraction is the most prevalent method for the detection of regions in a video frame having motion. This is accomplished by obtaining the absolute difference of the current frame with the reference background frame and applying a threshold to the difference frame. This threshold is cautiously selected in order to avoid any false detection.

1.2. Background Subtraction

Video processing is a substantially important branch of Image Processing and Computer Vision focused on extracting information from real scene videos. Among several other video processing techniques, Background Subtraction has attained great importance as a developing research area, during the past few years. It is a widely used technique particularly in surveillance videos, object tracking and detection, traffic or crowd monitoring etc. For a given scene, background subtraction aims at segmenting the foreground part having motion, from the background part that is stationary.

In order to achieve this, firstly a mathematical model is designed for the stationary background. This can be achieved by designing a mathematical model for the stationary background and each successive video frame is then compared with this model. By applying a certain threshold on this comparison, it is possible to decide which of the pixels in each frame can be classified as foreground.

Generally any background subtraction technique involves three main steps:

1. Background Modelling
2. Testing Phase
3. Maintenance Phase

1.2.1. Background Modelling

In any Background Subtraction technique, background modelling is a vital step that aims at obtaining a probabilistic representation of the static background present in the video sequence. Subsequently, this model will be used to perform background subtraction by comparing each newly coming video frame with this model.

Background modelling is performed by using the following equation:

$$BM_N(x,y) = \frac{\sum_{m=1}^{N} I_m(x,y)}{N} \tag{1.1}$$

where

$$BM_N(x,y) = Intensity\ of\ the\ background\ pixel\ (x,y)$$

$$I_m(x,y) = Intensity\ of\ the\ pixel\ (x,y)\ in\ the\ m^{th}\ frame$$

$$N = Number\ of\ frames\ considered\ for\ the\ construction\ of\ background\ model.$$

In order to construct the background model, generally the initial few frames (say 100-200 frames) of the video are considered, that are void of the foreground objects. This step is also referred to as the training phase.

1.2.2. Testing Phase

In the testing phase, for each newly coming frame, the pixels are characterized as either part of the foreground or the background.

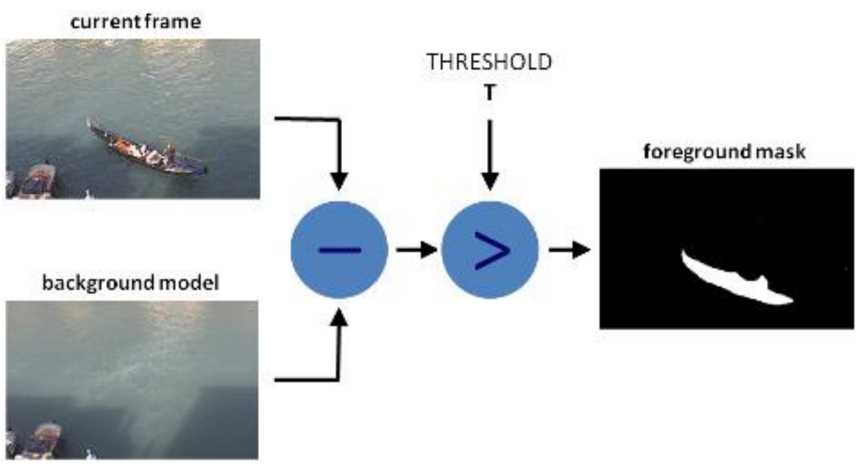


Figure 1.1: Training Phase in Background Subtraction

This is done in three steps: the first step evaluates the difference of the current frame with the background model, the next step is the generation of an appropriate threshold and the last step is to classify the pixels of the current frame based on the difference frame and the threshold value. The details of these steps are as follows:

1. **Frame Differencing:**

   In this step, the background model obtained in the first phase, also called as the reference frame, is subtracted from the newly coming video frame , called the current frame. If $I_t(x, y)$

denotes to the current frame and $BM_{t-1}(x,y)$ denotes the reference frame, then this difference can be achieved as:

$$D_t(x,y) = |I_t(x,y) - BM_{t-1}(x,y)| \tag{1.2}$$

2. Thresholding:

The next step involves the generation of a suitable threshold value τ which can be either a constant or a variable. The purpose of this thresholding step is to define a value that can be used for the segmentation of foreground mask from the background. All the pixel values in the difference image, that are greater than this threshold value are classified as the foreground and rest of the pixels as the background i.e.

$$F_t(x,y) = \begin{cases} 1 & if\ |D_t(x,y)| > \tau \\ 0 & otherwise \end{cases} \tag{1.3}$$

where τ is the threshold value and $F_t(x,y)$ is the foreground mask.

In general, the threshold value can be largely categorized as: Global threshold and Local threshold.

In global threshold, a fixed threshold value is used for all the pixels of an image. This method is appropriate only if the intensity histogram of an image has fairly distinguishable peaks corresponding to the foreground and the background. In case of strong intensity changes, it fails to produce accurate results.

In local thresholding, each pixel has an adaptive local threshold based on the intensity value of the local areas. This implies that unlike the global thresholding method, in local thresholding each pixel can be allocated a different threshold value which can be evaluated in terms of certain statistical parameters such as min, max, mean, median etc.

3. **Pixel Classification:**

In this step, the frame difference $D_t(x, y)$ obtained in the first step is compared with the threshold value τ defined in the second step. Based on this comparison, the pixels in the current frame are classified as either the foreground pixels or the background pixels as follows:

$$Obj(x, y) = \begin{cases} Foreground, \ if \ D_t(x, y) \geq \tau \\ Background, \ if \ D_t(x, y) < \tau \end{cases} \tag{1.4}$$

This step is also referred to as the foreground detection. The accuracy of this detection depends on the value of threshold τ. An appropriate value of τ reduces the probability of false detections i.e. the probability of wrongly classifying a pixel as belonging to foreground.

1.2.3. Maintenance Phase

For the improvement of pixel detection quality, background model needs to be updated repeatedly. This is done in the maintenance phase. Sometimes, it is possible that some pixels may be incorrectly classified as part of foreground. In order to solve this misclassification problem, there is a need that the background model be updated in order to lessen the chances of false detections. This update is performed according to some predefined rules. Two important factors in this maintenance phase are the learning rate and the frequency of update.

The learning rate, often denoted by "α" , determines how efficiently the background model can adapt to the dynamically changing scenes. It can have a constant value or may be adjusted dynamically to adapt to sudden changes in the subsequent video frames.

The update frequency determines how often the background model needs to be updated so that it can efficiently diminish the chances of false detections. If there is no significant change in the value of a certain pixel for successive frames, no update is required for this pixel but maintenance is required and done for every frame.

Furthermore, the quality of the detected foreground objects can be enhanced by applying certain morphological operators used in image processing such as erosion, dilation, opening, closing etc.

1.3. Major Challenges faced by a Background Subtraction Method

Some critical challenges faced in implementation of a Background Subtraction method are:

i. Selection of the initial frames.
ii. Development of a model for Background
iii. Selection of an effective threshold for the classification of pixels
iv. Updating process for the Background or the threshold or for both

Although significant amount of work has been done for object detection in sequential video frames, yet there is a need to develop robust methods that can cope up with most of the problems encountered in real time environments. The main challenges faced by any background subtraction technique, caused by the environmental conditions or external factors are as follows [2],[3]:

a. **Gradual Illumination Changes:**

In the outdoor environments, the illumination gradually changes along various times of the day. This illumination change has effect on the objects appearing in the scene being observed.

b. **Sudden Illumination Variations:**

In the indoor environments, sudden changes in illumination mainly occur due to turning on and off of the artificial sources of light. On the other hand, in outdoor environment the weather changes may result in sudden illumination changes such as if the sun suddenly gets covered by the clouds or thundering and lightening etc.

c. Shadows:

Shadows that are casted by moving objects complicate the process of background subtraction more as compared to the shadows of stationary objects, as they result in false and inaccurate object detection. Although the static objects, that are part of background, also cast shadows but since they are always casted at the same position or in the outdoor environment, at slightly moving positions due to the sun movement, they are not that much problematic and the background model can easily accommodate them.

d. Dynamic Background:

In the Background, there are some parts which exhibit changing appearances because they contain some sort of moving object for example, moving clouds in the sky, waving tree leaves, ripples in water, escalators etc. which have no significant importance in the scene interpretation.

e. Camouflage:

Camouflage refers to the appearance of objects that makes them difficult to be differentiated from the background. Their presence also makes the Background Subtraction process inaccurate, resulting in false detections.

f. Bootstrapping:

Bootstrapping is required because the training of a background model void of foreground part is generally unfeasible.

Additionally some other challenges are pointed out in [2], which the authors claimed that the background subtraction method should be capable of handling:

a. Moved Objects:

It refers to the detection of objects that are initially part of background but after sometime they are moved and needs to be detected as foreground.

b. Sleeping Person:

It refers to the foreground object that appears in the scene and after a while remains stationary, but needs to be detected as foreground.

c. Walking Person:

It refers to the objects that were initially stationary and have been learned as part of background. At some point later in time, they start to move and finally leave the scene. They need to be detected by the background algorithm as a foreground object.

But the above mentioned issues are application specific and need to be considered according to the priority list. In fact, the point in time from where a stationary object initially in background starts moving and is not fascinating any more, should be characterized by the application. Therefore, it does not need to be considered as a general upkeep issue for background. Additionally, it is surprising that the above mentioned issues can also be considered as singularities that a bootstrapping technique must handle.

1.4. Thesis Statement

The detection of moving objects in videos or other applications is a challenging task due to several issues such as shadow, illumination changes, occlusion, background motion, camera jitter as well as several different types of ambiguities like atmospheric disturbances or noise, object overlapping outliers etc. Furthermore, in real world videos, there are a lot of variations associated within the scene as well as the camera itself. These variations make background subtraction a challenging task and arises a need for the development of a robust technique that can not only efficiently extract the useful knowledge from these videos but also accounts for these variations. Till now, several techniques have been proposed to solve this issue but they cannot account for the variations in camera movements i.e. camera jitter effect. This work proposes a technique for background subtraction that is capable to account for the camera jitter effect while being computationally efficient as well as accurate i.e. having lesser number of false detections.

1.5. Objective

This research aims to improve the quality of extracted foreground and background with the following goals:

- Removing the shadowing effects as well as the dynamic camera jitter effects.
- Improving the efficiency by executing more frames per second.
- To reduce the computational complexity by using an efficient technique for background subtraction.

1.6. Advantages

The proposed technique for background subtraction has the following advantages:

- Adaption to dynamic variations in both foreground as well as background in videos from real scenes.
- Robustness to dynamic camera jitters.
- Fast processing speed i.e. process more number of frames per second while giving accurate results.

- Reduce computational complexity
- Better accuracy and improved efficiency

1.7. Area of Application

Background subtraction has a significant role in computer vision and image processing. Particularly, it founds its application in

- Object Detection and Tracking
- Long Term Scene Monitoring
- Video Surveillance
- Medical Surveillance
- Aerial Surveillance System
- Satellite Surveillance System
- Traffic Monitoring
- Human Machine Interaction
- Video Compression
- Content Based Video Coding
- Optical Motion Capture
- Biometric Identification System

1.8. Thesis Outline

Chapter 1 : This chapter gave a brief introduction of Background Subtraction, problem statement and the proposed objective for the thesis work.

Chapter 2 : In this chapter, an extensive literature survey has been carried out and the already existing techniques for background subtraction have been discussed in terms of their pros and cons.

Chapter 3 : This chapter is focused on the actual work done in this thesis. It includes all the theoretical as well as mathematical details of the proposed technique in detail.

Chapter 4 : This chapter covers the discussion on the results that are obtained when the proposed technique is applied to various video datasets.

Chapter 5 : This chapter includes the conclusion and further extension that can be done in the proposed work in future.

Chapter 2

LITERATURE REVIEW

In Computer Vision and Image Processing, video processing is a significantly important branch mainly focused on the extraction of information from real time videos. During the past few years, among various other video processing methods, background subtraction has attained significant importance and has become a developing area for research. Background subtraction aims at segmenting the foreground i.e. the moving part, from the background i.e. the stationary part. It is a well-researched area in Computer Vision, having numerous algorithms and a substantial amount of literature[1]. Most of the research in Background Subtraction is focused on the videos obtained from stationary cameras but now-a-days, numerous real life videos are captured from non-stationary cameras such as the hand-held camcorders, the smart phone cameras, traditional Pan-Tilt-Zoom (PTZ) cameras, the dashboard and wearable cameras etc. In such videos, Background Subtraction becomes a more challenging task as both the Foreground and Background pixels are non-stationary. Therefore, the need to develop state of the art algorithms for the videos obtained from non-stationary cameras is becoming increasingly important, as a large percentage of videos now-a-days are produced by non-stationary cameras.

Development of a robust method that can deal with the problems encountered during object detection using a background subtraction technique, should have a low memory requirement and can efficiently deal with the real life environment in minimum time , is another significant aspect that needs to be considered.

2.1. Basic Steps involved in any Background Subtraction Technique

Background Subtraction is considered as an essential pre-processing task for videos captured from static or moving cameras. It principally aims at obtaining a mathematical model for the static or slowly varying background and comparing each new frame of the video data with it. This comparison then classifies the pixels of each new frame as either part of the Background or the Foreground.

The fundamental steps that are part of any Background Subtraction method, are depicted in Fig. 2.1.

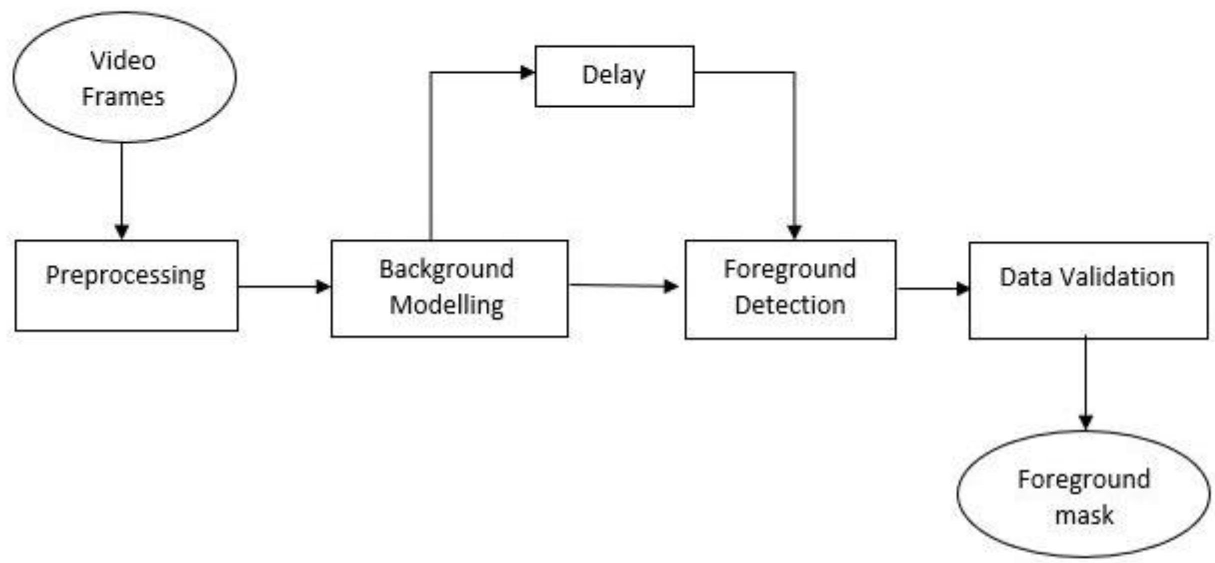


Figure 2.1: Fundamental Steps involved in Background Subtraction

In literature, several different techniques have been proposed for Background Subtraction, the simplest being the frame difference method which takes the difference of two consecutive frames of the video sequence by using some statistical method such as mean, median etc.

In moving object detection, the two main tasks are Background Modelling and Background Subtraction. In Background Modelling, a probabilistic representation of the static or stationary scene is obtained which is referred to as the Background model. In Background Subtraction, each newly coming frame is compared with this probabilistic model to perform subtraction [11],[12].

2.2. Relevant Approaches

In literature, a large number of techniques have been proposed for Background Subtraction that are classified according to different taxonomies. Based on spatial level consideration, these techniques are classified as follows:

i. Pixel-level Algorithms
ii. Block-level Based Algorithms
iii. Region-level Based Algorithms

2.2.1. Pixel-level Algorithms

In Pixel-level algorithms, only those features are utilized that are extracted for each single pixel position. Although these algorithms are efficient but the inter-pixel relationships are not taken into account in them. In literature, many such methods have been proposed, among which, Running Gaussian Averages [4], Median Filtering [5] and Gaussian Mixture Models [6] have achieved significant importance. From these aforementioned methods, a vast number of new systems have been derived.

a) Running Average Model:

For Background Modelling, one of the most basic approach is to use running average. In this approach, the average is obtained for the values observed at pixel position (x,y) in two consecutive frames. In order to approximate this average, the following recursive computation is used.

$$B_{t+1}(x,y) = B_t(x,y) + \alpha[X_t - B_t(x,y)] \tag{2.1}$$

Where

B_t = Resulting Background Image

α = Learning Rate

The learning rate controls how the background model can adapt effectively to the variations.

A difference image D_t can then be obtained from the background image B_t by computing the difference between all pixel values in each successive frame $I_t(x,y)$ and the values corresponding to those pixels in the Background image B_t. A decision rule is then applied to D_t in order to obtain the foreground mask F_t, as follows:

$$F_t(x,y) = \begin{cases} 1 & if\ |D_t(x,y)| > \tau \\ 0 & otherwise \end{cases} \tag{2.2}$$

where τ = Threshold value

The threshold value τ needs to be selected dynamically in such a way that it can adapt to the changes in viewing conditions. Several different approaches are mentioned in literature that can be used to compute its value. An overview of these approaches is provided in [14] in which a representative approach is evaluated for each of the recognized categories. The global thresholding procedure (mentioned above) is further extended by several other procedures, such as hysteresis thresholding or local thresholding, to improve the foreground detection.

Another approach using Running Average for Background modelling is presented in [15]. In order to enable the Background model to be adapted quickly, a selective update strategy is utilized which is able to maintain the foreground objects as well. The following equation is used for Background update:

$$B_{t+1}(x,y) = B_t(x,y) + \Big(\alpha_1\big(1 - F_t(x,y)\big) + \alpha_2\, F_t(x,y)\Big)\big(X_t - B_t(x,y)\big) \tag{2.3}$$

b) Median Model:

This approach models the background by computing median of the last N frames of the video sequence. The primary benefit of this method is that if the background is visible for greater than half of the frames (i.e. Number of frames having the background > N/2) , then the appearance of moving foreground objects does not degrade the background image. The drawback of this approach is its increased memory requirement because a buffer is required to store the last N frames for computation. In order to overcome this drawback, an iterative method is proposed in [5], that approximates the median model without requiring the buffering of frames. For each of the newly coming frame, the Background image updating is performed as follows:

$$B_{t+1}(x,y) = \begin{cases} B_t(x,y)+1, \ if \ I_t(x,y) > B_t(x,y) \\ B_t(x,y)-1, \ if \ I_t(x,y) < B_t(x,y) \end{cases} \tag{2.4}$$

Some other alternative methods [16], used to compute the median value, aim at the provision of more robust models for Background especially for the video sequences having frequently appearing foreground objects. With the increased robustness, the buffer size can be possibly reduced, consequently background image computation becomes faster.

The foreground objects are then detected by applying a threshold to the Difference image D_t obtained by estimating the difference for each of the incoming frame of the video I_t with the Background image B_t obtained by Median Model.

c) Running Gaussian Average Model:

For the computation of Foreground masks, a threshold value that has been derived statistically needs to be computed. For the computation of such threshold value, the variance of the pixel intensities at each pixel location can be estimated. One such background model has been used in

[4]. Two statistical measures commonly used for the computation of such threshold are the mean and variance. The mean value generally represents the expected background pixel value whereas the variance refers to the noise caused by the camera, which might vary for different Background positions as per the illumination conditions and the reflection properties. Therefore, a fully statistical based computations can be used to estimate the set foreground pixels.

d) Gaussian Mixture Model:

Gaussian mixture model can be used to describe a more complex distribution for the Background pixels. Gaussian mixture model consists of a combination of various classes, which are formed based on the appearance of the pixels, that taken together make up the model. The aim of this approach is to classify each pixel of the newly coming frame based on this model. Each Gaussian Mixture model consist of multiple Gaussian Distributions, each having its own mean and variance parameters and each distribution is assigned a weight that tells about the contribution of that distribution in the overall distribution of the model. These constituent Gaussian distributions are sometimes also referred to as the modes. The dominant model would perhaps refer to an observed scene's background model.

The main advantage of Gaussian Mixture Model is its ability cope up with multi-modal appearances of background e.g. moving clouds, waving tree leaves, ripples in water etc. Furthermore, it has the ability to adapt to the real time observed scene with a low memory requirement. In [17], a Gaussian mixture model having three gaussian distributions was proposed to model the appearance of vehicles, shadows and the road, at pixel level in a traffic monitoring application. A generalization of this model was proposed in [6] and it was further developed in [18] to make it capable of coping with non-stationary cameras.

2.2.2. Block-level Based Algorithms

In Block-level based method, the entire image is split into smaller blocks. Background is modelled by the features extracted from each of these blocks. Although block-level based methods are generally more robust to noise as compared to pixel-level based methods, they are computationally expensive and the detected foreground objects are not very smooth. Several different approaches based on Block-level are proposed in literature such as Normal Vector Distance based approach [7], Local Binary Pattern (LBP) Texture based approach[8] etc.

a) Local Binary Pattern:

With the purpose of vigorously coping up with the changing illumination circumstances, a block-level based approach using textures and its extension to the pixel-level approach has been considered in some Background Subtraction techniques. Rather than using intensity or color features, the background statistics are captured in these methods by using discriminative texture measures. Local Binary Pattern (LBP) is used for the computation of these features.

LBP is a basic yet extremely proficient texture operator that applies thresholding to the neighborhood of each pixel in order to label the pixels of an image and considers the result as a binary number.

This can be effectively computed as:

$$LBP = \sum_{p=1}^{P} s(X_i - X_c)2^p \tag{2.5}$$

where

$s(x)$ is a function defined as:

$$s(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{2.6}$$

X_c represents to the center pixel's intensity whereas X_p represents to the intensity of the 'p' pixels considered in the neighborhood of the center pixel.

In a generalized LBP operator as in [8] , a set consisting of 'P' neighboring pixels is considered. These pixels are equally spaced and located on a circle whose radius is 'R'. For each of the image location under consideration, 'K' number of weighted LBP histograms are estimated that are updated repeatedly using an updating process similar to the one proposed in [6] for the update of GMM. Similarly, the LBP histogram is also computed for each newly coming video frame. The distance between the two histograms, that is, the one computed for each newly coming video frame and the 'K' number of LBP histograms corresponding to each image position considered, is then computed in order to accomplish the background subtraction task.

In [25], Volume Local Binary Pattern (VLBP) operator had been presented, that takes into account the dynamic textures, by concatenating LBP histograms computed for the three orthogonal planes but its computational cost is increased. In order to alleviate this computational cost issue, Spatio-Temporal Local Binary Pattern (SLBP) operator was presented in [26], which basically comprises of the weighted sum computed for the two successive LBP histograms.

In dynamic scenes, the LBP histograms provide an efficient and robust way to handle the illumination changes, provided that the textures which are part of the observed scene are clearly noticeable. Nonetheless, no principled manner is provided by them for the evaluation of the distance between every new observation and the background model.

2.2.3. Region-level Based Algorithms

This approach splits the entire image into a set of regions. These regions are then categorized, based on spatial consistency criteria, as either the Background or the Foreground. In literature, there are a few such algorithms that are purely Region-level based algorithms because it can be computationally expensive to find a meaningful region in an image based on spatial consistency criteria. Therefore, most of the times, some alternative approach is combined with these approaches, for determining the regions and the region classification itself is carried out afterwards. However, some algorithms are still there that are purely region-based approaches such as the one proposed in [9], in which Partial Directed Hausdorff distance is utilized. Another such algorithm is proposed in [10] in which Spatial-Color Gaussian Mixture Models are used to model the Foreground and Background objects.

a) Non-Parametric Kernel Density Estimation:

The Non-parametric models are used for modelling Backgrounds in order to cope up with the arbitrary distributions and the high frequency variations. Kernel Density estimation (KDE) is one such non-parametric model in which an estimate of the probability of detecting a given pixel value X_t at any time t can be obtained non-parametrically. Given the pixel sample $X=\{X_1,X_2,\ldots,X_N\}$, using the Kernel estimator K, the probability can be computed as follows:

$$p(X_t) = \sum_{i=1}^{N} \alpha_i K(X_t - X_i) \tag{2.7}$$

Where α_i= Weighting Coefficients that are usually taken to be uniform i.e. $\alpha_i = \frac{1}{N}$

This probability can be computed efficiently by considering the Kernel Estimator to be a Normal function $N(0,\Sigma)$, with the assumption that the various color channels are not dependent on each other. The precalculated look up tables for the kernel function can be used, provided that the bandwidth and the difference in intensity value i.e. $(X_t - X_i)$ are given.

In [19] and [20], it was first proposed that non-parametric methods can be used for Background modelling. The problem with the initially proposed methods was high memory requirements as they needed to store the complete sample set of frames that were to be used for density estimation. In order to solve this problem, a mean-shift mode finding estimation technique was proposed in [21]. Another approach which uses the balloon variable size kernel is used in [22], which eliminates the need of estimating the kernel size parameter.

In general, the computational cost of Kernel Density Estimation (KDE) methods is high. Therefore, for less complex scenes Gaussian Mixture Model can be regarded as a better model for background [22], as it provides a more compact representation of Background making it suitable for subsequent processing steps such as shadow detection etc.

b) Eigen Background Model:

Eigen Background models are used to account for the changes in illumination in each frame, by taking into consideration the spatial correlation among the pixels. In order to compute an Eigen Background model, a set of N frames is considered. For these N frames, a mean Background image and its covariance matrix are computed. Next, eigenvalue decomposition is applied on the covariance matrix to diagonalize it. For the dimensionality reduction of the space, Principal Component Analysis (PCA) is used. In PCA, only the M eigenvectors are preserved which correspond to the highest eigenvalues. These eigenvectors, corresponding to the highest eigen values, are then saved in a matrix Φ_{Mp} having size (M x p) where p corresponds to the the number of pixels present in a frame.

Firstly, the mean standardized image vector is acquired for each input frame I_t , that is projected onto the eigenspace. Afterwards, it is projected back onto the image space by using the eigenvector matrix Φ_{Mp} and its transpose, respectively. As the eigenspace presents a robust model only for the background and not for foreground, therefore, there should be no moving objects in the image B_t

that is to be projected back on to the image space. Hence, the foreground containing the moving objects is detected by evaluating the Euclidean distance of the input image I_t with the back-projected image B_t and applying a threshold on the Euclidean distance thus obtained.

The first Eigenspace model based approach was proposed in [23] which was further extended in [24] to allow the threshold value to be computed automatically and the added ability to adapt to the dynamically varying scenes with the help of Incremental Principal Component Analysis (IPCA).

In real time computer applications, subspace modeling is very appealing as its computational cost at the classification time is very low but it needs all the training images to be allocated. Moreover, when the actual fundamental method is extended for background updation that is a fundamental constraint in the applications of visual surveillance, its complexity is considerably increased.

2.3. Level of Research carried out on the Topic

Over the past few years, various research studies have been carried out on background subtraction due to which it has attained great importance as a developing research area. Some of these recent researches have been taken as reference for the development of the proposed model.

As described earlier, background subtraction is a widely used technique particularly in surveillance videos, object tracking and detection, traffic or crowd monitoring etc. ,where the main focus is to extract the moving objects i.e foreground from the static background[27]. However, the detection of motion in videos or other applications is still a challenging task due to several issues such as shadow, illumination changes, occlusion, background motion, camera jitter as well as several different types of ambiguities like atmospheric disturbances or noise, object overlapping outliers etc.[28]. To overcome these challenges, statistical models are most effective ones. Statistical models may be non-parametric or parametric. Some of the most commonly applied non-parametric methods are the Kernel Density Estimation(KDE) and Eigen Value Decomposition but these methods have an extensive memory requirement as well as a high computational complexity [29]. On the other hand, parametric statistical models rely on the use of statistical distributions for background modelling. One of the most popular statistical model is finite Gaussian Mixture Model (GMM), capable of coping with slight illumination changes as well as moving background with small repetitive motion [29]. As with non-parametric method, parametric methods have their own limitations such as the learning parameters need to be set automatically, have to cope with complex dynamic background, have to dissociate shadows from object etc [29]. In order to accommodate these challenges and overcome the aforementioned limitations, [34] proposed an improvised version of GMM for detection of moving objects. It uses the Gaussian components to model the intensity values of a block of pixels and compensates for the learning rate limitation by using a dynamic learning rate. The consideration of pixel block instead of single pixel value reduced the computation time almost 4 times, keeping the performance almost similar to previous methods [34].

Recently Matrix decomposition methods, such as Robust Principal Component Analysis(RPCA), have become an efficient framework for background subtraction. These methods aim to break down a matrix into low-rank (for background) and sparse (for moving objects) components. However, in some scenarios, due to increase in the input data size and lack of sparsity constraints, matrix decomposition methods show weak performance as they are not able to handle the challenges faced in real time, resulting in misclassification of foreground areas. In order to resolve the aforementioned problem, an online framework that uses a single unified optimization for simultaneous detection of foreground as well as learning of background is proposed in [30]. This method has better performance, as it provides a more reliable and efficient low-rank component but it cannot be used for moving cameras. Although RPCA provides a good framework for background subtraction, it still has a very high computational complexity and huge memory requirements because of its batch optimization. In order to solve this issue, online RPCA is developed which can process such high dimensional data through stochastic manners. However, the sparse component obtained by OR-PCA cannot always handle numerous background modelling challenges, which degrades the performance of system. To overcome these challenges, [31] presented a multi-feature based OR-PCA scheme. Integration of multiple feature into OR-PCA not only improves the quality of detected foreground but also enhances the quantitative performance of this technique, as compared to single feature OR-PCA and RPCA through PCP based methods[31] . However, when OR-PCA is applied to real sequences which have dynamically changing background, the performance of OR-PCA is also reduced. Therefore, there is a need for enhancement in OR-PCA to cope up with the increased complexity and variety of videos. In [32], an online algorithm is proposed that is based on Incremental Nonnegative Matrix Factorization (INMF), which resolves the problems encountered in OR-PCA by using non negative and structured sparsity constraints. In complex scenes, this algorithm reduces the number of missed and false detection.

Subspace Learning methods such as Matrix Completion (MC) and Robust Principal Component Analysis (RPCA) have been explored and attained significant attention during the last few years [35,36]. These methods are based on low rank modeling and are meant to reduce the dimensionality in a very high dimensional space. Unfortunately, there are some prevalent challenges with most of the matrix decomposition algorithms that are based on conventional matrix completion and RPCA. Firstly, these methods use batch processing, secondly for every iteration of the optimization process, all the frames have to be accessed. As a result, a large amount of memory is required for these methods and are computationally inefficient. The method proposed in [33] considers the sequence of image as constructed from a low-rank matrix for background and a dynamic tree-structured sparse matrix for foreground. It solves the decomposition by the use of estimated Robust Principal Component Analysis (RPCA) which is augmented to make it capable of handling camera motion. This method decreases the complexity level, requires less time for computation and does not require substantial amount of memory for larger videos. Similarly, to get an estimate of a robust background model, in [35] a spatiotemporal low rank matrix completion (SLMC) algorithm is presented for dynamic videos. In the proposed method, spectral graphs are regularized for encoding the spatiotemporal constraints. Furthermore, for dynamic frames extraction, SLMC algorithm is augmented to Spatiotemporal RPCA (SRPCA). Together these algorithms make the process robust and accurate but in SRPCA as both the foreground and background are optimized simultaneously, hence the computational time is increased.

Considering the pros and cons of all the techniques reviewed in the literature survey, in this work we have proposed a technique that is computationally efficient and has improved accuracy in terms of detected background and foreground, with lesser number of false detections.

2.4. Summary

In this chapter, we carried out a literature survey in order to explore the already existing techniques for background subtraction in videos, considering all their benefits and drawbacks. In the next chapter, we will discuss the technique developed in this work based on the knowledge obtained from this literature survey.

Chapter 3

PROPOSED METHODOLOGY

In this chapter, the proposed technique for background subtraction in videos is discussed in detail along with the mathematical modelling involved at each step. This technique is then applied to different datasets and the obtained results are discussed in the next chapter.

3.1. How it Works?

Firstly the video frames are preprocessed in order to convert the raw input video to a format that is suitable for the subsequent steps. This step eliminates all the unwanted noise as well as the jitter effects, if present in any frame. In order to eliminate the camera jitter effects, affine transformation operator is involved that acts to align the successive frames. This makes the proposed technique proficient to adapt to an extensive variety of transformations in the background such as rotation, translation, scaling or a combination of all these, thereby making it more robust for the videos having dynamic camera jitters.

The next step is to perform principal component analysis (PCA) for reducing the dimension of the large video data matrix **X.** The reduced data matrix thus obtained is taken as input for the L_1 low rank matrix factorization (L_1-LRMF). The purpose of using PCA followed by LRMF is to obtain the initial subspace for the background and the initial MoG parameters by using the MoG algorithm on the extracted noise, and then calculating the initial subspace matrices for the subspace learning. L_1- LRMF is performed by using the cyclic weighted median (CMW) method which decomposes the LRMF problem into several subproblems in order to make the minimization step easy and fast. As opposed to the conventional methods which used the same noise distribution for the entire video frames, in this method a separate distribution from a mixture of Gaussians (MoG) is used to model the noise or foreground for each frame of the video. Expectation Maximization (EM) algorithm is exploited to optimize the Gaussian Mixture Model (GMM).

The gist of proposed work is that for each frame $\mathbf{x}_t$ of the video , the aim is to progressively fit a certain subspace for background that is obtained from L1-LRMF and a certain MoG distribution of noise for the moving foreground. This fit is achieved by regularization of the background and foreground knowledge ascertained from the prior frames.

The basic block diagram for background subtraction as described above by our proposed method is depicted in Fig. 3.1.

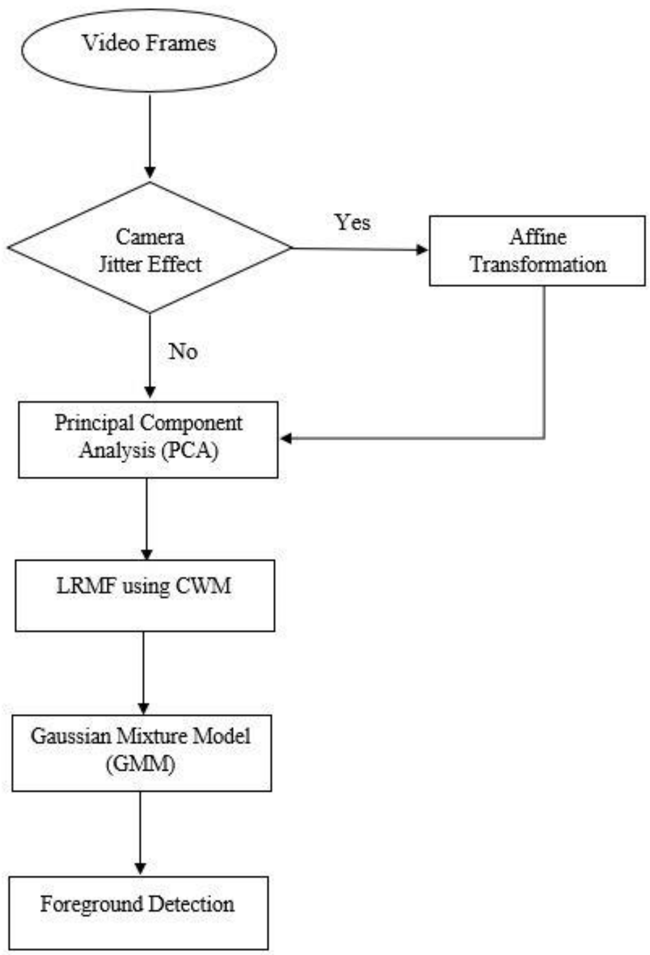


Figure 3.1: Block Diagram of the Proposed Methodology

Each of these block is discussed in detail in the subsequent section along with the explanation of how each block is contributing in the overall process of Background Subtraction.

3.2. Principal Component Analysis

Now-a-days, possibly the most extensively utilized statistical tool for dimensionality reduction and data analysis is principal component analysis (PCA). Its function as a dimensionality reduction tool is to reduce a larger set of variables to a smaller set while preserving maximum information contained within the original larger variable set.

In multivariate analysis, PCA is one of the simplest true eigenvector-based method, the main aim of which is to reveal the internal organization of data in such a way that gives the best description of the variance present in data. For a multivariate dataset which is envisioned as consisting of a coordinates set that is in a high dimensional space of data i.e. every axis represents one variable, PCA can provide a lower dimensional view for such a high dimensional data. This is achieved by considering the first few principal components having maximum variance, so that the dimensionality of the transformed data is reduced.

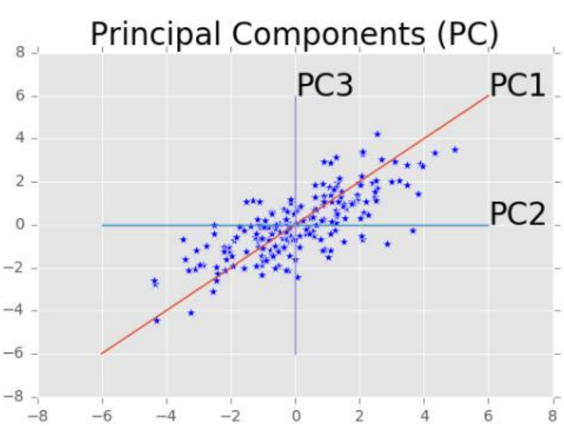


Figure 3.2: A data set having three Principal Components

3.2.1. Intuition

PCA can be perceived as a method that fits an ellipsoid having n-dimensions to the considered dataset, where each ellipsoidal axis represents a principal component. A smaller ellipsoidal axis implies a lesser variation along that axis, therefore even if we omit the principal component corresponding to that axis from our dataset representation, we will merely drop a proportionately insignificant amount of information.

In order to determine the ellipsoidal axes, firstly there is a need to center the data set around origin. This can be done by subtracting the mean of all variables from their respective dataset points. Next, the covariance matrix of the data has to be computed, and the eigenvalues as well as their corresponding eigenvectors are estimated for the obtained covariance matrix. Each of the orthogonal eigenvectors needs to be a unit vector therefore we need to normalize each of the determined eigenvectors. Now each unit eigenvectors that are mutually orthogonal, can be taken as an ellipsoidal axis that fits to the data. These basis which we will choose, will tend to transform our covariance matrix in such a way that its diagonal elements will correspond to the variance of each axis making it a diagonalized matrix. In order to determine the proportion of variance represented by each eigen vector, divide the corresponding eigen value of that eigen vector by the sum of all the eigenvalues.

However, this method is sensitive to data scaling, and no such consensus is there as how the data can be best scaled in order to obtain optimal results.

3.2.2. Steps involved in Principal Component Analysis

In the present age, the enormity of data has not only been a challenge for the computer hardware but for the performance of various machine learning algorithms as well. The identification of patterns in data is the main goal of PCA. It aims at detection of correlation among variables. The attempt for dimensionality reduction only makes sense when there exists a strong correlation among the variables. In a nutshell, PCA is used to project a dataset having a high dimension onto a subspace having a low dimension, by determining the directions of maximum variability in the dataset of high dimension in such a manner that most of the information is retained. In order to achieve this dimensionality reduction, PCA involves five main steps which are depicted below in in Fig. 3.3 in the form of a flow chart.

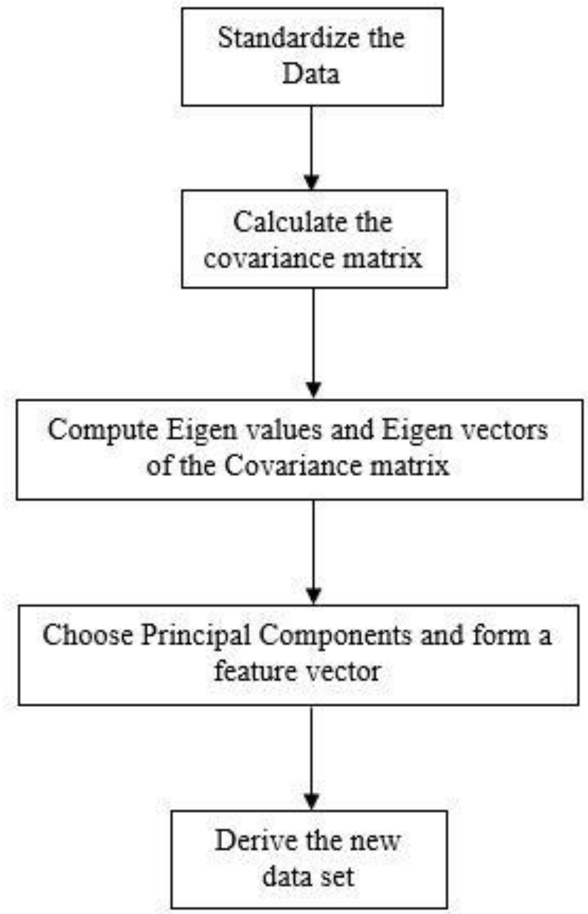


Figure 3.3: Flow chart of Principal Component Analysis (PCA)

Step 1: Standardize the data

The first step is to standardize the available data set so that PCA can work properly. This is achieved by computing the mean of each variable and subtracting it from all the values of that variable in the data set. This step will re-center the data i.e. the standardize data will have center at (0,0).

Step 2: Calculate the Covariance Matrix

The next stage is to compute the covariance matrix of the data. Denoted by Σ, the covariance matrix is a *dxd* matrix, each element of which corresponds to the covariance between two variables. If $\boldsymbol{X}$ represents the data matrix, then we can compute its covariance matrix as:

$$\boldsymbol{\Sigma} = \frac{1}{n-1}\left((\boldsymbol{X} - \vec{\boldsymbol{x}})^T \, (\boldsymbol{X} - \vec{\boldsymbol{x}})\right) \tag{3.1}$$

where $\vec{\boldsymbol{x}} = \frac{1}{n}\sum_{i=1}^{n} x_i$ is the mean vector $\vec{\boldsymbol{x}}$, which is a d-dimensional vector. Each element of this vector is the sample mean of variables in the dataset. On the other hand, if we take X to be the standardize data set then we can compute the covariance matrix simply as:

$$\boldsymbol{\Sigma} = \frac{1}{n}\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}} \tag{3.2}$$

The covariance matrix thus obtained is a symmetric matrix with the diagonal values corresponding to the variances of the variables and all the other values correspond to the covariance between any two variables.

Step 3: Computation of the Eigen values and the Eigen vectors

After computation of the covariance matrix, the next step focuses on finding out the eigenvalues with the corresponding eigenvectors for the obtained covariance matrix. This is known as eigen decomposition. The computed eigen values will correspond to the variance present in the dataset whereas the eigen vectors will represent the direction corresponding to that variance.

For the data matrix $\boldsymbol{X}$, the following characteristic equation is solved in order to obtain the the eigen values λ:

$$\det(\lambda \boldsymbol{I} - \boldsymbol{X}) = 0 \tag{3.3}$$

where $\boldsymbol{I}$ represents identity matrix having same dimensions as that of $\boldsymbol{X}$. The eigen vector $\boldsymbol{v}$ corresponding to each eigen value λ can be acquired by solving the subsequent equation:

$$(\lambda I - \boldsymbol{X})\boldsymbol{v} = 0 \tag{3.4}$$

Step 4: Formation of Feature vector

By ordering the eigen vectors according to their eigen values arranges the components according to their significance. The first principal component corresponds to the eigen vector having the largest eigen value. The data dimensionality can be lessened by ignoring the components that have lesser significance, without loosing much of the useful information contained in the data. The number of dimensions of the reduced data set will depend on the number of eigenvectors chosen. These eigenvectors are then arranged in the form of a matrix $\boldsymbol{W}$ which is called the feature vector.

Step 5: Derivation of new dataset

Finally the new dataset with reduced dimensions is obtained as follows:

$$\boldsymbol{Y} = \boldsymbol{XW} \tag{3.5}$$

3.2.3. Properties of Principal Component

A principal component analysis (PCA) can be primarily characterized as a set of observed variables that are linearly related to each other, and each of them is assigned an optimal weight. These principal components constitute the output of PCA. Generally, the principal components can be less than or equal to the original variables, in number. In case of dimensionality reduction, the principal components are always lesser in number as compared to the original variables. These principal components possess some useful properties which are as follows:

1. Principal components (PCs) can be thought of as the linear combinations of the original variables. The weight assigned to each variable in this linear combination is in fact the eigen vector evaluated by the eigen decomposition of covariance matrix, which consequently fulfills the principle of least squares.
2. Each of these principal components (PCs) are orthogonal which means that they are uncorrelated to each other i.e. the correlation between any two variables comes out to be zero.
3. The first principal component has the maximum variance, which tends to decrease towards the last principal component. This variance value is proportional to the significance of each principal component, ones having the maximum variance are most significant whereas the ones with minimum variance are least significance. Due to this reason, in order to reduce the dimensionality we can omit the least significance principal components, however, these least significant PCs are sometimes effective in outlier detection, regression etc.

3.2.4. Role of PCA in Background Subtraction

In the proposed technique for Background Subtraction in videos, principal component analysis (PCA) is exploited for background modelling to reduce the dimensions of data. For a given video sequence, if we assemble its frames in the form of columns of a matrix *M*, then the stationary background in the frames will be the low rank component L_0 and the moving foreground objects will be captured by the sparse component S_0. Nevertheless, there are thousands or tens of thousands of pixels in each video frame and in turn each fragment of the video consists of hundreds or thousands of frames. Therefore, such a decomposition of matrix *M* into a low-rank matrix L_0 and a sparse matrix S_0 would only be possible if we have some truly scalable solution for this problem.

3.3. Low Rank Matrix Factorization (LRMF) using CWM

In data science, low rank matrix factorization is considered to be a significantly important technique. The main idea of matrix factorization is that sometimes the data contains latent structures by uncovering which a compressed representation of the data can be obtained. Matrix factorization provides a unified method for dimensionality reduction, matrix completion and clustering by factorization of the original data matrix into low rank matrices.

Some important properties of matrix factorization are as follows:

1. In order to address the problem of data sparsity, matrix factorization uncovers the latent structures present in the data [37].
2. The probabilistic interpretation of matrix factorization is quite simple and effective [38].
3. Matrix factorization is well suited for several real world problems as it can be extended quite easily provided that prior knowledge specific to the domain is available e.g. homophily in linked data [39].
4. In order to find a good solution, several optimization methods such as stochastic methods e.g. gradient-based methods can be applied.

3.3.1. What is Low Rank Matrix Factorization (LRMF)?

Several computer vision and machine learning problems can be formulated as the problems that aim to extract the intrinsic low dimensional subspace from the high dimensional input data. This subspace thus extracted, has the tendency to provide the refined latent information underlying the data, therefore, it has extensive application range in information retrieval [40], object detection [41], structure from motion [42], plane based pose estimation[43], layer extraction [44], collaborative filtering [45], face recognition [46],[47] and social networks [46] etc. For subspace learning, one of the most commonly employed technique is low rank matrix factorization (LRMF).

Furthermore, if the data lacks missing entries, Singular Value Decomposition (SVD) based efficient algorithms can possibly be utilized but the inadequacy of such algorithms is that they are unable to take into account the outliers that occur commonly in the realistic data sets. This drawback arises because of the fact that these algorithms are based on the least square estimation techniques. Moreover, in some applications it is possible that the data set contains missing entries because of several reason such as analog to digital converter errors[42], tracking failure [41] or faulty memory locations in hardware [48]. Therefore, L_1-norm low rank matrix factorization (L_1-LRMF) can also be used, in order to deal with such incomplete or corrupted datasets.

A generalized L_1-norm LRMF problem can be formulated as follows. Let **X** be the data matrix such that $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ where $\mathbf{X} \in \mathrm{R}^{dxn}$, d represents the dimensionality and n represents the number of data elements. Every column of matirx **X** i.e x_i corresponds to a d-dimensional measurement. The data matrix **X** contains some missing entries which are represented by an indicator matrix **W**, where $\mathbf{W} \in \mathrm{R}^{dxn}$. The elements of **W** i.e w_{ij} are taken in such a way that it is zero when the corresponding element is missing and one otherwise [49]. Given **X** and **W** , it is possible to formulate a general LRMF problem as:

$$min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{W} \odot (\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T)\|_{L_1} \tag{3.6}$$

where **U** and **V** denote the basis and coefficient matrices respectively. Furthermore, $\mathbf{U} = [u_1, u_2, \ldots, u_k]$ and $\mathbf{V} = [v_1, v_2, \ldots, v_k]$ where $\mathbf{U} \in \mathrm{R}^{d \times r}$ and $\mathbf{V} \in \mathrm{R}^{n \times r}$, with $r << min(d,n)$, here ʘ symbolizes the Hadamard product i.e. component- wise multiplication, different from common matrix product. Here $r << min(d,n)$ basically indicates the property of low rank for $\boldsymbol{U}\boldsymbol{V}^T$. Unfortunately, it is somehow difficult to solve the above mentioned L_1-norm minimization because of two reasons. In general, the optimization of LRMF problem is non-convex which generally makes it a bit difficult task to find out a global minimum and in the presence of missing entries it is even found that it becomes an NP hard problem [50]. On the other hand, standard optimization tools can hardly find an effective closed form iteration formula [51] because L_1-norm minimization is non-smooth.

Various contemporary approaches use the variants of Wiberg method [51],[52],[53] in order to solve the problem of L_1-norm LRMF. These general purpose methods are, however, inefficient to reach the minimum often requiring too much cost, particularly for the real world high dimensional data. In the proposed technique, we have made use of the simple cyclic coordinate descent algorithm [54] in cyclic weighted median (CWM) method which shows an outstanding performance on L_1-norm LRMF.

3.3.2. LRMF using Cyclic Weighted Median (CWM) Method

In the proposed work, the L_1-norm low-rank matrix factorization (LRMF) problem is solved using a cyclic weighted median (CWM) technique which is constructed on the basis of coordinate descent algorithm.

The core idea of the cyclic coordinate descent algorithm is to split the fundamental complex minimization problem into a sequence of simple basic sub-problems. Every subproblem, having only one scalar parameter, is then recursively optimized. Being convex optimization problems, each of them can be readily solved with the help of weighted median filter, which eliminates the need of the time consuming inner loops for numerical optimization. Moreover the recursive employment of weighted median filter further makes the method robust to the missing entries as well as the outliers to a large extent. Through experimental results, it has been perceived that our proposed CWM method has improved efficiency compared to the other methods. It reduces the computational complexity for solving L_1-norm LRMF problem as compared to the other methods by decreasing the computation speed from $O(dn)$ to $O(d+n)$ for the input data matrices having high degree of sparsity. This makes it principally effective for the real-time problem solving having large as well as sparse data sets.

Coordinate descent can be an effective algorithm for the minimization of non-smooth convex functions. In multivariate minimization, coordinate descent method solves a sequence of scalar minimization subproblems in order to optimize the objective function. Therefore, in the proposed technique, coordinate descent algorithm is used to minimize the subproblems quickly. In order to improve the estimated solution, each subproblem minimizes along a single chosen coordinate keeping rest of the coordinates fixed. This technique can be considered analogous to Gauss-Seidel iterative algorithm that is used for obtaining solution for linear system of equations [55]. Furthermore, weighted median filter is used to solve each of the cyclic coordinate descent subproblem and hence giving the name cyclic weighted median (CWM) method.

3.3.2.1. CWM Algorithm for solving L_1- Norm LRMF Problem

The core idea of CWM algorithm, in order to solve the minimization in equation (3.6), is to apply recursively the weighted median filter in order to update every single element of **U** = [$u_1,u_2,\ldots,u_k$] and of **V** = [$v_1,v_2,\ldots,v_k$] where **U** $\in$ $R^{d\,x\,r}$ and **V** $\in$ $R^{n\,x\,r}$. In general, the steps involved in the algorithm are as follows:

Step 1:

In order to update each element v_{ij} of **V** where ($i=1,\ldots,k$) and ($j=1,\ldots,n$), the weighted median filter is applied cyclically while keeping rest of the components of **U** and **V** fixed. This is done by minimizing:

$$v_{ij}{}^{*} = argmin_{v_{ij}}\left\|\boldsymbol{w}_j\odot\boldsymbol{e}_j^i - \boldsymbol{w}_j\odot\boldsymbol{u}_i v_{ij}\right\|_{L_1} \tag{3.7}$$

where w_j represents j-th column vector of **W** and $e_j{}^i$ represents j-th column vector of $\mathbf{E}_i$ which is obtained as:

$$\boldsymbol{E}_i = \boldsymbol{X} - \sum_{j \neq i} \boldsymbol{u}_j \boldsymbol{v}_j^{\,T} \tag{3.8}$$

Step 2:

In the next step, apply cyclically the weighted median filter in order to update each element u_{ij} of **U** keeping rest of the components of **U** and **V** fixed. This can be done by solving the following minimization problem

$$u_{ij}^{\,*} = argmin_{u_{ij}} \left\| \widetilde{\boldsymbol{w}}_j \odot \widetilde{\boldsymbol{e}}_j^{i} - \widetilde{\boldsymbol{w}}_j \odot \boldsymbol{v}_i u_{ij} \right\|_{L_1} \tag{3.9}$$

where $\widetilde{\boldsymbol{w}}_j$ represents to the j-th row vector of **W** while $\tilde{e}_j{}^{i}$ represents the j-th row vector of $\mathbf{E}_i$.

The **U** and **V** matrices can be recurrently updated via iterative implementation of the above procedures till the fulfillment of the termination constraint.

In first step of algorithm, the initial values of **U** and **V** are obtained from PCA, performed prior to the L_1-LRMF. For the termination condition, as in the iteration process the objective function of equation (3.6) is decreasing monotonically, therefore the algorithm will terminate either when updating rate of **U** and **V** is less than some specified threshold or when the maximum iterations have been achieved.

3.4. Mixture Model

3.4.1. What is a Mixture Model?

A mixture model is described in statistics as a probabilistic model used to represent the occurrence of smaller subpopulations within an entire population, devoid of the need that an observed set of data must recognize, for each individual observations, the subpopulation in which it fits best. More appropriately, a mixture model refers to a mixture of distributions characterizing the probability distribution for the observations in the overall population. The problem related to mixture of distributions is that it needs to derive the properties for the overall population from the properties of subpopulations. In mixture models, this problem is resolved by the use of statistical inferences about the subpopulation properties, when only the observations on the pooled population are available and no subpopulation identity information is there.

Some of the methods for executing mixture models comprise of steps that characterize hypothesized identities for subpopulations for each individual observation. These methods can be considered as types of clustering or unsupervised learning procedures. Nevertheless, such steps are not involved in all the inference methods.

Unsupervised learning or clustering has gained interest owing to its emerging exploitation in numerous fields for instance biology, physics, astronomy, social sciences etc. The primary objective of cluster analysis is to ascertain the internal construction of clustered data, in case when the only data available are the observed values.

Generally heuristic or distance-based methods for instance iterative relocation procedures or hierarchical agglomerative clustering, are used for clustering which have two main advantages, their intuitive construction and their reasonable computational time but they have certain limitations because of lack of statistical basis. Due to lack of this statistical basis, the heuristic procedures can hardly handle the classical clustering questions (such as the number of clusters) theoretically. A prime alternate to heuristic-based algorithms are clustering methods that are based on probability models. In the framework of probability models that form basis for clustering methods, the data is regarded to come from a mixture of probability distributions, each of which represents a separate cluster.

Finite mixtures of distributions find their applications in an extensive range of statistical problems apart from clustering, including image analysis, survival analysis, discriminant analysis etc. To this scope, the mixture models have kept on getting expanding consideration from both theoretical as well as practical perspective.

3.4.2. What is Gaussian Mixture Model?

Basically a Gaussian mixture model (GMM) represents a distribution which is assembled from weighted multivariate Gaussian distributions. The weight assigned to each distribution refers to the level of importance given to that distribution in the overall model. Thus the resulting model is the superposition of the bell shaped curves of the individual Gaussian distributions.

In order to get the in depth sight of Gaussian Mixture model, firstly there is a need to understand Gaussian Distribution and Multivariate Gaussian Distribution. In the subsequent section, we will discuss these distributions in detail so that the Gaussian Mixture Model is better understood.

3.4.2.1. Gaussian (Normal) Distribution

A Gaussian distribution, which is also known as Normal distribution, is the most extensively known and used of all the probability distributions. For various probability problems, it is considered as a standard of reference because of its ability to approximate many natural phenomena so well.

Gaussian Distribution is useful because of the Central Limit Theorem. According to the Central Limit Theorem, under certain conditions, even if the random variables, that are independent, do not have a Gaussian (or normal) distribution, their appropriately normalized sum tends to have a normal distribution. In probability theory, this theorem is an important concept as it infers that the statistical and probabilistic approaches that are applicable for normal distributions can work well for various problems that involve other types of distributions.

i. **Univariate Normal Distribution:**

It describes a probability density function of a single continuous random variable which has a bell shaped curve and is symmetric as shown in Fig. 3.4.

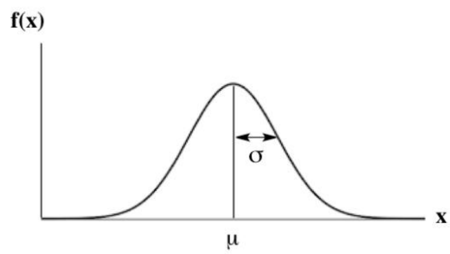


Figure 3.4: Gaussian Distribution

The probability density function of a univariate Gaussian distribution is given as:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{3.10}$$

where

μ = Expectation or mean value of the distribution

σ = Standard deviation

σ^2 = Variance

Therefore, a Gaussian distribution can be effectively described by two parameters:

a) μ = Mean of the distribution

b) σ = Standard Deviation of the distribution

The position and shape of the probability density function can be altered by changing these two parameters.

ii. Bivariate Normal Distribution:

Bivariate means involving two variables. A bivariate normal distribution is the one which involves two random variables that are independent, each having a normal distribution. Furthermore, when these two random variables are added together, the resultant distribution will also be a normal distribution having a three dimensional bell shaped curve as shown in Fig. 3.5.

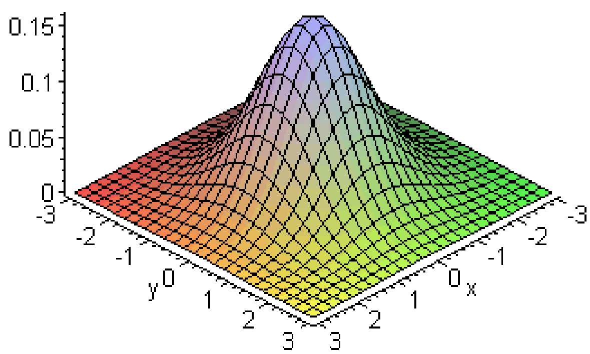


Figure 3.5: Bivariate Normal Distribution for two independent random variables ‘x’ and ‘y’

The probability density function of a two dimensional vector $X = \begin{bmatrix} x \\ y \end{bmatrix}$,where x and y represent the two independent random variables, is defined as:

$$p(x, y) = \frac{1}{(2\pi)|\Sigma|^{\frac{1}{2}}} e^{\left(-\frac{1}{2}\right)\left[\left(\frac{x-\mu_x}{\sigma_x}\right)^2 + \left(\frac{y-\mu_y}{\sigma_y}\right)^2\right]} \tag{3.11}$$

where $\mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}$ represents the mean vector and $\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix}$ represents the covariance matrix.

Covariance matrix represents the covariance among the elements of the two random variable vectors. The element at the (*i*,*j*) position of the matrix gives the covariance among the *i*-th and the *j*-th element of the random vector. Each element on the covariance matrix's principal diagonal represents one of the random variable's variance since covariance of a random variable with itself corresponds to its variance. Furthermore, each covariance matrix is symmetric as the covariance of the *i*-th random variable with the *j*-th one is identical to the covariance of the *j*-th random variable with the *i*-th one.

iii. Multivariate Normal Distribution:

A Multivariate normal distribution refers to a vector of multiple normally distributed variables, and any linear combination of these variables will have a normal distribution as well. Its usefulness is not restricted to extending the Central Limit theorem to several variables, but also in Bayesian inference and hence in machine learning, where it can be used for the approximation of features for some characteristics such as for face detection in images.

A Multivariate normal distribution defines the joint distribution of a random vector, the components of which are univariate, mutually independent, normal random variables with zero mean and unit variance. It is a generalization of the univariate normal distribution involving two or more than two variables. According to an alternative definition, a random vector is k-variate provided that any linear combination of its k-components gives a univariate normal distribution.

As the Multivariate normal distribution involves two or more random variables, the bivariate normal distribution can be considered as a particular case of the Multivariate normal distribution. The Bivariate case of a Multivariate normal distribution is easy to visualize but it is difficult to visualize it for the case of more than two variables.

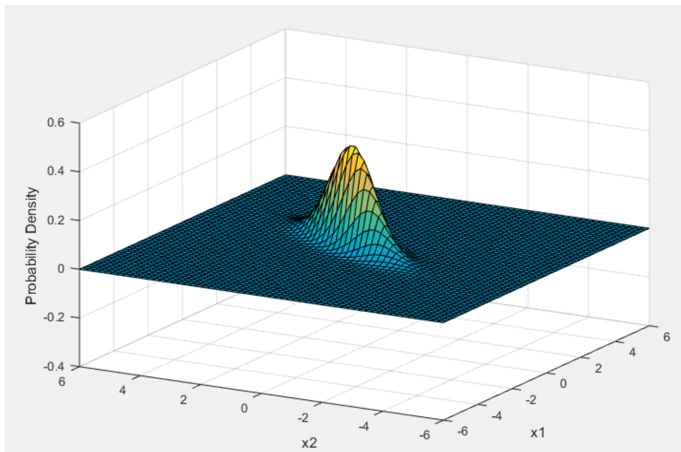


Figure 3.6: Multivariate Normal Distribution involving two variables

For a random vector $\boldsymbol{x} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ X_n \end{bmatrix}$ which is Multivariate normal , the probability density function is

given as:

$$p(\boldsymbol{x}; \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \, e^{\left(-\frac{1}{2}(\boldsymbol{x}-\mu)^T \Sigma^{-1}(\boldsymbol{x}-\mu)\right)} \tag{3.12}$$

μ represents the Mean vector which is D-dimensional whereas Σ represents the Covariance matrix having DxD dimension. The covariance matrix for the random vector $\boldsymbol{x}$ is described as:

$$\Sigma = \begin{bmatrix} E[(X_1-\mu_1)\,(X_1-\mu_1)] & \cdots & E[(X_1-\mu_1)\,(X_n-\mu_n)] \\ \vdots & \ddots & \vdots \\ E[(X_n-\mu_n)\,(X_1-\mu_1)] & \cdots & E[(X_n-\mu_n)\,(X_n-\mu_n)] \end{bmatrix} \tag{3.13}$$

where every (i,j) element of Σ corresponds to the covariance of the two random variables i.e.

$$\Sigma_{i,j} = cov\left(X_i, X_j\right) = E\left[(X_i-\mu_i)\left(X_j-\mu_j\right)\right] \tag{3.14}$$

where $\mu_i = E(X_i)$

Now-a-days, in machine learning, multivariate normal distribution is of incredible importance. The main aim of machine learning is the classification of input data into labels, given some pairs of training data. The foremost approach for doing this is to analyze the distribution and approximate it by a multivariate normal distribution. Moreover, to check the validity of this approximation, several normality tests can be used. Classification on the basis of multivariate distribution proves to be quite effective practically, even when it is known to be not a good model for the data.

3.4.3. Gaussian Mixture Model

In general, Gaussian mixture model (GMM) represents a distribution which is assembled from weighted multivariate Gaussian distributions. The weight assigned to each distribution refers to the level of importance given to that distribution in the overall model. Thus the resulting model is the superposition of the bell shaped curves of the individual Gaussian distributions.

Gaussian Mixture Model (GMM) is a probabilistic model with the assumption that every point in a dataset is created from a mixture of distributions having finite number of Gaussians with unknown parameters. Generally any 1-D,2-D and 3-D data set is considered to have a Gaussian distribution because of its intimacy to natural distribution and because of the ease of doing mathematical manipulation with Gaussian distribution. However, there are certain situations in which the distribution is not strictly Gaussian, rather the data sets have clusters in their structure as shown in the Fig. 3.7.

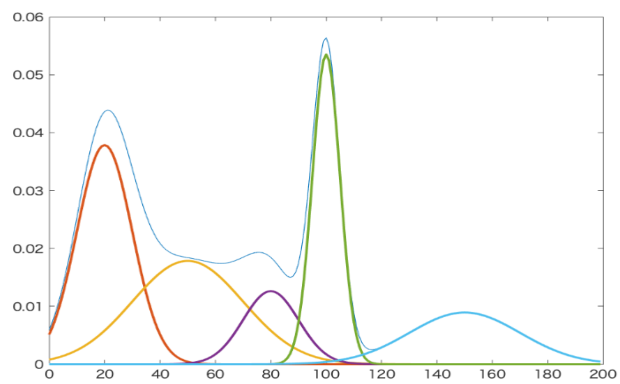


Figure 3.7: Mixture of Gaussian Distributions showing five Gaussians and their sum

Such data sets can be effectively described by the linear superposition of Gaussians because it is difficult to characterize them by a single Gaussian distribution. By the use of appropriate number of Gaussians and careful adjustment of their parameters i.e means, covariance and weights assigned to each distribution, it is possible to accurately approximate any continuous distribution. Therefore, the superposition of K Gaussian distributions is defined as :

$$p(x) \ = \ \sum_{k=1}^{K} \pi_k \, N(x|\mu_k \Sigma_k) \tag{3.15}$$

This is called Gaussian Mixture model ,the three dimensional view of which is shown in Fig. 3.8.

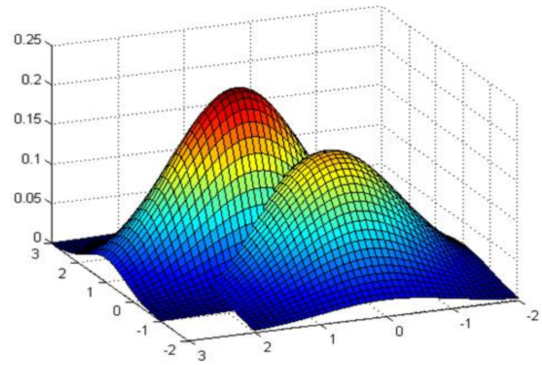


Figure 3.8: A Gaussian Mixture Model having two Gaussian distirbutions.

3.4.3.1. The Model

Generally a Gaussian mixture model (GMM) can be completely described by three parameters, the means and covariances of each mixture component and the weight assigned to it. For the univariate case of GMM, each *kth* element has mean μ_k and variance σ_k whereas in the multivariate context, each *kth* component has a mean vector $\boldsymbol{\mu_k}$ along with a covariance matrix $\boldsymbol{\Sigma_k}$. For each component C_k, the mixture weights are defined as π_k having constraint that $\sum_{i=1}^{K} \pi_i = 1$ in order to normalize the total probability distribution to 1. These weights can be regarded as an a-priori distribution over the components if they are not learned. On the other hand, if the component weights are learned instead, they are considered to be the a-posteriori estimates of the component probabilities.

3.4.3.2. One Dimensional Model

$$p(x) = \sum_{i=1}^{K} \emptyset_i \, N(x|\mu_i, \sigma_i) \tag{3.16}$$

$$N(x|\mu_i, \sigma_i) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{\left(-\frac{(x-\mu_i)^2}{2\sigma_i{}^2}\right)} \tag{3.17}$$

$$\sum_{i=1}^{K} \pi_i = 1 \tag{3.18}$$

3.4.3.3. Multi-dimensional Model

$$p(\boldsymbol{x}) = \sum_{i=1}^{K} \pi_i \, N(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) \tag{3.19}$$

$$N(\boldsymbol{x}|\boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) = \frac{1}{\sqrt{(2\pi)^K |\boldsymbol{\Sigma_i}|}} \, e^{\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_i})\right)} \tag{3.20}$$

$$\sum_{i=1}^{K} \pi_i = 1 \tag{3.21}$$

3.4.3.4. Learning the Model

Once we have defined the model, we need to evaluate the parameters of mixture model i.e. the mean, covariance and the component weights. Expectation maximization (EM) is the frequently utilized technique for this purpose, provided that the number of mixture components ‘K’ is known. In accordance with the frequentist probability theory, generally the maximum likelihood estimation techniques can be employed for learning of models. Given the model parameters, such techniques tend to maximize the probability or the likelihood of the observed data. Unfortunately, it is analytically impossible to find out the solution of maximum likelihood for the mixture models via differentiation of log-likelihood and then evaluate it for 0.

Expectation maximization (EM) is an algorithm that iteratively approximates the maximum likelihood (ML) of the parameters of mixture model. Alternatively, we might think of it as an estimation problem for the data sets that involve unobserved latent variables or some missing data points or the data is somewhat incomplete. For such data sets, it can approximate the maximum likelihood estimates for the model parameters iteratively. One very useful property of EM algorithm is that for each subsequent iteration, there is a strict increase in the maximum likelihood of the data, which implies that it will certainly approach a saddle point or local maximum.

3.4.3.5. Expectation Maximization (EM) Algorithm

Expectation maximization (EM) algorithm, like the name suggests, essentially involves two steps.

i. The first step is the E-step or the Expectation step. In this step, for every data point $x_i \epsilon X$, for the given model parameters μ_k, σ_k and π_k, the expectation of the component assignment C_k is calculated.
ii. The second step is the M-step or the Maximization step. In this step, the expectations estimated in the first step are maximized considering the model parameters. Furthurmore, the values of model parameters μ_k, σ_k and π_k are also updated in this step.

These steps will be repeated iteratively until the algorithm converges, giving the maximum likelihood estimate. Intuitively, the effectiveness of the algorithm is due to the fact that if we know the component assignment C_k , for each x_i, it makes it easy to solve for the parameters μ_k, σ_k and π_k. This corresponds to the M-step. Furthermore, having known the parameters μ_k, σ_k and π_k, it becomes easy to infer $p(C_k|x_i)$, which is the E-step. Therefore, by alternating between which values are assumed to be fixed and which are known, the maximum likelihood estimate of the varying values can be determined efficiently.

For the Gaussian mixture models, the expectation maximization (EM) algorithm begins with an initilaization step, which assigns appropriate values to the model parameters based on the data. The model then iterates over the expectation and maximization steps until the convergence of the parameters' estimates i.e. at iteration *t* for all parameters θ_t , $|\theta_t - \theta_{t-1}| \leq \epsilon$ where ϵ is some user-defined tolerance.

3.4.3.5.1. Basic Steps

1. Initialize the model parameters, that is, means μ_k, covariances Σ_k and the mixture weights π_k where k=1,2,…,K. Evaluate the initial value of log likelihood.
2. After initialization, the second step is the expectation step. In this step, using the current parameters, the responsibilities are evaluated.

$$\gamma_k(x) = \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N\left(x\middle|\mu_j, \Sigma_j\right)} \tag{3.22}$$

   where $\gamma_k(x)$ corresponds to the latent (hidden) variable for the k^{th} Gaussian.

3. The next step is the maximization step. This step involves the recalculation of the model parameters by using the currently obtained values. This recalculation is done using the following equations.

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)} \tag{3.23}$$

$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n)(x_n - \mu_j)(x_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)} \tag{3.24}$$

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j\,(x_n) \tag{3.25}$$

4. Finally evaluate the log-likelihood.

$$ln\ \ p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} ln\{\sum_{k=1}^{N} \pi_k\, N(x_n|\mu_k, \Sigma_k)\} \tag{3.26}$$

   If either the parameters or the log-likelihood has converged, it means that the desired results have been achieved, if not, then return to step-2 and iterate until convergence.

3.4.3.6. Unsupervised Learning:

After the convergence of the expectation maximization (EM) algorithm, the fitted model obtained can be utilized to carry out several types of inference. Mostly two common types of inference are made on GMMs, the density estimation and the clustering.

a) Density Estimation:

Since the GMM can be completely defined by just the parameters of each individual component Gaussian distribution, therefore, an estimate of the probability of both the available as well as the hiden or missing data points can be obtained from a fitted GMM. This is known as density estimation.

b) Clustering:

Using the model parameters estimated by the EM algorithm and the Bayes' theorem, one can also obtain an estimate of the posteriori component assignment probability. One way to learn clusters is knowing that given any two component distributions, the data point is likely to belong to which of the two. For a univariate model's parameters, Bayes' theorem can be used to determine the probability that a data point *x* belongs to the component C_i as follows:

$$p(C_i|x) = \frac{p(x, C_i)}{p(x)} \tag{3.27}$$

$$= \frac{p(C_i)p(x|C_i)}{\sum_{j=1}^{K} p(C_j)p(x|C_j)}$$

$$= \frac{\pi_i N(x|\mu_i, \sigma_i)}{\sum_{j=1}^{K} \pi_j N(x|\mu_j, \sigma_j)}$$

The cluster assignment is detemined by the most likely cluster assignment. In machine learning, clustering has a variety of applications from medical imaging for tissue differentiation to market research for customer segmentation.

3.4.4. Background Subtraction using Gaussian Mixture Model (GMM)

Uptill now various techniques have been proposed for background subtraction in videos but some of the current prevelent techniques show obvious defects when they are utilized for real time videos. Majority of the background subtraction methods consider a fixed loss term in their models such as L_1 or L_2 norm losses, which leads to an implicit assumption that the noises i.e. foreground in the videos have a predetermined probability distribution such as Gaussian or Laplacian. Though, in real scenarios this assumption is not true because the foreground always show obvious variations over time. It is possible that for some of the frames, no foreground objects are present. For such frames, the noises can be effectively modelled by a Gaussian distribution i.e. L2 norm loss. For the frames having a foreground object that is obstructing a considerable portion of background, the noises can be effectively modelled by extended tail Laplacian i.e. L1 norm loss. Furthermore, in some situations, multiple modalities of noise might be conatined in the forground, which requires the consideration of more intricate noise models. Ignorance of such significant insight of video foreground variations makes the current methods not sufficiently robust to finely adapt the variations in real time foreground or noise.

In the proposed technique, rather than using a fixed distribution of noise for entire set of video frames, for each frame the noise or foreground is modelled as a separate mixture of Gaussian (MoG) distributions, which is regularized by a penalty that enforces its parameters close to the one calculated from the prior frames. Such penalty can also be reformulated as the conjugate prior for the Mixture of Gaussian (MoG) of the current frame, by encoding the knowledge of noise learned previously. As the Mixture of Gaussians (MoG) has the ability to effectively approximate to a wide variety of distributions, the proposed method has the ability to finely adapt to variations in the video foreground, even for the video noises with complex dynamic structures.

3.5. Summary

In this chapter, the proposed technique for background subtraction in videos was explained with all the mathematical modelling involved at each step. In the next chapter, the results obtained by employing this technique on different video datasets will be discussed and it will be shown both quantitatively and qualitatively that how the proposed technique is efficient as well as accurate in comparison to the techniques discussed in the literature survey in chapter 2.

Chapter 4

RESULTS AND DISCUSSION

This chapter comprises of the performance evaluation of the proposed technique by applying it on different video datasets having static and dynamic backgrounds.

4.1. Performance Evaluation

The quantitative metric used to benchmark the performance of foreground detection is the F-Measure. It basically gathers the scores of recall and precision and is defined as:

$$F - measure = 2 \times \frac{Precision\ .\ Recall}{Precision + Recall} \tag{4.1}$$

Recall is a measure that tells how many of the true positives are identified correctly. Mathematically recall is defined as:

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

On the other hand, precision tells us that out of all the positive classifications, how many are actually correct. It can be defined mathematically as:

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

In the above equations,

$$TP = True\ Positive: Correctly\ assigned\ as\ positive$$

$$TN = True\ Negative: Correctly\ assigned\ as\ negative$$

$$FP = False\ Positive: Wrongly\ assigned\ as\ positive$$

$$FN = False\ Negative: Wrongly\ assigned\ as\ negative$$

The value of precision and recall always lies between 0 and 1. A recall value of 1 indicates that no false negative classification is there. Similarly if the precision value is 1, it is an indication that are no false positives are there, that is, all the positive samples are classified as positive without any missclassification.

4.2. Simulation Results

For illustrating the efficiency of our proposed technique, it has been applied to three different datasets, i.e.

a. Airport video without camer jitter effect

b. Synthetically transformed airport video with camera jitter effect

c. An unaligned face with different illuminations

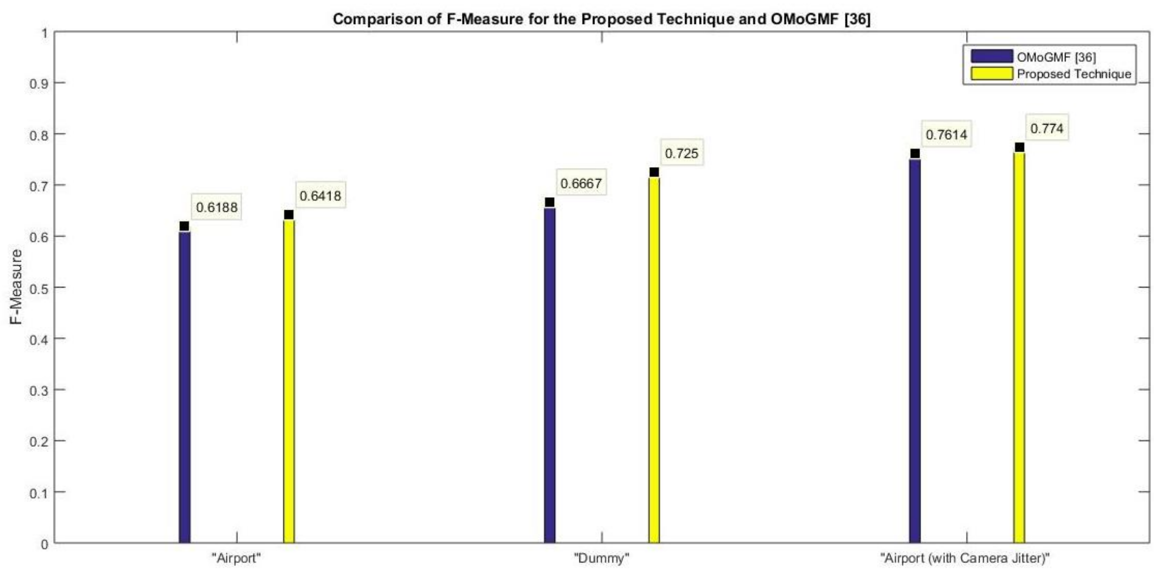


Figure 4.1: Graphical Comparison of F-Measure of OMoGMF [36] and Proposed Technique

The qualitative analysis is performed by comparing the results obtained by our proposed technique and OMoGMF method proposed in [36] as shown in the Table 4.1. A graphical comparison of these F-Measure results is illustrated in Figure 4.1.

Table 4.1. Comparison of F-measure of OMoGMF [36] and Proposed Technique

Dataset	**OMoGMF [36]**	**Proposed Technique**
Airport	0.6188	0.6418
Dummy	0.6667	0.7250
Airport with Camera Jitter	0.7614	0.7740

In these experiments, we have taken into account the two major problems which were not resolved in previously proposed methods. One being the dynamic background that is illumination changes, waving trees, ripples in water etc. and second is the camera jitter effects like translation, rotation, scaling or a combination of all these , thereby making our proposed technique more robust for dynamic background changes. The obtained results are shown in Figure 4.2.

OMoGMF [36] **Proposed Technique**

Figure 4.2: *Airport* sequence : first row (left to right) Original Frame, Extracted Background, Foreground (Residuals). Second row (left to right) the three gaussian noise components extracted by the method.

To exemplify the effectiveness of our technique for the camera jitter effects, we took the same *Airport* sequence as shown in Fig. 4.2 but this time it has camera jitter effects in it. Each successive frame is either rotated or translated by a certain amount as compared to the previous frame. The results of applying the proposed and OMoGMF [36] background subtraction algorithm on such a video sequence are shown in Figure 4.3.

OMoGMF [36]

Proposed Technique

Figure 4.3: *Airport* sequence with Camera Jitter Effect. The frames are slightly rotated as compared to the original video sequence shown in Figure 4.1 of Airport Sequence.

Under certain low-rank assumption, it is possible to reconstruct a larger matrix of low-rank from a fewer number of its elements [56]. Stimulated by some previous efforts [57] on this issue, the efficiency of the proposed method is further improved by appending the subsampling technique as used in [36]. The introduction of subsampling technique can accelerate the execution process to, on average, more than 250 frames per second. Furthermore, this add on does not affect the accuracy of the proposed method, i.e. a good performance in accuracy is maintained. In these experiments the subsampling rate is The results obtained by using the subsampling technique on the four video sequences are shown in Figure 4.4 and Figure 4.5.

OMoGMF [36]

Proposed Technique

Figure 4.4: *Airport* sequence : Using subsampling (Subsampling rate =0.01) From left to right, Original Frame, Extracted Background, Foreground (Residuals).

OMoGMF [36]

Proposed Technique

Figure 4.5: *Airport* sequence with camera jitter effect : Using subsampling (Subsampling rate =0.01) From left to right, Original Frame, Extracted Background, Foreground (Residuals).

In order to show the effectiveness of our proposed alignment approach using affine transformation, we have tested the proposed approach for aligning the frames of "Dummy" dataset shown in Figure

4.6. This dataset contains multiple images that are not only misaligned but also suffer from illumination variation and block occlusion.

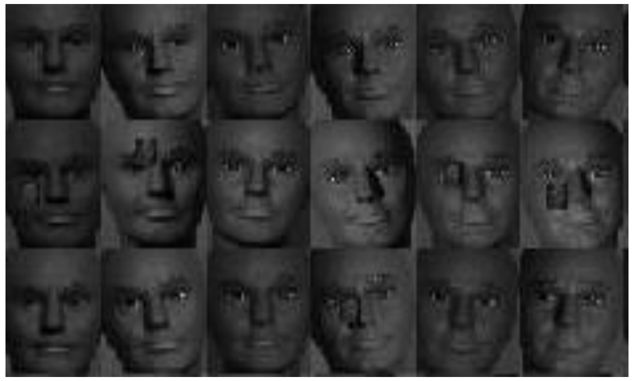
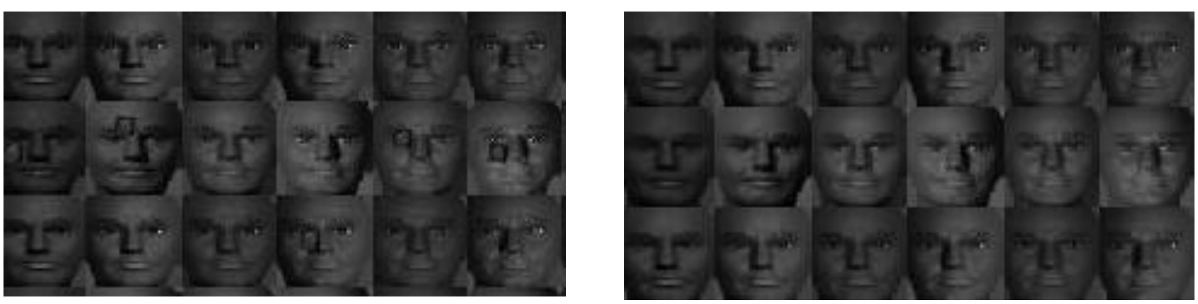


Figure 4.6: Original images in “Dummy” dataset containing misaligned images with block occlusion and illumination variation

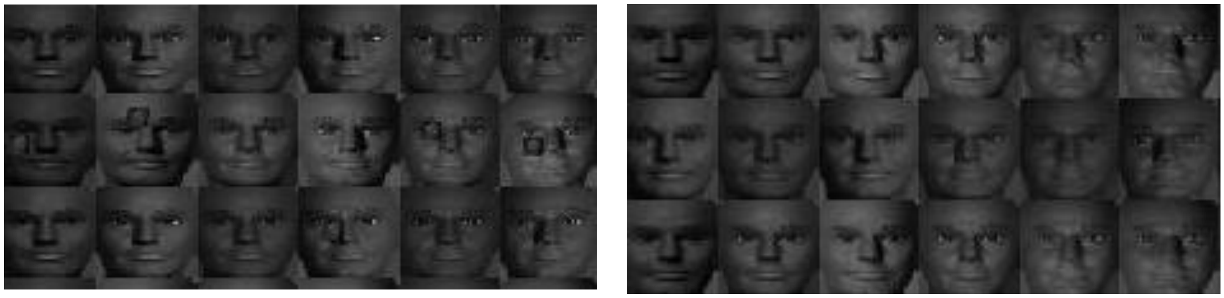
The results obtained by applying the technique proposed in [36] are shown in Figure 4.7 (a) and (b). It can be seen that in some images, the face has become enlarged during the alignment process. Furthermore, there are more darker images which means that the effect of illumination variation has not been removed effectively.



(a) (b)

Figure 4.7: Results obtained by OMoGMF [36] (a) Aligned Frames; (b) Frames with block occlusion removed and illumination effect minimized

As compared to the technique proposed in [36], the proposed technique can better align the images and minimizes the illumination variation much better as shown in Figure 4.8 (a) and (b).



(a) (b)

Figure 4.8: Results obtained by proposed technique (a) Aligned Frames; (b) Frames with block occlusion removed and illumination effect minimized

4.3. Computational Complexity

Compared to the other state-of-the-art background subtraction methods, our proposed approach has comparatively reduced computational complexity. This is basically due to the Cyclic Weighted Median (CWM) method that we have employed for the low rank matix factorization (LRMF) using L_1- norm. If we take n and d as the size and dimensionality of the input data matrix, respectively, then the proposed technique has a computational complexity of the order $O(d+n)$ whereas that of other state-of-the-art algorithms is $O(dn)$ [58] .

A comparison of computaional time of our proposed technique with that of OMoGMF [36] algorithm is given in Table 4.2 and also shown graphically in Figure 4.9 and Figure 4.10.

Table 4.2. Comparison of the Computational Time (in seconds) of OMoGMF [36] and Proposed Technique

Dataset	**OMoGMF [36] (Without subsampling)**	**OMoGMF [36] (With subsampling)**	**Proposed Technique (Without subsampling)**	**Proposed Technique (With Subsampling)**
Airport	14.28812	2.711615	13.884685	1.890548
Dummy	0.996848	0.543907	0.863588	0.204113
Airport (With camera jitter effect)	14.866191	3.232390	13.287818	2.246179

From Figure 4.9, it is obvious that the computational time of the our technique is reduced compared to the OMoGMF method [36]. For Airport dataset, it is reduced from 14.29 seconds to 13.88 seconds, from 0.9968 seconds to 0.8636 seconds for Dummy dataset and from 14.87 seconds to 13.29 seconds for airport dataset with camera jitter effects.

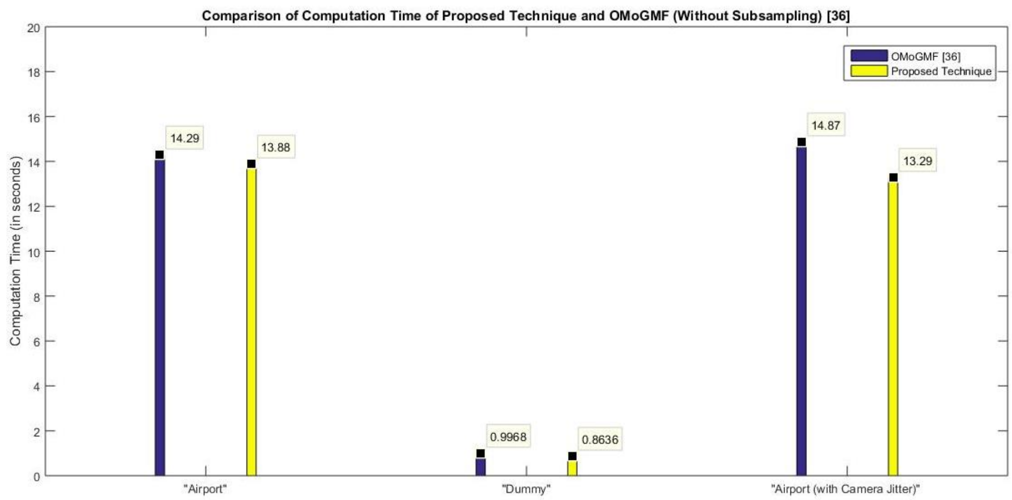


Figure 4.9: Graphical Comparison of Computation Time of the Proposed Technique and OMoGMF [36] without subsampling

For furthur enhancement of efficiency of our method, subsampling technique is embedded into the calculation. The subsampling rate is taken to be 0.01 i.e. 1%. This can accelerate the method to execute on an average more than 250 frames per second while maintaining a good performance accuracy. Figure 4.10 illustrates a comparison of computational time when the subsampling technique is used.

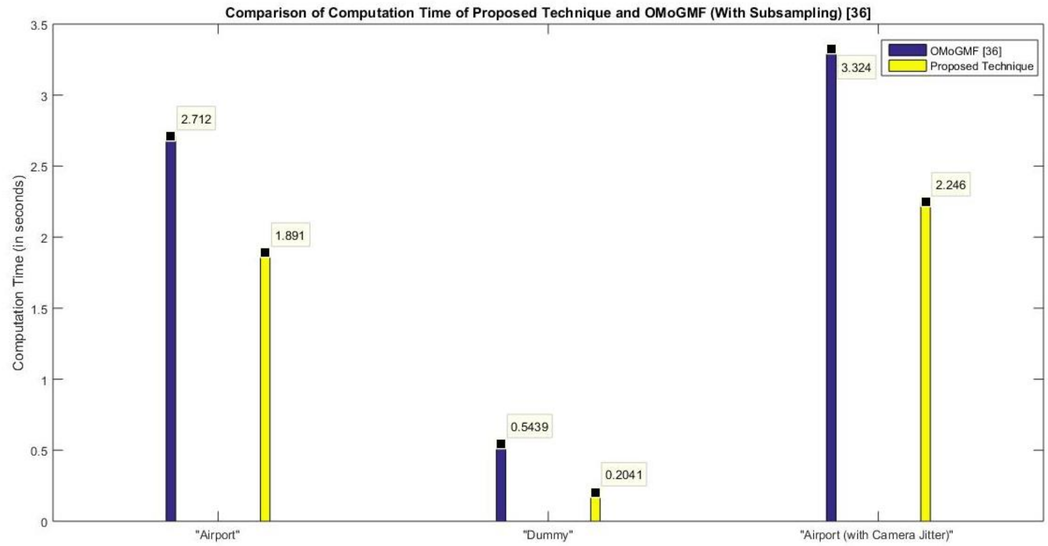


Figure 4.10: Graphical Comparison of Computation Time of the Proposed Technique and OMoGMF [36] with subsampling

From Figure 4.10, it can be seen that the computation time of our proposed appraoch has been comparatively reduced as that of OMoGMF [36], even with the subsampling technique.

4.4. Summary

In this chapter, the results acheived by applying our background subtraction technique on Li dataset were discussed, in terms of accuracy and computational complexity. It was deduced that our technique not only reduces the computational time as compared to the reference technqiue in [36] but also has better detection accuracy in terms of F-measure.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this research work, an efficient technique has been proposed which aims at making background subtraction available for real time videos both in terms of speed and accuracy. The proposed method gradually fits a specific subspace for the background that is obtained from L1-LRMF using CWM and a certain MoG distribution of noise used for the foreground. This fit is achieved by regularization of the background and foreground information acquired from the preceding frames. As opposed to the conventional methods which used a fixed noise distribution for the entire video frames, in this method a separate mixture of Gaussian (MoG) distribution is used to model the noise or foreground for each frame of the video. Gaussian Mixture Model (GMM) is optimized by exploiting the Expectation Maximization (EM) algorithm. For the elimination of camera jitter effects, affine transformation operator is involved that acts to align the successive frames. The efficiency of our background subtraction technique is augmented via subsampling technique that can accelerate the proposed method to execute on an average more than 250 frames per second while maintaining a good performance in accuracy.

5.2 Future Work Recommendations

Although the results obtained from the proposed approach are encouraging, still there is always space for enhancement in order to achieve better results. In the proposed approach, we have considered only gray scale videos. In future, it can be extended to give the desired results for RGB videos as well. Furthermore, all the videos taken under consideration in this work, are of short duration. In future, we can extend our methods for larger videos in real time.

BIBLIOGRAPHY

[1] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection-a systematic survey," *Recent Patents on Computer Science*, vol. 4, Issue 3, pp. 147–176, 2011.

[2] K. Toyama, J. Krumm, B. Brumitt and B. Meyers, "Wallflower: Principles and Practice of Background Maintenance," In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, page 255, Los Alamitos, CA, USA. IEEE Computer Society, 1999.

[3] S. Brutzer, B. Hoferlin and G. Heidemann, " Evaluation of Background Subtraction Techniques for Video Surveillance," In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1937–1944, Colorado Spring, USA, 2011.

[4] C. R. Wren, A. Azarbayejani, T. Darrell and A. P. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, Issue 7, pp. 780–785, 1997.

[5] N. J. B. McFarlane and C. P. Schofield, "Segmentation and Tracking of Piglets in Images," *Machine Vision and Applications*, vol. 8, Issue 3, pp. 187–193, 1995.

[6] C. Stauffer and W. E. L. Grimson , "Adaptive Background Mixture Models for Real-time Tracking," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, Fort Collins, CO, USA, 1999.

[7] T. Matsuyama, T. Ohya and H. Habe , " Background subtraction for non-stationary scenes," In *Asian Conference on Computer Vision*, pp. 622–667, 2000.

[8] M. Heikkilä, M. Pietikäinen and J. Heikkilä , "A texture-based method for detecting moving objects," In *Proccedings of the British Machine Vision Conference*, vol. 1, pp.187–196, 2004.

[9] S. S. Huang, L. C. Fu and P. Y. Hsiao, "A region-based background modeling and subtraction using partial directed hausdorff distance," In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 1, pp. 956 – 960, 2004.

[10] T. Yu, C. Zhang, Y. Yui, M. Cohen and Y. Wu, "Monocular video foreground/background segmentation by tracking spatial-color gaussian mixture models," In *Proceedings of the IEEE Workshop on Motion and Video Computing (WMVC'07)*, Austin, Texas, 2007.

[11] M. Piccardi , "Background Subtraction Techniques: A Review," *2004 IEEE International Conference on Systems, Man and Cybernetics, IEEE*, vol. 4, pp. 3099-3104, 2004.

[12] D. K. Yadav and K. Singh, " Motion-Based Object Detection in Real-Time Visual Surveillance System using Adaptive Learning," *Journal of Information Assurance and Security*, March, vol. 10, Issue 2, pp. 089-099, 2015.

[13] L. S. Sharma, D. K. Yadav and S. K. Bharti, " An improved method for visual surveillance using background subtraction technique," *2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN), IEEE*, pp. 421-426, 2015.

[14] P. L. Rosin, "Thresholding for change detection," *Computer Vision and Image Understanding*, vol. 86, Issue 2, pp. 79–95, 2002.

[15] M. Kilger , "A shadow handler in a video-based real-time traffic monitoring system," In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 11–18, 1992.

[16] R. Cucchiara, C. Grana, M. Piccardi and A. Prati, " Detecting moving objects, ghosts, and shadows in video streams," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, Issue 10, pp. 1337–1342, 2003.

[17] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic Approach," In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI),* 1997.

[18] E. Hayman and J. Eklundh, "Statistical Background Subtraction for a Mobile Observer," In *Proceedings of Ninth IEEE International Conference on Computer Vision (ICCV'03)*, pp. 67–74, 2003.

[19] A. Elgammal, D. Harwood and L. S. Davis, "Non-parametric model for background subtraction," In *Proceedings of the 6th European Conference on Computer Vision*, pp. 751-767, 2000.

[20] A. Elgammal, R. Duraiswami, D. Harwood and L. S. Davis, "Background and Foreground Modeling Using Non-parametric Kernel Density Estimation for Visual Surveillance," In *Proceedings of the IEEE*, vol. 90, Issue 7, pp. 1151–1163, 2002.

[21] B. Han, D. Comaniciu and L. Davis, "Sequential kernel density approximation through mode propagation: Applications to background modelling," In *Proceedings of the Asian Conference on Computer Vision*, vol. 1, 2004.

[22] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognition Letters*, vol. 27, Issue 7, pp.773–780, 2006.

[23] N. M. Oliver, B. Rosario and A. P. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol. 22, Issue 8, pp. 831 –843, 2000.

[24] B. Han and R. Jain, "Real-time subspace-based background modeling using multichannel data," In *Advances in Visual Computing*, vol. 4842 of *Lecture Notes in Computer Science*, pp. 162–172. Springer Berlin Heidelberg, 2007.

[25] G. Zhao, and M. Pietikainen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, Issue 6, pp. 915–928, 2007.

[26] S. Zhang, H. Yao and S. Liu, "Dynamic background modeling and subtraction using spatio-temporal local binary patterns," In *Proceedings of the 15th IEEE International Conference on Image Processing (ICIP)*, pp. 1556–1559, 2008.

[27] B. Wang, P. Dudek, "A Fast Self-tuning Background Subtraction Algorithm," *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* pp.401 - 404, 2014.

[28] D.K. Yadav, L. Sharma and S. K. Bharti, “Moving Object Detection in Real-Time Visual Surveillance using Background Subtraction Technique,” *2014 14th International Conference on IEEE Hybrid Intelligent Systems (HIS)*, pp.79 - 84, 2014.

[29] A. Boulmerka and M. S. Allili, “Background Modeling in Videos Revisited using Finite Mixtures of Generalized Gaussians and Spatial Information,” *2015 IEEE International Conference on Image Processing (ICIP)*, pp.3660 - 3664, 2015.

[30] S. Javed, S. Ho Oh, A. Sobral, T. Bouwmans and S. Ki Jung, “Background subtraction via superpixel-based online matrix decomposition with structured foreground constraints,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 90–98, 2015.

[31] S. Javed, A. Sobral, T. Bouwmans, and S. K. Jung, “OR-PCA with dynamic feature selection for robust background subtraction,” in *Proceedings of the 30th Annual ACM Symposium on Applied Computing. ACM*, , pp. 86–91, 2015.

[32] R. Chen and H. Li, “Online Algorithm for Foreground Detection Based on Incremental Nonnegative Matrix Factorization,” in *2016 The 2nd International Conference on Control, Automation and Robotics (ICCAR)*, pp. 312 - 317, 2016.

[33] S. E. Ebadi , V. G. Ones and E. Izquierdo , “Dynamic tree-structured sparse RPCA via column subset selection for background modeling and foreground detection,” in *2016 IEEE International Conference on Image Processing (ICIP), IEEE*, pp. 3972–3976, 2016.

[34] S. T. Ali, K. Goyal and J. Singhai, “Moving Object Detection Using Self Adaptive Gaussian Mixture Model for Real Time Applications,” *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, pp. 153 – 156, 2017.

[35] S. Javed , A. Mahmood, T. Bouwmans and S. K. Jung, “Spatiotemporal Low-Rank Modeling for Complex Scene Background Initialization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, Issue 6, pp. 1315-1329, 2018.

[36] H. Yong, D. Meng, W. Zuo and L. Zhang, “Robust Online Matrix Factorization for Dynamic Background Subtraction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, Issue 7, pp. 1726 – 1740, 2018.

[37] Y. Koren, “Factorization meets the neighborhood: a multifaceted collaborative filtering model,” In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining KDD’08*, pp. 426–434, 2008.

[38] A. Mnih and R. Salakhutdinov, “Probabilistic matrix factorization,” In *Proceedings of the 20th International Conference on Neural Information Processing Systems NIPS’07*, pp. 1257–1264, 2007.

[39] J. Tang, H. Gao, X. Hu and H. Liu, “Exploiting homophily effect for trust prediction,” In *Exploiting homophily effect for trust prediction WSDM’13*, pp. 53–62. ACM, 2013.

[40] S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American Society for Information Science*, vol.41, Number 6, pp. 391-407, 1990.

[41] M. Turk and A. Pentland, “Eigenfaces for recognition,” *Journal of Cognitive Neuroscience,* vol. 3, Issue 1, pp. 71–86, 1991.

[42] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: A factorization method,” *International Journal of Computer Vision (IJCV),* vol. 9, Issue pp. 2, 137–154, 1992.

[43] P. Sturm, “Algorithms for plane-based pose estimation,” In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000*, vol. 1, 2000.

[44] Q. Ke and T. Kanade, “Robust norm factorization in l1the presence of outliers and missing data by alternative convex programming,” In *IEEE Conference on Computer Vision and Pattern* (*CVPR)*, 2005.

[45] Y. Koren, “Factorization meets the neighborhood: A multifaceted collaborative filtering model,” In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (*SIGKDD)*, pp. 426-434, 2008.

[46] C. Cheng, H. Yang, I. King and M. R. Lyu, “Fused matrix factorization with geographical and social influence in location-based social networks,” In *26th AAAI Conference on Artificial Intelligence*, pp. 17-23, 2012.

[47] J. Wright, Y. Peng, Y. Ma, A. Ganesh and S. Rao, “Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization,” *In Advances in Neural Information Processing Systems (NIPS)*, 2009.

[48] H. Ji, C. Liu, Z. Shen and Y. Xu, “Robust video denoising using low rank matrix completion,” In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (*CVPR)*,pp. 1791-1798, 2010.

[49] A. M. Buchanan and A. W. Fitzgibbon, “Damped newton algorithms for matrix factorization with missing data,” In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 2, 2005.

[50] N. Gillis and F. Glineur, “Low-rank matrix approximation with weights or missing data is NP-hard,” *SIAM Journal of Matrix Analysis and Applications*, vol. 32, Issue 4, pp. 1149-1165, 2011.

[51] A. Eriksson and A. van den, “Efficient computation of robust low-rank matrix approximations in the presence of missing data using the norm,” In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR. l1)*, pp. 771-778, 2010.

[52] T. Okatani and K. Deguchi, “On the Wiberg algorithm for matrix factorization in the presence of missing components,” In *International Journal of Computer Vision IJCV*, vol. 72, Issue 3, pp. 329–337, 2007.

[53] D. Strelow, “General and nested Wiberg minimization,” In *ECCV'12 Proceedings of the 12th European conference on Computer Vision*, pp. 195-207, 2012.

[54] P. Tseng, “Convergence of a block coordinate descent method for nondifferentiable minimization,” *Journal of Optimization Theory and Applications*, vol. 109, Issue 3, pp. 475–494, 2001.

[55] H. Jeffreys and B. S. Jeffreys, “Methods of Mathematical Physics,” *Cambridge University Press*, 1988.

[56] E. J. Cand`es and B. Recht, “Exact matrix completion via convex optimization,” *Foundations of Computational mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[57] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 1568–1575, 2012.

[58] D. Meng, Z. Xu, L. Zhang, and J. Zhao, "A cyclic weighted median method for l1 low-rank matrix factorization with missing entries," In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence,* pp. 704-710, 2013.