Nonchronological Video Synopsis and Indexing

Ph.D. Thesis

Tapas Badal

(ID No. 2013RCP9004)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

MARCH, 2018

Nonchronological Video Synopsis and Indexing

submitted in

fulfillment of the requirements for the degree of

Doctor of Philosophy

by

Tapas Badal

ID: 2013RCP9004

Under the Supervision of

Dr. Neeta Nain

Dr. Mushtaq Ahmed

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR, INDIA

MARCH, 2018

CERTIFICATE

This is to certify that the thesis entitled, **"Nonchronological Video Synopsis and Indexing"** being submitted by **Tapas Badal (2013RCP9004)** is a bonafide research work carried out under our supervision and guidance in fulfillment of the requirement for the award of the degree of **Doctor of Philosophy** in the Department of Computer Science & Engineering, Malaviya National Institute of Technology Jaipur, India. The matter embodied in this thesis is original and has not been submitted to any other University or institute for the award of any other degree.

Dr. Neeta Nain
(Supervisor)
Assistant Professor
Computer Science and Engineering
MNIT Jaipur
India

Dr. Mushtaq Ahmed
(Co-Supervisor)
Assistant Professor
Computer Science and Engineering
MNIT Jaipur
India

Place: Jaipur
Date:

DECLARATION

I, **Tapas Badal**, declare that this thesis titled,**"Nonchronological Video Synopsis and Indexing"** and the work presented in it, are my own, I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this university.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself, jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Tapas Badal
(2013RCP9004)

Date:

Place: MNIT Jaipur

ACKNOWLEDGMENTS

I would first like to thank God who gave me the grace and privilege to pursue this program and successfully complete it in spite of many challenges faced. I express my heartfelt gratitude to my supervisors **Dr. Neeta Nain** for constant inspiration, encouragement, valuable advice, enormous support, and blessings. It is needless to say without her support and encouragement; this research work would not have been possible. I also would like to thank my co-supervisor **Dr. Mushtaq Ahmed** for his constant support, availability and constructive suggestions, which were determinant for the accomplishment of the work presented in this thesis.

My special thanks to the members of Doctoral Research Ethics Committee (DREC), **Dr. Neeta Nain, Dr. Mushtaq Ahmed, Dr. Girdhari Singh, Dr. Dinesh Gopalani** for their constructive criticisms and valuable suggestions. I am also grateful to all other faculty members of the department especially **Dr. Yogesh Meena** for their helpful discussions and ideas at various stages. I am also thankful to the officials of the department for their kind help in official work.

Let me express special thanks to Head of the Department, **Dr. Girdhari Singh** , for the keen support and consistent encouragement in our academic activities. I am extremely thankful for **Prof. Udaykumar R Yaragatti**, Director, MNIT Jaipur for providing me infrastructural facilities to work in, without which this work would not have been possible.

I would also like to thank all the referees who reviewed this work as pieces of it were submitted to Journals and conferences. Their detailed reviews, constructive criticism and excellent advice have improved both the presentation and content of this thesis.

I would like to thank my co-scholars **Rajat, Ravindra, Abhishek, Anurag, Maninder, Riti, Maroti, Sandeep, Praveen, Mohit** and other research colleagues of the department for their loving cooperation, positive criticism, excellent advice, consistent support and consideration during the preparation of this thesis and to spend quality time together. I can not forget the valuable conversation and suggestions of **Dr. Prakash Choudhary** and **Sonu Lamba**. I have often looked towards them for their valuable suggestions, and they always helped me wherever I needed support in my research.

I am truly grateful to my parents for their immeasurable love and care. They have always encouraged me to explore my potential and pursue my dreams. They

helped me a lot to reach this stage in my life. I would like to thank my wife **Abha** for supporting and keeping me motivated throughout this work. Their unfailing love and support have always been my strength. Their patience and sacrifice will remain my inspiration throughout my life.

Finally, I gratefully, acknowledge one and all who are directly or indirectly involved to shape this research work.

Contents

Abstract

Security in public domain has become a critical issue from mid 90^s which causes an exponential increase in the numbers of surveillance systems installed around the world. These cameras and sensors generate an enormous amount of data. Manual browsing of millions of hours of digitized video from thousands of cameras proved impossible within the time sensed period. The need of automated smart video analysis system has increased as most of the data captured by surveillance or traffic camera are not possible to analyze by human operators in a time-bound period. Also, hours of these videos show no activity or events of interest. It also requires a system to segment various semantic level information present in a video with limited storage, power, and communication bandwidth. There is an urgent need for an automated system for significant video event modeling that can facilitate many applications. Thus, various groups of researchers, intellectuals, industries, and institutions are paying much interest in such a field. If a system can replace human resources to automate the analysis of activities of objects in a video scene that can solve many real world problems existing in various domains like military, banking sector, biometric identification, shopping areas, institutes, railway stations, and airports. Some other important applications include road traffic analysis, air traffic control, satellite imaging and terrain analysis.

Video Synopsis summarizes hours of video surveillance recordings into a short duration that takes only minutes to review. Video synopsis simultaneously presents multiple objects, events, and activities that have occurred at a different period of times in a video. It tracks and analyzes moving objects (also called events), and converts video streams into a database of objects and activities. The database keeps only key events that contain valuable information and removes unwanted video sequences which have no activity.

This dissertation aims to contribute in each area of video-based analysis of object motion structure. We also find out the limitations of approaches present in existing literature to bring out the solution of fully automatic detection and analysis of video activities. This dissertation focuses on four topics, namely video-based moving object detection, multiple object tracking, activity analysis and synopsis video generation.

The moving object detection is an essential step in any video-based analysis system because the performance of the whole system depends on the result of this phase. We propose a multi-layer codebook model for segmenting dynamic pixels,

for which a novel weight mechanism is used to add new background codewords into codebook. It is also used to decide the sufficient number of codewords needed to represent background model with a significant reduction in false positives. The adaptive distance measure is presented to eliminate the effects of background dynamics. Furthermore, to reduce shadow and illumination effects, a cone-shaped color distance map is utilized instead of cylindrical. This incorporates spatial context by applying random neighbor selection policy.

With the recent advancements in computer vision, it can be reasonably claimed that there are feasible solutions available to address the robust tracking of the single target. However, simultaneous analysis of multiple targets motion structure in a video remains one of the most challenging tasks to accomplish. A tracking approach is presented which applies to trajectory formation of multiple objects with complex random motion structure. The multiple-instance tracking framework is formulated to incorporate spatiotemporal information. It selects significant features and establishes the statistical correlation between a prior model of the target and its recent observation. Proximity measurement scheme is applied to initialize structural context information in tracking-by-detection framework. It improves performance by completing target trajectory, reducing ID switches, and trajectory segmentation.

Although the existing video synopsis approaches work well in condensing activities present in video over space, they do not classify natural and abnormal movements present in the video. It is crucial to have an automatic method that can analyze video content and generate various types of Meta data, such as time-stamped tags and highlights. This information improves user experience by generating high-quality video synopsis. In this regard, an automatic and scalable solution is proposed that is based on different criteria as a signature for an activity. A hierarchical fashion is employed to efficiently search important activities present in video sequence while considering both the spatial collision and the temporal consistency among objects.

The moving object detection approach is tested over numerous (six) videos from benchmark dataset available at Goyette et al. (2012) with complex illumination and background situations. We experimentally demonstrate the improvement over the state-of-the-art background subtraction models. The performance of our multi-object tracking approach is evaluated on publicly available Benchmark datasets MOTChallenge Leal-Taixé et al. (2015a). The method performs a lot better than other state-of-the-art methods used for multiple objects tracking in videos. Fi-

nally, synopsis video of important activity is generated on several real-world video sequences. Comparative accuracy with minimum computational complexity is achieved for each phase and reported in this research work.

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Smart video surveillance systems are required to automate the analysis of video data and extract various analytical information as per needs of particular applications. Any system capable of automatic analysis of object activities in a video, can be a solution to many real world activity analysis problems. It is a very challenging and fertile research domain with many promising applications like video surveillance, biometric identification, satellite imaging, terrain analysis, augmented reality, face or human detection, user tracking, gesture recognition, behavior analysis, traffic analysis etc. Thus, it has drawn attention of several researchers, institutions and commercial industries.

Video synopsis is a compact representation of video sequences that enable the browsing and retrieval of hours of video footage in few minutes. It segments activities present in a video at different time period and arranges them over a common spatial region. The video synopsis provides valuable information that can be used in various applications which demand video-based monitoring in short duration of time. With this compact representation, one can quickly analyze many hours of traffic videos, segment human behavior, and can introduce new interaction methods in a gaming console and for law enforcement. Also, the intelligence agencies and security industries can utilize this representation by analyzing human activities in surveillance videos to identify suspicious behavior or to determine the individuals in real time.

While generating the synopsis video of activities present in a video, it comprises primarily four steps. First, it involves moving object segmentation which is also termed as foreground segmentation from the stationary region considered as background. Although the traditional methods of moving object segmentation like

optical flow, background subtraction, and temporal differencing etc., provide satisfactory results in detection of single moving objects, challenges are encountered in the case of multiple object streams and poor lighting conditions. To overcome these challenges, a robust motion detection method is required which can handle foreground commonage, shadows, and moving backgrounds. This method needs to update the background model continuously to maintain high-quality segmentation over extended periods of time. Also, it should efficiently detect multiple moving objects in adverse lighting conditions.

Tracking as a second step is defined as following the trajectory of an object across frames as it moves around the scene. It requires assigning consistent labels to the segmented objects in the whole frame sequence of a video. Further, based on the tracking domain, a tracker can give useful information such as movement, shape, and orientation of the object of interest. With the recent advancements in computer vision, it might be reasonably claimed that there are feasible solutions available for addressing the robust tracking of the single target, however, simultaneously analyzing multiple target's motion and tracking them in video stays as one of the most challenging problems in computer vision. The output produced by the tracking step is used to support and enhance motion segmentation, object classification and higher level video analysis.

Third, to maintain the critical context cues present in a video it is essential to employ a method to search important activities present in a video sequence efficiently. The final step in generating synopsis video of important activities is to organize the motion structure of segmented objects and create short descriptions of their actions. It may simply be considered as a representation of object motion structure over common spatial and temporal domain.

We aim to study and implement an optimized approach for each step in generating the synopsis video. A robust algorithm for object detection and tracking is presented that can detect and track multiple objects in a variety of challenging real-world scenarios. A hierarchical fashion is employed to efficiently search important activities present in a video sequence, and generating synopsis video of those important events. Here both the spatial collision and the temporal consistency are considered while creating synopsis video.

1.1 Motivation

Security in public domain has become a critical issue from mid 90^s which causes an exponential increase in the numbers of surveillance systems installed around the world. These cameras and sensors generate an enormous amount of data. Manual browsing of millions of hours of digitized video from thousands of cameras proved impossible within the time sensed period. Applications, where storage, communication bandwidth, and power are limited, require a system to represent information of a video in a comparatively less spatial and time domain. The role of such systems has shifted from purely passively recording information for forensics to pro-actively providing analytic information about potential threats and dangers in the real-time fashion. Selecting an adequate reasoning mechanism coupled with suitable events modeling is crucial for event-driven applications.

There are places of public domain where video analysis has broad applications for security purpose such as in the military, banking sector, biometric identification, shopping areas, Institutes, train stations, and airports. Figure 1.1on page 4 depicts some applications of segmentation of moving object trajectory also known as motion structure. Various other important applications include road traffic analysis, air traffic control, satellite imaging, terrain analysis, augmented reality and robotics.

One of the primary driving applications of motion analysis has been an automated analysis of moving objects present in a video, partially motivated by the focus on security and prevention of terrorist attacks increased in recent years.

Object detection and activity analysis have a wide range of applications in the field of video analysis, surveillance systems and natural science. Object tracking has gained the attention of researchers where security is the prerequisite, and it is not possible for human operators to monitor every surveillance system continuously. Motion structure of moving objects represents vital information about the movement and activities present in a video. It can also be used for further knowledge extraction in various video processing applications such as behavioral analysis, crowd analysis, etc.

The smart surveillance system is the main scope of this research work. The goal of this thesis is to implement a robust algorithm for foreground segmentation using computer vision approach and to develop an efficient method for tracking the segmented objects across the frames, which leads to the selection of important ac-

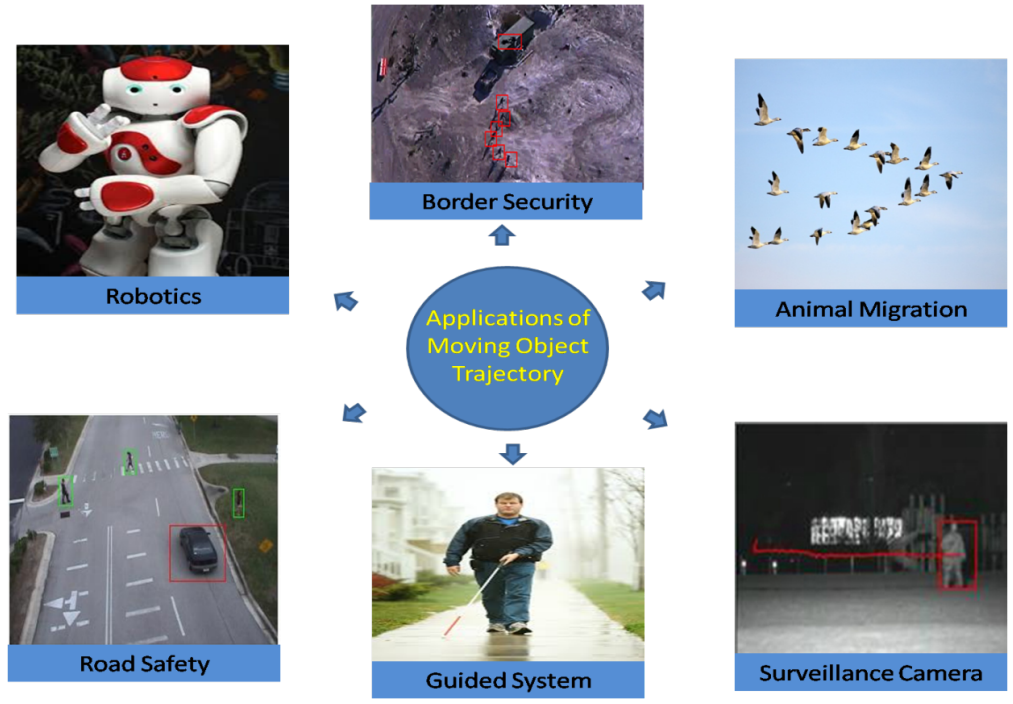


Figure 1.1: Application examples for object trajectory segmentation. Next to common scenarios involving (a) robotics accident prevention, (b) border security (c) it can also be used to study animal behavior , (d) road safety (e) guided system (f) Surveillance camera.

tivities from motion structure of tracked objects enabling compact representation in the form of synopsis video.

1.2 System Description

A system to generate synopsis video of important activities present in a video comprises mainly four functional blocks. These are object detection, tracking, activity analysis and finally synopsis video generation. A brief explanation about these blocks are given below:

- **Foreground Segmentation:** Detection of moving objects as the foreground region is performed at lowest semantic level.
- **Object Classification and Tracking:** At this level, objects of interest like humans or vehicles segmented in the previous stage are tracked across the frames.

- **Activity Analysis:** Based on low-level semantic knowledge extracted from earlier stages, the more sophisticated understanding of the videos focuses on identifying object actions, understanding the behavior and activities of each object by segmenting motion structure of the individual object.
- **Synopsis Video Generation:** Arranging those activities segmented in the previous phase over the spatial and temporal domain to represent them in less space and time.

The design of smart synopsis video system requires fast, reliable and robust algorithms for object detection, classification, tracking, and activity analysis. A basic block diagram of synopsis video system is given below in Figure 1.2 on page 5.

Figure 1.2: System overview of generating synopsis video in this work. Given a set of video frames, the task is to generate the synopsis video of important activities.

1.3 Aims and Objectives

The computer vision research presented in this dissertation does not target a particular application of the movement analysis rather shows work on systems that can incorporate many different applications. The outputs of this research can be used for providing the human operator with high-level data to help him to make the decisions more accurately and in a shorter time. This work also provides offline indexing and efficient searching of stored video data. The advances in the development of these algorithms would lead to breakthroughs in applications that use visual surveillance.

Methods for foreground segmentation do not separate objects in space due to which analysis of individual objects is not possible. It is required to formulate an efficient algorithm for tracking the segmented object across the frames. Also, while segmenting information such systems are expected to maintain spatial and temporal dependencies between objects. Computational complexity has to be low

as a large number of cameras need to be observed simultaneously or the algorithms are required to be embedded in a camera or as an embedded system with limited computational power. Manual browsing and analysis of recorded footage is still a costly, labor-intensive and time-intensive task. Our motivation for studying this problem is to create a visual surveillance system with real-time object detection, classification, and tracking and activity analysis capabilities.

Given the above shortcomings, this research work focuses mainly on following objectives:

1. Study recent issues and challenges in moving object detection, multi-object tracking, activity classification and synopsis video generation so that they can be optimized and applied to real-life applications.

2. It has been observed that an adaptive background subtraction model is needed to segment moving objects in the dynamic environment. It is also required to formulate a proper learning rate for background updation to reflect changes occurred in the background in an optimized manner.

3. To find a robust and efficient background subtraction method to segment moving objects in a dynamic environment. The effectiveness of this step is crucial for the whole system because outcomes of this phase are input for further tracking, as the final result depends mainly on the outputs generated in this stage.

4. With the recent advancements in computer vision, it can be claimed that there are feasible solutions available for addressing the robust tracking of the single target. However, simultaneous analysis of multiple target's motion structure in a video stays as one of the most challenging tasks in computer vision. Our objective is to develop and optimize feasible solutions for multiple object tracking in the presence of various tracking challenges such as occlusion and miss detection.

5. The performance of activity analysis mainly depends upon the accurate representation of the segmented trajectory of the tracked object. We aimed at representing trajectory information in the non-visual form that can be analyzed effectively. Array representation of segmented trajectory is also done for activity detection of significant events.

6. We also addressed the crucial objective to have an automatic method to analyze video content and produce various types of metadata, such as time-

stamped tags and highlights. This information can help to improve the user experience by providing indexing of activities, especially when activities are shifted over time-domain in synopsis video.

7. To propose a solution for generic important event classification for generating high-quality event specific video synopsis.

8. It has been observed that the current approach of video synopsis works well in condensing activities present in video over space, while they do not preserve interaction between objects. While going through the various surveillance videos, it is observed that the object interaction in video possessed vital information such as information exchange, accidents, and theft.

9. To implement a method to generate a synopsis video of important activities present in a video with minimum computation cost as well as maximize the accuracy.

10. Evaluate the proposed methods on benchmark dataset and compare the results with state-of-art approaches.

1.4 Contributions

Our contributions to this study are to create a visual surveillance system with real-time object detection, tracking, classification, and synopsis video generation. The study summarizes as:

1. Explore the literature work to find various steps followed in generating synopsis video.

2. This work presents an adaptive multi-layer background subtraction method by implementing various improvements over basic codebook model. These enhancements are:

   (a) To control learning rate of background model, each codeword is assigned a weight using the Sigmoid function.

   (b) The adaptive color distance measure is applied to compute the decision threshold for foreground pixel classification.

(c) As background may have the spatial motions that may lead to shift in a pixel, this kind of situation is resolved by considering spatial context shared between neighboring pixels, by using a random neighbor selection policy.

(d) The proposed method reduces false positive pixels detected conventionally as ghost regions by maintaining codeword belonging to uncovered background region in background codebook.

3. We formulate a data association task for track initialization; which intends to detect all high probability tracks.

4. A novel framework is designed for coupling the subproblems of data association and appearance based tracker by the formulation of proximity measurement between interacting objects.

5. A new sparsity-driven target specific proposal distribution technique is devised that takes segmented foreground regions as input to select features belonging to a target.

6. This thesis presents an automatic approach of condensing the specific activities in surveillance video by considering the spatial and temporal relationship between them. In this work, we aim to segment movements present in a video that is found interesting. To this end, we propose to combine several cues to assess significant events in a novel manner that leads to increasing in performance.

7. This research work presents an approach of condensing the activities in surveillance video while preserving the interaction between the objects.

1.5 Dissertation Organization

The rest of this dissertation is structured as follows:

In Chapter 2, we review existing literature on video synopsis and its functional blocks. In Chapter 3, we present our approach on background subtraction for moving object detection using improve codebook model. Chapter 4 provides the detailed description of our approach for multi-object tracking using multiple instance learning techniques. Chapter 5 describes an algorithm for generating video

synopsis and its implementation. Finally, Chapter 6 concludes the thesis with a discussion of the identified directions for future work.

Next chapter provides a literature survey of different functional blocks in synopsis video generation technique. In this chapter, we investigate challenges present in various phases involved, to complete the process of video synopsis. It also addresses the limitations of widely used techniques for each of these phases.

In Chapter 3, explains the various applications and necessity of moving object detection in video analysis system. This chapter covers detailed description of various object detection approaches given in literature and compares them in respect of performance and complexity with their limitations. The chapter describes challenges occur in segmenting moving object region. The chapter includes detail study of background subtraction model used for foreground segmentation and our proposed approach for segmenting moving object region that gives better performance than state-of-art.

Chapter 4 discusses, the most widely used methods available for the multi-object tracking and trajectory segmentation. The chapter describes the details about the various types of situations where the performance of existing techniques degrades. The chapter also explains methodology proposed for multiple object tracking and its implementation details. Further, covers a comparative study of the proposed model for multiple object tracking with existing methods and report result over benchmark dataset.

Chapter 5 describes the complete framework of generating synopsis video along with various notations used while implementing the proposed model. The chapter further concludes with the quantitative and qualitative evaluation of result using proposed approach. We compare the result with a number of existing methods.

Finally, Chapter 6 summarizes the research work and concludes the study that has been made. It also furnishes some future direction of the video synopsis study.

Chapter 2

LITERATURE REVIEW

A lot of literature is available for object detection, tracking, and activity analysis applied to a video surveillance system. In this chapter, we report a survey of those researchers that share a mutual interest in our work. However, for presenting complete review, we also include some similar approaches and cover techniques used for the same task, but we have not used them in this work. The structure explained in the previous chapter includes four fundamental steps in generating synopsis video of important activities, those are moving object detection, object tracking, activity classification and synopsis generation. In this framework information flow from one stage to other, therefore, in this order will be a better choice for giving the literature. This chapter covers recent advances and research related to each of these stages.

2.1 Moving Object Detection

Each application that benefits from video processing have different needs thus requires different treatment. Moving objects are the common and most important thing between them all. Thus, detecting moving objects such as people and vehicles in a video is the fundamental step in various video processing tasks. Most of the object detection methods endeavor to locate connected regions of pixels that represent the moving objects within the scene. Moving object detection is also termed as foreground segmentation, it is the process of separating image pixel or region into two parts, foreground and background. Foreground region represents moving objects present in the scene like the person, vehicle, etc. Background belongs to the region that does not change its appearance over the time such as

wall, road, sky, etc. An effective technique for object detection is needed because it provides a focus of attention and simplifies the processing in subsequent analysis steps.

The simplest and earliest work reported to segment moving object present in a video is given by Lipton et al. (1998). In this work, they proposed frame differencing that uses pixel-wise differences of two images for segmenting the moving regions. However, the result of frame difference is not suitable for the majority of surveillance applications. Also, the performance is severely degraded with the dynamic background.

The recent and widely used moving object detection approaches can be mainly grouped into two main categories named as: orientation-based and distribution-based.

Orientation-based object detection estimates a vector to represent direction and velocity of each pixel (x, y) in an image. In this category optical flow (OF) is the most popular technique for moving object detection Liu et al. (1998). While considering **distribution-based** moving object detection techniques, background subtraction is the most widely used and successful method. Background subtraction mainly depends on the estimation of the reference model used to represent the background of a scene. The moving object is detected by subtracting the current image from the background reference model or image.

In the following subsections, we present a complete study of these widely used moving object detection methods.

2.1.1 Frame Difference Method

The frame differencing is the simplest of all the moving object detection techniques. In this method, moving objects are recognized by subtracting the corresponding pixel values from current frame with the previous frame. The difference is then compared to a threshold value for determining the background and foreground region. It is a non-recursive technique Mashak et al. (2010) where no history of the video frames are required.

Let us denote the intensity value of a pixel at location (x, y) at time t as $I(x, y, t)$. Then according to the method, the difference between the frame at time t and the

frame at time $t-1$ is determined as follows:

$$D = \mid I(x,y,t) - I(x,y,t-1) \mid \tag{2.1}$$

For each pixel, the value of D is compared to a threshold Th and classified as follows:

$$P_{(x,y)} = \begin{cases} Foreground & \text{if } D \geq Th \\ Background & \text{else} \end{cases} \tag{2.2}$$

Here, choice of an optimal threshold value is an important consideration. A too low value of Th will add unwanted noise, while a too high value may classify a foreground pixel as background.

Frame differencing is easy to implement, and computational complexity is low. But, it suffers from aperture problem and use of a single threshold value for all pixels degrade the system performance. It is not suitable for many surveillance systems.

2.1.2 Optical Flow

Optical flow techniques use the flow vectors of moving objects over time for detection of moving regions in an image sequence Li et al. (2010). Lucas and Kanade Lucas et al. (1981) used optical flow for motion detection. It is based on the assumption used by most of the optical flow methods that intensity I of moving pixel is constant in subsequent frames. It is computed by taking two images at time t and $t+\delta t$.

$$I(x,y,t) = I(x+\delta x, y+\delta y, t+\delta t) \tag{2.3}$$

By using Taylor series, above equation is expanded to:

$$I\left(x+\delta x, y+\delta y, t+\delta t\right) = I(x,y,t) + \frac{dI}{dx}\delta x + \frac{dI}{dy}\delta y + \frac{dI}{dt}\delta t + HOT \tag{2.4}$$

Avoiding the Higher Order Terms (HOT) in Equation 2.4, the equation reduces to :

$$\frac{dI}{dx}\delta x + \frac{dI}{dy}\delta y + \frac{dI}{dt}\delta t = 0 \tag{2.5}$$

$$\frac{dI}{dx}\frac{\delta x}{\delta t} + \frac{dI}{dy}\frac{\delta y}{\delta t} + \frac{dI}{dt} = 0 \tag{2.6}$$

$$\frac{dI}{dx}V_x + \frac{dI}{dy}V_x + \frac{dI}{dt} = 0 \tag{2.7}$$

$$I_x \times V_x + I_y \times V_y = -I_t \tag{2.8}$$

where V_x, V_y represents optical flow vectors and I_x, I_y represent derivatives of the image intensities at coordinate (x, y) of image I at time t.

The values V_x, V_y are used to get the motion vector for the object detection by applying thresholding technique. The magnitude of a motion vector is computed as:

$$M = \sqrt{V_x^2 + V_y^2} \tag{2.9}$$

Thresholding is applied on this M value. Finally, the moving object region is refined by using morphological operations. Optical Flow can cope even when the camera is shaking. It is computationally complex and needs specialized hardware to do processing in real time Lucas et al. (1981).

2.1.3 Background Subtraction Model

Background subtraction is an extensively followed technique for moving object detection where the significant changes in an area of interest are considered as foreground. It detects moving pixels by subtracting current frame from the reference frame represented as background model or by any other statistical parameters. Numerous methods are proposed in the literature for background subtraction where each of them follows different techniques for representing background model and its subtraction with the current image. There are several surveys like Benezeth et al. (2008); Brutzer et al. (2011) etc., devoted to this topic. It is mainly a three-step process: background modeling, background subtraction, and updation.

Background modeling is the most decisive step in generating significant outcomes. Here the background is represented as a reference image or by using any statistical model. A background model should be efficient enough to deal with noise introduced due to camera motion or natural variation in the scenes like blow wind effect, change in light and illumination change. It should also be adaptive to changes occurring in the case of dynamic backgrounds like a permanent variation in some portion of background or scene. In the following subsections, we explain various background subtraction techniques used in this work for comparative analysis.

2.1.3.1 Approximate Median Filter

One of the most commonly used background modeling techniques is median filtering. McFarlane and Schofield (1995) proposed this method. Here, the background model is generated by taking the median of the last n frames. It is based on the assumption that a pixel belongs to the background for at least half of the frames present in the buffer.

A buffer of size n is maintained to store the recent n pixel values at each location of the past n frames. Median is then calculated from values present in the buffer. This median value acts as a reference value for the next input frame. Foreground pixels are determined by calculating the difference of the current pixel value to the median value as follows:

$$P(x,y,t) = \begin{cases} Foreground & \text{if } \mid I(x,y,t) - Med(x,y,t) \mid \geq Th \\ Background & \text{else} \end{cases} \tag{2.10}$$

Where $Med(x,y,t)$ is the median value of the buffer at time t, $I(x,y,t)$ is the intensity value of pixel at location (x,y) at time t, Th is the threshold value. The median value is updated for every new frame. It has the disadvantage of maintaining a buffer in memory of recent pixel values.

Lo and Velastin (2001) proposed the recursive technique for median filtering which does not maintain a buffer. Instead, the background model is updated recursively for each new input frame. In this method, if the value of the input pixel is $>$ the value of the corresponding background pixel then the median is incremented by 1. If it is $<$ the estimate, the median is decremented by 1, otherwise it remains same. It uses the following update equation:

$$B_{t+1} = \begin{cases} B_{t+1} & \text{if } I(x,y,t) \geq B_t \\ B_{t-1} & \text{if } I(x,y,t) \leq B_t \\ B_t & \text{if } I(x,y,t) = B_t \end{cases} \tag{2.11}$$

This method gives a value which is larger than half of the pixels and smaller than half of the rest pixels. The value thus obtained is the median of all the pixel values. This approach is simple, computationally efficient, and robust to noise. The recursive technique requires less space as compared to non-recursive techniques. As seen by Hung et al. (2014), it has the drawback of adapting to

a background slowly when there is a large change in the background. Thus, if a long time stationary object starts moving suddenly, it may show as faded into the background before it gets many frames to learn the new background region. Also, the variance in a pixel intensity is not modeled.

2.1.3.2 Running Gaussian Average

The unimodal representation of background is introduced by Wren et al. (1997). The running Gaussian model consists of fitting a Gaussian probability density function (PDF) over the last n pixel's intensity values. The PDF of each pixel has two parameters : mean (μ) and variance (σ^2). Initially, mean and variance for the first input frame is initialized as follows:

$$\mu = I_0 \quad \text{and} \quad \sigma^2 = V \tag{2.12}$$

Where, I_0 ans V are the intensity and variance values assigned to first initial frame , V is assigned any default value generally taken as 36. At each new frame, the mean is updated as:

$$\mu_t = \alpha I_t + (1 - \alpha)\mu_{t-1} \tag{2.13}$$

Where I_t is the current pixel intensity value at time t, μ_t is the previous average, α is the empirical weight which ranges from 0 to 1.

Similarly the variance is updated as:

$$\sigma_t^2 = \delta^2\alpha + (1 - \alpha)\sigma_{t-1}^2 \tag{2.14}$$

Where δ is the distance between intensity value I_t and average μ_t at time t is calculated as:

$$\delta = \mid (I_t - \mu_t) \mid \tag{2.15}$$

The foreground pixels are then determined for each frame if the following condition holds true:

$$p(x, y, t) = \begin{cases} foreground & if \mid I_t - u_t \mid > k\sigma \\ background & if \mid I_t - u_t \mid \leq k\sigma \end{cases}$$

Where k is generally taken as 2.5.

Su and Chen (2008) introduced a variant of the method which updates the mean only when the corresponding pixel is characterized as background. It optimizes the outcomes by preventing newly added moving objects from fading into the background. The model is updated as:

$$\mu_t = M\mu_{t-1} + (1 - M)(\alpha I_t + (1 - \alpha)\mu_{t-1}) \tag{2.17}$$

Where $M = 1$ if I_t is classified as foreground, and $M = 0$ if I_t is classified as background. It required fitting the *pdf* from scratch on each pixel at the time of each new input frame. To avoid this, an average running method is used.

It is computationally efficient and requires comparatively less memory. As it uses a single Gaussian distribution of color values for representing background at each pixel. It is not able to handle backgrounds having multiple histograms peak values for a single pixel. It cannot cope with multi-modal backgrounds such as a background with waving trees and sky. Such scenes and similar will be incorrectly classified as foreground regions with this method. Also, it can not handle gradual or sudden lighting changes in the scene. Lahraichi et al. (2016) uses bimodal intensity distribution for each pixel. Although it is efficient and gave satisfactory results for an ideal environment, the performance degrades for complex situations like dynamic background and change in illumination.

2.1.3.3 Gaussian Mixture Model

Among the many pixel-level background subtraction methods Gaussian Mixture Model (GMM) is used most widely. This method is used when multiple surfaces form part of a background, and thus multiple Gaussian are necessary. Stauffer and Grimson (1999) proposed this method by modeling each pixel as a mixture of multiple Gaussians.

A "pixel process" is considered which contains the pixel's history as the intensities values till time t:

$$X_{1...t} = \left\{ \; I(x_0, y_0, i) : 1 \leq i \leq t \; \right\} \tag{2.18}$$

where I is the frame sequence.

This history of a pixel at time t is modeled by a mixture of K Gaussian distribu-

tions. The probability of occurrence of the current pixel value is:

$$P(X_t) = \sum_{i=1}^{k} \omega_{i,t} \times \eta \left\{ X_t, \mu_{i,t}, \sum_{i,t} \right\} \tag{2.19}$$

where K is the number of Gaussian distributions, $\omega_{i,t}$ is the weight estimate of the i^{th} Gaussian in the model at time t, $\mu_{i,t}$ is the mean value of the i^{th} Gaussian in the model at time t, $\sum_{i,t}$ is the co-variance matrix of the i^{th} Gaussian in the model at time t, and where η is a Gaussian probability density defined as follows:

$$\eta(X_t, \mu, \sum) = \frac{1}{(2\pi)^{\frac{n}{2}} \mid \sum^{\frac{1}{2}} \mid} e^{\frac{-1}{2}(X_t - \mu_t)^T \sum^{-1} (X_t - \mu_t)} \tag{2.20}$$

K is assigned value as 3 to 5 depending on available memory. By assuming the red, green and blue pixels as independent of each other for computational reasons, the co-variance matrix reduces to diagonal:

$$\sum_{k,t} = \sigma_k^2 I \tag{2.21}$$

An online K-means approximation is used where every new pixel value, X_t, is checked against the existing K Gaussian distributions until a match is found. A pixel is said to be matched if its value lies within 2.5 standard deviation of the distribution. If there is no match for all of the K distributions, the least probable distribution having the lowest weight is replaced with the current value as its mean value, an initially high variance, and low prior weight. The weights for K distributions are updated as follows:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{2.22}$$

Where α is the learning rate assigned values between 0 to 1.

The parameters μ_t and σ^2 for the matched component are updated as follows:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{2.23}$$

$$\sigma^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t) \tag{2.24}$$

Where ρ is another learning parameter approximated as :

$$\rho = \frac{\sigma}{\omega_{k,t}} \tag{2.25}$$

The Gaussian distributions are ordered by the value of $\left(\frac{\omega_{k,t}}{\sigma}\right)$. Then simply the first B distributions are chosen as the background model:

$$B = argmin_b \left(\sum_{k=1}^{b} w_k \geq T \right) \tag{2.26}$$

Where T is a threshold. Then, those pixels whose color $I_{x,t}$ is located at more than 2.5 standard deviations away from every B distributions are labeled foreground.

A different threshold is selected for every pixel. This threshold is not global and adapts with time for each pixel. It has high accuracy.

Although GMM gives a real-time performance in moving object detection, it fails to include shadows with the background. Another serious issue to deal with Gaussian-based approaches is parameters estimation mainly to adapt to changing the background. Its parameters require careful tuning and should be selected intelligently. It is computationally intensive. It cannot deal with sudden drastic changes in illumination. This method is followed and extended by various researchers like Zivkovic (2004); KaewTraKulPong and Bowden (2002) etc.

2.1.3.4 Codebook Model

The Codebook model by Kim et al. (2005) presents a real-time algorithm for segmentation of foreground and background. The Codebook presents a quantization approach to represent dynamic background model using a set of codewords at each pixel in compressed form. It reduces the memory requirement to represent background variations by building background model for each pixel as codebook consisting one or more codewords as $M = \{c_1, c_2, ..., c_m\}$. Codewords comprise information of background pixel regarding parameters like minimum and maximum intensity and color values instead of frame samples representing background, which significantly reduces the memory requirement. This method can handle scenes containing moving backgrounds or illumination variations, and it achieves robust detection for different types of videos.

This model follow three steps while segmenting background and foreground:

1. Construction of the initial Codebook
2. Codebook refinement
3. Foreground detection

In the training period the initial Codebook model to represent background is constructed. Each pixel value x_t sampled at time t is compared to the current Codebook to determine which codeword c_m (if any) it matches (m is the matching codewords index). To determine which codeword will be the best match, a color distortion measure and brightness bounds is used.

In Codebook refinement phase the large Codebook is refined by separating the codewords that might contain moving foreground objects from the true background codewords, thus allowing moving foreground objects during the initial training period. After the training phase true background is modeled, which includes both static pixels and periodically moving background pixels. The criterion of maximum negative run-length ($MNRL$) as λ is used to refine Codebook which is defined as the maximum interval of time that the codeword has not recurred during the training period.

A straight technique of computing the distance of the sample from the nearest cluster mean is used to perform background subtraction. The subtraction operation for an incoming pixel value x_t in the test set include two operations:

- Color distance measure
- Brightness Measure

This color distance is sensitive to brightness change over the pixel. To compensate the brightness change, the minimum and maximum value of brightness is calculated during codebook updation. A logical brightness function is used to find pixel values within the range.

Improvements in codebook model as multi-layer and integrating it with spatial and temporal information of pixel has been proposed by Sigari and Fathy (2008). Layered codebook model is proposed to extract structure of background and models. Layered codebook is a simple data structure containing two codebooks defined per pixel. The first layer is main codebook represented by M, while the second is cache codebook denoted as H, and both contain some codewords relative to a pixel. Main codebook models the current background images and cache codebook

is used to model new background images during input sequence. This method can model moving backgrounds, multi backgrounds and illumination changes and this is efficient in both memory and computational complexity.

In training phase, such as basic codebook model, only main codebook is constructed and cache codebook is empty. During input sequence, foreground-background is segmented and layered codebook model is updated. For layered codebook model three threshold are defined: T_H , T_{add} and T_{delete}. These thresholds are used to refine main and cache codebooks. If λ of a codeword in H is $> T_H$, this codeword will be deleted from H. If a codeword stays in H longer than a certain time (T_{add}), then it will be moved to M. If a codeword of M does not appear for a certain time (T_{delete}), then it will be removed from M.

2.1.3.5 ViBe-Visual Background Extractor

Barnich and Van Droogenbroeck (2011) proposed the foreground object segmentation algorithm $ViBe$ which stands for Visual Background Extractor.

Pixel Model and Classification Process: Contrary to the background models which are based on probability distribution function, the background model in $ViBe$ consists of a set of observed pixel values.

Let the value of a pixel at location x in a given color space be denoted as v_x, and v_i represents the i^{th} sample of the background model. Then for each pixel x, the background model is defined as the collection of N background sample values as given below in Equation 2.27:

$$M(x) = \{v_1, v_2, ..., v_N\} \tag{2.27}$$

The Figure 2.1 shows the classification of background and foreground pixels using Euclidean distance in the RGB color space as computed by Equation 2.28.

$$d(v_x, v_i) = \sqrt{(R_x - R_i)^2 + (G_x - G_i)^2 + (B_x - B_i)^2} \tag{2.28}$$

where $v_x = (R_x, G_x, B_x)$ and $v_i = (R_i, G_i, B_i)$ represents the red, green and blue components of a pixel x the reference pixel, and the sample pixel i respectively in background model $M(x)$ of pixel x.

Classification of a pixel is done by defining a sphere S_R of radius r having center

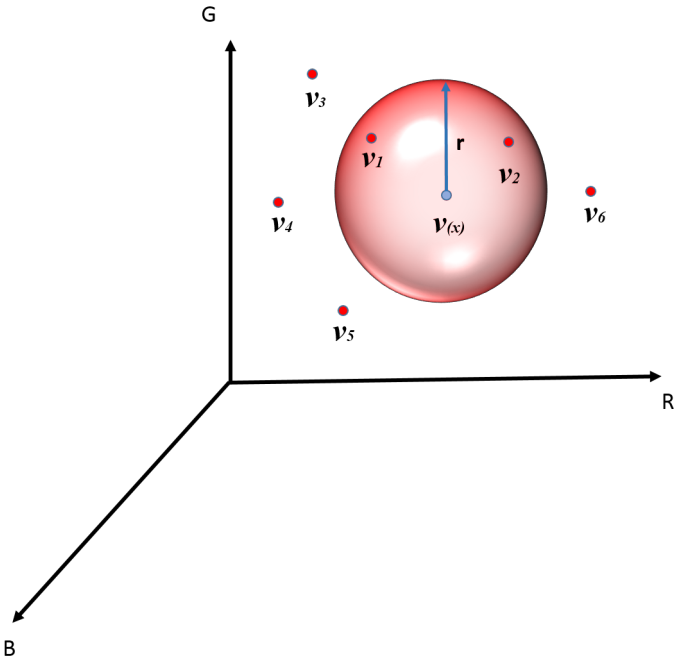


Figure 2.1: This figure shows the classification of background and foreground pixels using Euclidean distance in the RGB color space. Here, a 3D coordinate system is used, where the three-axis represent R, G, and B color space.

at the point v_x under consideration. A minimum cardinality, a priory set, denoted as, $\#_{min}$ is computed for every pixel. A pixel is considered as background if at least $\#_{min}$ samples matches to the model $M(x)$:

$$S_R(v_x \cap \{v_1, v_2, ..., v_N\} > \#_{min} \tag{2.29}$$

where # denotes the cardinality of the set intersection. This involves the computation of N distances between v_x and model samples v_i, and N comparisons with a threshold over Euclidean distance $d(v_x, v_i)$.

Update Policy: In updating a pixel model, the sample to be replaced is chosen randomly. The new value then replaces the chosen random value as shown in Figure 2.2. The expected remaining lifespan of any sample value of the model decays exponentially as:

$$P(t_0, t_1) = e^{-ln(\frac{N}{N-1})^{t_1 - t_0}} \tag{2.30}$$

Where $P(t_0, t_1)$ is the probability of a sample at time t_0 to be still present at time t_1. The Figure 2.2 gives pictorial representation of three possible models after update.

Model Initialization: The first frame is used to initialize the model. Values from

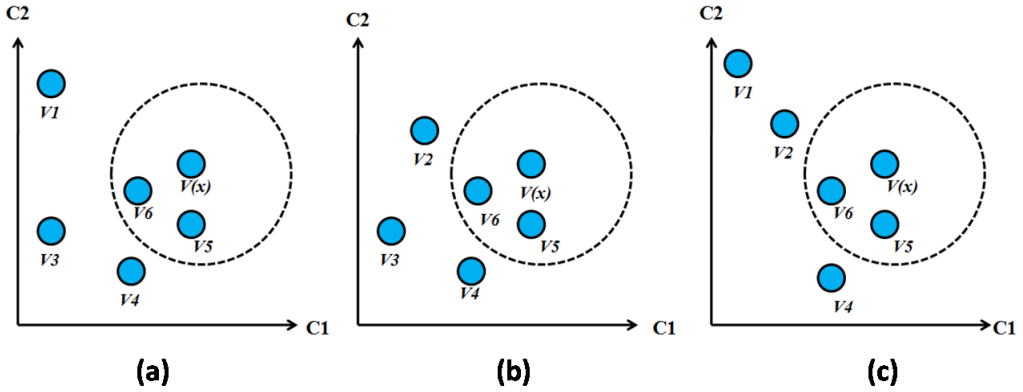


Figure 2.2: The three possible background models after the update step. It illustrate the random selection of neighbors that participated in background model updation process.

spatial neighborhood of each pixel are used to populate the model. These neighbors are chosen randomly. Segmentation of video sequences then starts from the second frame.

$ViBe$ shows accurate results in various environments without requiring any fine tuning of parameters. It has three fixed parameter values (matching threshold between a sample and a pixel value, number of samples stored in each pixel model, and the cardinality for the matches). It is stable for changes in illumination and camera shake. A moving object in the first frame will not be detected and introduces a ghost, which fades over time. It cannot generate a background image for each frame. It is not deterministic, as the results always differ if the algorithm is applied to same video multiple times.

A number of recent multi-object tracking methods practice pedestrian tracking Breitenstein et al. (2009); Führ and Jung (2014) and thus pursue detectors for a specific target, they mostly use Histogram of Gradient (HOG) as feature vectors and Support Vector Machine (SVM) for classification. These are pre-trained models that allow the system to accomplish robustness against partial occlusion, false detection, and illumination variations. However, high computational requirements and inability to deal with high variability in pose are a major issue in using specific detectors. Furthermore, applying a specialized detector limits the applicability of tracker in a multi-classes environment (e.g. scene with cars, pedestrian, animals, etc.).

The primary challenge in object detection methods those practice model based detectors is missed detection. It is mainly due to the low resolution or variation

in the orientation that requires adjusting those detectors online. The model based detector Dalal and Triggs (2005); Felzenszwalb et al. (2010) does not fit in several applications like surveillance systems where there is a requirement of tracking a moving object with varying appearance model. Alternately, if a possible background subtraction implementation is available, a primary objective is to establish a correspondence between observation and object silhouette in addition to scene information.

A lot of challenges occur while segmenting moving object from a scene. One of the primary challenges in moving object detection is sudden illumination changes. Unwanted noise will then be included in the results if the algorithm could not cope with the lighting changes and camera shaking. Non-static backgrounds would further increase the problem. Waving trees and weather changes could lead to improper result for the detection stage. Another difficulty is variations in the scene. A moving object may come to a stop for a moment and may diffuse in the background, or a stationary object may start moving. All these challenges should be met by a moving object detection method.

2.2 Object Tracking

Multi-object tracking is the inherent part of video analysis task in many video-based applications like smart surveillance system, augmented reality, crowd analysis and much more. Furthermore, trajectory analysis of target object in videos is serving as a foundation tool for various other significant computer vision techniques used for knowledge discovery. Many real-time applications need to have tracking methods which give reliable result even in situations like object with random movement, an interaction between objects, scale variation, and occlusion. Above parameters are mainly responsible for variation in accuracy of different tracking methods.

The object of interest in a tracking application primarily depends on the requirement of analysis. For example, it may consider moving vehicles on the road for surveillance, face tracking for identification, motion structure of person for behavior analysis, etc. Proper feature selection in object detection and tracking is a crucial part that may influence the result of tracking significantly. By simply imposing the parameters like motion velocity, direction, scale and structure of object the problem of feature selection can be overcome. Recently multiple moving object tracking has become a popular research topic in computer vision community. In

this section, we review some of the most related works over online multi-object tracking.

Many recent advances in techniques allied with object detection and tracking consider motion structure, color cues and data association techniques to assign a consistent label to an object across the frames Chen et al. (2015); Butt and Collins (2013); Milan et al. (2016).

With the recent advancements in computer vision, it can be claimed reasonably that there are feasible solutions available for addressing the robust tracking of the single target. However, simultaneous analysis of multiple targets motion structure in a video stays is one of the most challenging tasks in computer vision.

Although many tracking approaches operate on domain specific target representation that is either determined manually or trained using the initial frame sequence Bao et al. (2013); Yang and Jia (2016), these methods tend to have challenges while tracking objects that show convincing variation in their appearance. It has been demonstrated that in many scenarios an adaptive appearance model, which evolves during the tracking process as the appearance of the object varies, is pivotal to achieve high performance. Another choice in the design of appearance models is whether to model only the object or both the object and the background. Numerous approaches have shown that application of discriminative classifier in training a model to separate the background from the target object results in superior performance outcomes Milan et al. (2015). As these procedures commonly practice object detection they have been called "tracking by detection".

The tracking-by-detection approach use detectors output to associate current observations with existing trajectories. It is mainly categorized into the batch and online methods. The batch method also termed as global optimization tracking, and it links fragmented tracks together by using detections of complete frame sequence. Here general data association method is used to link short trajectories into a long trajectory Kamvar et al. (2004); Milan et al. (2016); Zhang et al. (2008). Although by analyzing complete frame sequences it may resolve some of the ambiguities present due to detection failure and make tracking system robust. The computational complexity increases exponentially if there is growth in the number of targets, and that questions the suitability of the system for real-time applications.

Batch method performance depends upon detection of the complete frame sequence, and hence computational complexity is a major limitation of these meth-

ods. Online tracking methods sequentially connect detection in the current frame with existing trajectories or create new trajectory if it does not find a match with any of the current trajectories. As it depends only on the current information, It is suitable for real-time applications. However, long-term occlusion and miss detection are two major issues with online multi-object tracking methods that tend to generate fragmented trajectories Jacobs et al. (2007); Yang and Jia (2016). Following subsection explain existing methods used for object tracking in perspective of data association and appearance based mechanism.

2.2.1 Data Association

The distance based approaches are the simplest and earliest techniques used for tracking, though they are not applicable in challenging situations like track overlapping, occlusion, and crowded environment. The Kalman filter Julier and Uhlmann (1997) has become a popular choice among state estimation model. It estimates the probability density function of posterior state and combines it with observation model to predict the future state of a target object.

Multi-hypothesis tracker (MHT) Reid (1979) and the joint probabilistic data association ($JPDA$) Fortmann et al. (1980) are two early concepts widely used for multiple targets tracking. The MHT finds an optimal assignment by associating hypothesis with each detection using position, speed, size, and appearance of an object over frame sequence. In order to assign hypothesis to an occluded object Song et.al. Song et al. (2008) apply online-trained classifiers while Breitenstein Breitenstein et al. (2009) combines confidence density map with the output of the detector to overcome occlusion. As these methods rely on local information between two consecutive frames, they produce fragmented trajectories belonging to the same object. Moreover, resulting trajectories may drift under occlusion.

$JPDA$ considers the probabilistic measurement between all tracks and detection points using different features. Recently, a lot of effort has been put in multi-target tracking for discrete-continuous energy minimization Milan et al. (2015), network flow problem Pirsiavash et al. (2011) and integer linear program Berclaz et al. (2011). To overcome the formation of multiple trajectories due to miss detection Fragkiadaki and Shi Fragkiadaki and Shi (2011) have formulated clustering of trajectories that improve multiple object tracking performance in cluttered environment.
Missed/false detection is still an inescapable issue, and these ambiguities could not

be resolved completely in the data association phase. Occluded objects are treated as missed detection or track as a single entity, and fragmented tracks belong to occluded object before and after occlusion are merged in further processing. To resolve occlusion, the primary task in above process is to layout an affinity model that measure similarity and fill gaps in between fragmented tracks by stitching them.

Fundamentally these approaches used a tracking-by-detection approach where the performance of system primarily depends on the initialization of appearance template. There is a capacity limit for a tracking system to resolve missed detection. It constrained the system to assume that these events are unlikely to occur. However, if the system shows frequent occlusion relying solely on data association, it may show a significant decline in results.

As detection and data association individually are incapable of resolving occlusion, a reasonable expansion is to consider coupling these two subproblems and treat them as a single objective function.

2.2.2 Appearance Based Learning

Appearance models using features like color histogram remain a popular choice for multi-object tracking Kuo and Nevatia (2011). However, orientation variation in object appearance is still a major issue needed to resolve correctly. To deal with change in appearance Breitenstein et al. (2009) propose target specific appearance update-models. However, they update appearance model only to distinguish an object from the background but not with other objects. To separate appearance model for different objects Kuo and Nevatia (2011); Yang and Nevatia (2012) collect positive features from the associating tracks and negative features from other tracks after primary associations task. These algorithms learn appearance model in batch and are not appropriate for online learning. In this situation, the sparse particle based tracker is the most popular method Gordon et al. (2004); Czyz et al. (2005).

A Gaussian based model is proposed to discriminate background and object and use their weight to assign weights to each clusters in Xiao et al. (2015). Particle filter methodology is applied to the clusters for score based tracking of objects across the frame. In Wang et al. (2013) represent sparse representation of an object and uses $\ell 1$ regularization into the PCA reconstruction to track objects. For further details, we refer Wu et al. (2013) that presents a list of benchmark datasets

and state-of-art methods used in object tracking. Weight Adjusted Particle Swarm Optimization ($WASPO$) Liu et al. (2016) is designed to maintain particle diversity and prevent premature convergence.

Although object tracking based on color histogram appearance models can achieve efficient tracking through partial occlusion and pose variation, tracking success or failure depends primarily on how distinguishable the object is from its surroundings. Surprisingly, most tracking applications use a fixed set of features, determined a priori Stern and Efros (2002). These approaches ignore the fact that it is the ability to distinguish between object and background that is most important and that appearance of both object and background will change as the target object moves.

Collins et al. (2005) presented an effective method for continuously evaluating multiple features while tracking and for selecting a set of features that improve tracking performance. They have developed an online feature ranking mechanism based on applying the two-class variance ratio to log likelihood distributions computed for a given feature from samples of object and background pixels. This feature ranking mechanism is embedded in a tracking system that adaptively selects top-ranked features for tracking. The result is a system in which the features used for tracking and the appearance models of object and background both evolve over time. Although the variance ratio is a computationally efficient mechanism for selecting tracking features, it does not take into account the spatial distribution of background values in the weight image and thus does not appropriately penalize features that produce spatially-correlated background clutter or strong distractors.

Babenko et al. (2011) presented a novel way of updating an adaptive appearance model of a tracking system. They have argued that using Multiple Instance Learning (MIL) to train the appearance classifier results in more robust tracking and presented an online boosting algorithm for MIL.

Danelljan et al. (2014) proposed an accurate scale estimation approach for visual tracking. Their method learns discriminative correlation filters for estimating translation and scale independently. This scale estimation approach is independent, and it can be incorporated in any tracking method lacking this component.

Danelljan et al. (2017) investigate the problem of accurate and robust scale estimation for real-time visual tracking. They have proposed a novel scale-adaptive approach for accurately estimating the size of the target. This approach is based

on learning separate discriminative correlation filters for translation and scale estimation. The explicit scale filter is directly learned from samples of the appearance change induced by scale variations. Furthermore, they have also discussed and proposed strategies to reduce the computational cost of proposed tracking approach. They have used a larger target search space without sacrificing real-time performance.

2.3 Synopsis Video

Techniques of video analysis in literature are broadly classified into two categories: Static image based summarization to generate a sketch of all activities in original videos and dynamic content based video summarization Lee and Grauman (2015). In static image based method, each shot is represented by keyframes, which are selected to generate a representative image. One of the examples of static image based summarization is video mosaic in which video frames are found using region of interest which are stitched together to form a resulting video. Another form is video collage in which single image is generated by arranging region of interest on a given canvas. Storyboards and narratives are some more basic form of image based summarization. However, static image based methods generate resulting summary in less space, but here it does not take care of temporal dependencies between important events. Additionally, researchers also want to maintain the resulting summary visually more appealing than watching static images.

As an example of dynamic content based video summarization method Veltkamp et al. (2013), video synopsis condenses video content in both spatial and temporal dimensions and present short video that helps in fast browsing.

Video synopsis presents some limitations as it requires large memory area to store foreground and background regions. While video synopsis save space it does not maintain consistency between different objects, also the pleasing effect of a video is highly dependent upon the length of the final synopsis. Some more examples of dynamic methods are video fast-forward, video skimming, space-time video montage method and video narrative where selected frames are arranged in order to form a highly condense video.

The overall framework of generating video synopsis using energy minimization is given by Rav-Acha et al. (2006). They have presented dynamic video synopsis, where most of the activity in the video is condensed by simultaneously showing

several actions, even when they originally occurred at different times. One of the resulting video representation is known as stroboscopic movie, where multiple dynamic instances of a moving object are played simultaneously. The synopsis video is also an index of the original video by pointing to the original time of each activity. Video synopsis can be applied to create a synopsis of endless video streams, as generated by webcams and by surveillance cameras. A query that could be answered by the system may be similar to "I would like to watch in one minute a synopsis of the video from this camera captured during the last hour" or "I would like to watch in five minutes a synopsis of the last week", etc. Responding to such a query, the most interesting events ("tubes") are collected from the desired period and are assembled into a synopsis video of the desired length. This process includes two major phases: 1) an online conversion of the endless video stream into a database of objects and activities (rather than frames) and 2) a response phase, generating the video synopsis as a response to the users query.

Pritch et al. (2008) presented two approaches of generating video synopsis: one approach uses low level graph optimization, where each pixel in the synopsis video is a node in this graph. This approach has the benefit of obtaining the synopsis video directly from the input video, but the complexity of the solution may be very high. An alternative approach is to first detect moving objects, and perform the optimization on the detected objects. While a preliminary step of motion segmentation is needed in the second approach, it is much faster, and object based constraints are possible. The activity in the resulting video synopsis is much more condensed than the activity in any ordinary video.

Although video synopsis technology is presented for fast browsing a day's worth of video in several minutes. However, for most existing solutions, motion structure in original videos may be destroyed even considering the temporal consistency of related objects.

Fu et al. (2014) proposed a solution to maintain temporal consistency of related objects by measuring the sociological proximity distance to find an interaction between objects. To maintain the important context cues, they have proposed an online motion structure preserved synopsis approach, which can preserve behavior interactions between different objects in the original video while condensing as much content as possible. In their work they have employed a hierarchical fashion to efficiently search an optimal solution for the problem of video synopsis, in which both the spatial collision and the temporal consistency are considered.

Embedding this information, the final synopsis video could condense as much activities as possible while maintaining their behavior interactions. However, due to the introduction of behavior interaction, the optimization problem appears to take more computational time than basic synopsis generation technique proposed by Rav-Acha et al. (2006).

Lee et al. (2012) proposed the method to generate video synopsis by discovering important object from the egocentric video. Li et al. (2016) proposed an approach of generating synopsis video by scaling down the objects.

The term nonchronological refers to the random order of events while arranged in synopsis video. If events or activities are shown in chronological order, they are arranged to be shown in the order in which they happened. A non-chronological synopsis video is a video in which events are not shown in series of time order. It is required to achieve compression criteria like showing activities happening in original video in a certain limited amount of time. While arranging activities over the spatial domain, we select trajectories of maximum length for maximum utilization of space. It may lead to a synopsis video that contain events in an order other than the order in which they occurred in the original video.

2.3.1 Activity Analysis

It is most desirable to have a method that can automatically analyze a video and generate various types of content related information, such as segmenting specific activities and video indexing. This information can help in improving video browsing and searching experience by producing high-quality video synopsis as suggested by Money and Agius (2008a) and in efficient online video indexing Chen et al. (2015). A number of video based applications require an automatic activity detection technique, some of them are video summarization (e.g., Money and Agius (2008b)), event detection (e.g., Morris and Hogg (2000)), or content-based image retrieval (e.g., Dhar et al. (2011)).

Activity analysis in this work is divided into three research topics. The first step is to segment space belonging to a target by subtracting it with background model. The second includes work that focuses on determining target region across the frames. Having information related to object movement for activity analysis is essential. The third task includes works that focus on determining the importance of an activity in video. It can benefit in generating effective synopsis video by estimating video timeline to include important activity (e.g., more effective

video browsing user experience). The target importance is estimated by computing its deviation from normal behavior when performing video content analysis (e.g., interactions with other objects in the window, object deviating from normal path and significant variation in size and velocity). This work has not made any category-specific assumption and thus can generalize for different categories of videos.

In video content analysis important activity estimation is a primary task to be accomplished properly. Many research works are based on inferring the user feedback in deciding importance of the activity in a video. The collective reactions of users are used to assign ratings to video content Bao et al. (2013). Similarly, Wu et al. (2011) prefer users choice of activities while generating video abstract. A more efficient way of selecting the important scene by using EEG headset is proposed by Shirazi et al. (2012). Zen et al. (2016) proposed user feedback through crowd mouse activity analysis technique. The cost of recruiting annotators and setting up devices to collect user reaction, also the decision based on collective user opinions is not effective enough. Due to above issues, these approaches do not seem to be suitable for online activity analysis from a video. Target behavior based content analysis techniques are having the advantages of scalability (i.e., they can perform online over large sets of video database) and generalize for being applied to a variety of video classes.

2.3.2 Video Content Analysis

The research work in this category focuses on using a prior information of video content for deciding the interestingness of the activities. The applications of different audio and visual features along with different machine learning techniques are the primary consideration of approaches in this category. Potapov et al. (2014) assign weights to a particular scene based on prior information about video category and its semantic taxonomy. Other similar methods determine the importance of scene based on similarity with image and videos available over the web Sun et al. (2014); Mazloom et al. (2015). These methods are category specific and also require prior information to decide the importance of an event.

2.3.3 Abnormality Detection

The long duration videos are not suitable for manual annotation of activities. Thus it requires an automatic method of detecting important activities. Different methods have been proposed for the detection of important activities in a video. For example, Morris and Hogg (2000) consider interesting event as statistical outliers. In many important activity detection algorithms such as Morris and Hogg (2000); Stauffer and Grimson (2000), a model is used to represent the normal behavior in the video. The behavior not belonging to these models is considered as abnormal or important. In most of the video based activity detection approaches, specific or low-level features are provided to apply with some machine learning techniques. It is required to have labeled training data to generate a model of normality, which is not available in sufficient quantity and quality.

2.4 Summary

In this chapter, we presented the literature survey of fundamental phases of synopsis video generation. Here, we have discussed the various techniques used for moving object detection, especially background subtraction. This step is a critical step in activity analysis because the result of later steps is primarily dependent on it. Most of the background subtraction methods focus on specific issues related to change detection or based on the assumption that background is stationary or having objects that show uniform motion.

In real-life situations an effective background subtraction method should be able to deal with dynamic background (permanent change in background geometry), noise (due to image capturing process), gradual and sudden illumination changes (scene captured at different times of the day or night or at different locations like indoor or outdoor), shadows (introduced by moving objects but are not considered as areas of interest), and small sized moving objects in background (moving tree branches or leaves due to wind).

After object detection, multi-object tracking is the inherent part of video analysis systems. We have surveyed the various existing techniques of multiple object tracking. Our review of study is that proper feature selection in object detection and tracking is a crucial part that may influence the result of tracking significantly. Appearance-based learning technique of multiple object tracking generate better results than data association approach, but it becomes computationally expensive

with the increase in a number of targets. Long-term occlusion and miss detection are two major issues with online multi-object tracking methods that tend to generate fragmented trajectories.

Next, we surveyed the traditional methods used for segmenting the important activities present in a video. As these methods are using labeled data for training of activity pattern, external knowledge combined with task-specific assumptions or constraints are used to guide the learning process to converge to a reasonable result. In many cases, the same spatial-temporal smoothness constraint is utilized in object discovery, adapting object detectors to video, and learning unsupervised representations. In this thesis, we further extend the same high-level idea of utilizing external knowledge and internal constraints to multi-object tracking and activity detection.

In next chapter, an adaptive approach to represent a background model is explained. It is pertaining to state-of-the-art detection systems, as well as to the algorithms used for object detection, considering many of relevant algorithms provided in this chapter.

Chapter 3

OBJECT DETECTION

Object detection is the process which divides a digital image into multiple segments (set of pixels) termed as region of interest whose pixel shares certain visual characteristics. Each application that benefit from keen intellectual video processing has different needs and thus requires different treatment. Moving objects are a prevalent thing between them all. Thus, detecting moving objects such as people and vehicle in video is the primary step since it provides a focus of attention and simplifies the processing on subsequent analysis steps. Most moving object detection methods endeavor to locate connected regions of pixels that represent the moving objects within the scene. Sometimes the region mark the boundary of the object is known as bounding box. Different approaches include frame-to-frame difference, optical flow techniques and background subtraction.

In this chapter we present an adaptive multi-layer background subtraction method by implementing various improvements over fundamental Codebook model. These improvements are:

1. To control learning rate of background model each codeword is assigned a weight using Sigmoid function.
2. The adaptive color distance measure is applied to compute the decision threshold for foreground pixel classification.
3. As background may have the spatial motions by considering spatial context shared between neighboring pixels, so a random neighbor selection policy is followed.

4. The proposed method reduces erroneous positive pixels detected conventionally as ghost region by maintaining codeword belonging to uncovered background region in background codebook.

3.1 Foreground Segmentation

The real-time segmentation of moving object is an fundamental and critical task in many computer vision application like visual surveillance systems. Super pixel region with sufficient difference between consecutive frames in their color space are treated as foreground and static region is termed as background. Background subtraction becomes an effective and obvious choice in segmenting moving objects in surveillance videos. Background subtraction segments moving objects present in a scene by subtracting current frame from background model of a scene. To detect moving object background subtraction approaches follow a three step process: background model initialization, background subtraction and model updation. Figure 3.1 gives an overview of the whole process.

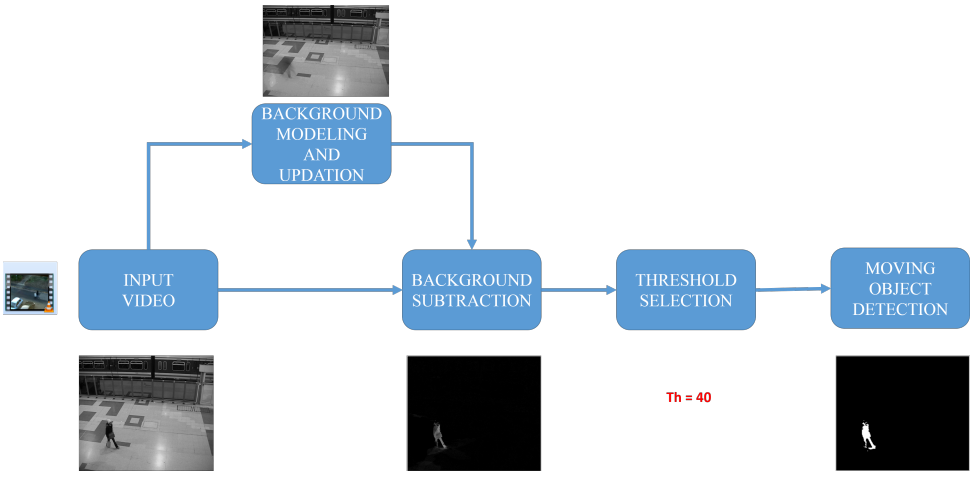


Figure 3.1: A systematic overview of the process representing moving object detection background subtraction methodology. The complete process follow three steps as background modeling, background subtraction and model updation.

Background initialization is the most critical step for generating effective output. Here the background model is generated using initial sequence of frames which is represented by using a static image or any statistical model. Background subtraction is generating the difference image by subtracting the corresponding pixels belonging to background model (i.e. reference image or statistical model) and the current image. Although initial background model can be effective in a situation

when background is static, it is required to update background representation if video is having dynamic characteristics.

Most of the background subtraction methods are based on an assumption that background is static or having objects that show uniform motion. In real life situations an effective background subtraction method should be able to deal with dynamic background (change in background geometry), gradual and sudden illumination changes (scene captured at different times of the day or night or at different location like indoor or outdoor), shadows (introduced by moving objects but are not considered as areas of interest), noise (due to image capturing process) and small sized moving objects in background(moving tree branches or leaves due to wind).

3.2 The Fundamental Codebook Model

Codebook model Kim et al. (2005) presents a quantization approach to represent dynamic background model using a set of codewords in compressed form. It reduces the memory requirement by building background model for each pixel as codebook consisting one or more codewords as $M = \{c_1, c_2, ..., c_m\}$. The fundamental codebook model efficiently detects moving objects using innovative color distance measure. Each codeword c_k, where $k = \{1, 2, ..., m\}$ is made up of a color vector $v_k = \{\overline{r}_k, \overline{g}_k, \overline{b}_k\}$ and a 6 tuple $aux_k = \left(\check{I}_k, \hat{I}_k, f_k, \lambda_k, p_k, q_k\right)$. Where, $\check{I}_k$ and $\hat{I}_k$ denotes minimum and maximum brightness respectively, f_k is used to represent frequency with which codeword k has occurred, λ_k is Maximum Negative Run Length ($MNRL$) (i.e., It represents the maximum number of subsequent frames for which codeword does not match with a pixel value.), and p_k and q_k are first and last access time of the k^{th} codeword.

Figure 3.2 pictorially represents a tuple structure of codebook of a pixel and Figure 3.3 shows the color vector using three color channels R, G and B.

Here, the number of training frames and the threshold value for adding codeword to background model M depends on the density of the moving object during initial frame sequence. The Codebook model can be trained using a minimum number of frames if there are no moving objects, but if it contains no ideal frame it requires an initial frame sequence for training. Initially, first 100 frames are used for training, where each pixel value x_t is compared to find a match with the codeword present in the current codebook using color distance measure and the logical brightness

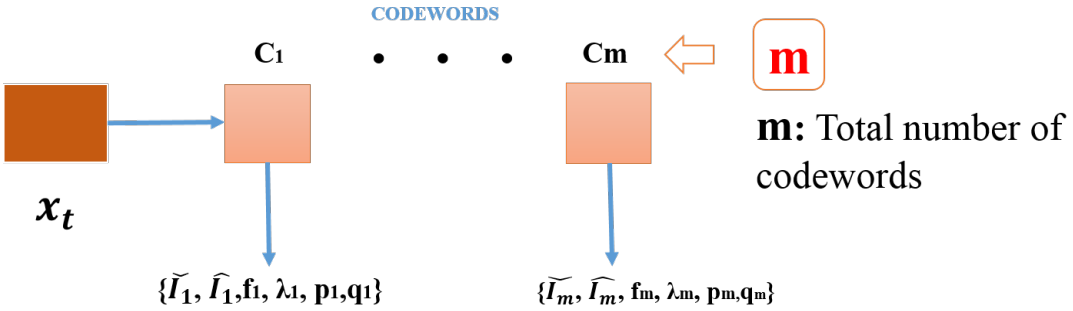
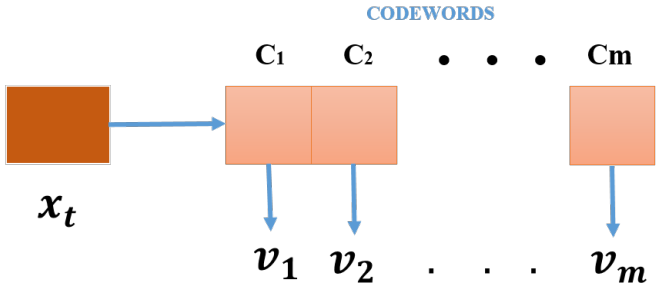


Figure 3.2: An illustration of background codebook model of a pixel consisting multiple codewords. Here input pixel is represented by x_t, c_m represent codewords present in the background model of pixel x_t, where m denotes the number of codeword. Each codeword is made up of 6 tuple $aux_k = \left(\check{I}_k, \hat{I}_k, f_k, \lambda_k, p_k, q_k\right)$.



$$v_k = \{\bar{r}_k, \bar{g}_k, \bar{b}_k\}$$

Figure 3.3: An illustration of color vector v_k of codewords present in background model of a pixel x_t. There is a color vector for different channel as $(\bar{r}_k, \bar{g}_k, \bar{b}_k)$.

function. Each pixel $x_t = (R, G, B)$ is compared with the codeword in M using color distance measure as expressed in Equation 3.1 Kim et al. (2005).

$$Colordist(x_t, v_k) = \sqrt{(R^2 + G^2 + B^2) - \frac{\left(\bar{r}_k R + \bar{g}_k G + \bar{b}_k B\right)^2}{\bar{r}_k^2 + \bar{g}_k^2 + \bar{b}_k^2}} \tag{3.1}$$

This color distance measure as explained above is sensitive to brightness change over the pixel. To compensate the brightness change, the minimum and maximum value of brightness is calculated during codebook updation. A logical brightness function is used to find pixel values within the range, as defined in Equa-

tion 3.2 Kim et al. (2005) .

$$Brightness\left(I, \langle \check{I}, \hat{I} \rangle\right) = \begin{cases} True & \text{if} \alpha \hat{I_k} \leqslant \|x_t\| \leqslant min\{\beta \hat{I_K}, \check{I}_k/\alpha\}, \\ False & \text{otherwise} \end{cases} \tag{3.2}$$

Where, α and β are the fixed parameters taken as 0.5 and 1.3 respectively, which is the brightness bound used to adapt illumination changes whenever a shadow is falling over an object.

After training, the codewords having frequency greater than threshold $Th = \frac{N_t}{2}$ are stored in M. Where N_t are number frames used to generate background codebook during training period. The pixel is classified as background if it match with any codeword present in background model of pixel. The matching operation include subtraction operation BGS_x for incoming pixel based on the two conditions defined as:

$$BGX_x = \begin{cases} True & \text{if } \{(colordist(x, c_m) \leqslant \varepsilon) \wedge (brightness(I, \langle \check{I}, \hat{I} \rangle) == True)\} \\ False & \text{otherwise} \end{cases} \tag{3.3}$$

where ε is the decision parameter used for separating foreground from background, is taken as 25 whenever there is a 30% or more variation between intensity of two entities.

As fundamental codebook model does not update background model, it is not enough for many useful situations where the background is not static. A multi-layered approach is suggested in Kim et al. (2005). After the initialization period of codebook construction for background model, for every new frame, the observed pixel color value x_t at each pixel position is compared against each codeword present in permanent background codebook model M. If pixel x_t matches with any of the codewords in M it is classified as background, and the corresponding codeword is updated. If none of the codewords in M match with current pixel it is classified as foreground. The codewords are then checked for matching codeword in non-permanent background codebook model H as defined by Sigari and Fathy (2008), if a match is found then the codeword is updated. If no match is found, a new codeword is created in H.

For the dynamic environment, where the number of moving objects is large the size of H keeps increasing. In the fundamental approach to maintaining memory

efficiency, the codewords are deleted or shifted from H to M if they satisfy the following conditions:

1. Equation 3.4 is used for deleting codewords from H (no codeword match for long duration), when λ exceeds threshold T_H.

$$H \leftarrow H - \{h_i | h_i \in H, if \lambda_{h_i} \geq T_H\} \tag{3.4}$$

2. Equation 3.5 is used to shift codeword h_i from cache codebook H to codebook M if it finds match with the pixel in more than T_{add} consecutive frames i.e. $freq_{h_i} = T_{add}$.

$$M \leftarrow M \cup \{h_i | h_i \in H, if freq_{h_i} \geq T_{add}\} \tag{3.5}$$

3. The codeword is deleted from M if it satisfies Equation 3.6.

$$M \leftarrow M - \{c_i | c_i \in M, if \lambda_{c_i} \geq T_{delete}\} \tag{3.6}$$

Further improvement in the codebook model is suggested as an addition of another cache codebook H, which is used in layered codebook model Sigari and Fathy (2008) to update non-background pixels. Due to the compressed representation of background model, codebook has attracted the attention of many researchers Guo et al. (2011); Xu et al. (2011); Sun et al. (2011); Syed et al. for improvement. The performance of Codebook model deteriorates when the background contains objects showing dynamic characteristics.

3.3 The Proposed Modified Codebook Model

The following subsections presents the strategy used to choose a decision threshold value adaptively. The background updation is explained followed by the estimation of appropriate value for adding codewords to the background model. Further, the spatial context is included by adding codewords of the uncovered background region by applying random neighbor selection policy.

3.3.1 Adaptive detection threshold

In primary codebook model detection threshold ε is kept constant for all video sequences. As different video sequence may have different lighting conditions depending upon various locations and time, when the sequence is captured. In between indoor and outdoor videos, there are significant differences in the color distance over a pixel in successive frames in the same way video obtained during day or night. Ideally, the detection threshold should be increased for a highly dynamic background like water surface, waving tree leaves, lightning change etc., so that they are not included in a foreground. Likewise, for static background, a low value should be assigned to ε to allow detection of even small changes in scene and to deal with camouflage.

Figure 3.4(a) Kim et al. (2005) is a pictorial representation of having fixed decision parameter δ and Figure 3.4(b) shows the modified varying decision parameter ε_{adap} over pixel i. Where, $\check{I}_k$ and $\hat{I}_k$ denotes minimum and maximum brightness respectively. I_{low} and I_{hi} applied as upper and lower bound over decision parameter. X_t represents the intensity value of input pixel. Instead of using constant distance parameter ε that build cylinder around mean color distance vector, an adaptive decision threshold ε_{adap} is used in this work. It creates a cone with the center as μ_m and radius that depends upon decision threshold ε_{adap} as shown in Figure 3.4(b).

By using a cone instead of a cylinder along the color distance vector, video having dark color are forced to have small variance, and the high color component will generate high variance leading to high decision threshold. While cylinder representation used fixed threshold value(δ) as decision parameter which leads to false detection, a cone representation uses an adaptive decision threshold $\varepsilon_{adap} = \frac{\delta}{\sigma}$ which compensates the variance of color distance measure. Normalizing the color difference with σ stabilizes the color distance in situations with low lighting conditions and static background. Even, for low difference in color distance measure a small value of σ does not affect δ very much.

For each codeword $c_i, i = 1....m$ consisting of color vector $v_i = (\overline{r}_i, \overline{g}_i, \overline{b}_i)$, the value of each color component of color vector v_i is adaptively updated as:

$$\overline{r}_i = \gamma \overline{r}_i + (1 - \gamma)R \tag{3.7}$$

$$\overline{g}_i = \gamma \overline{g}_i + (1 - \gamma)G \tag{3.8}$$

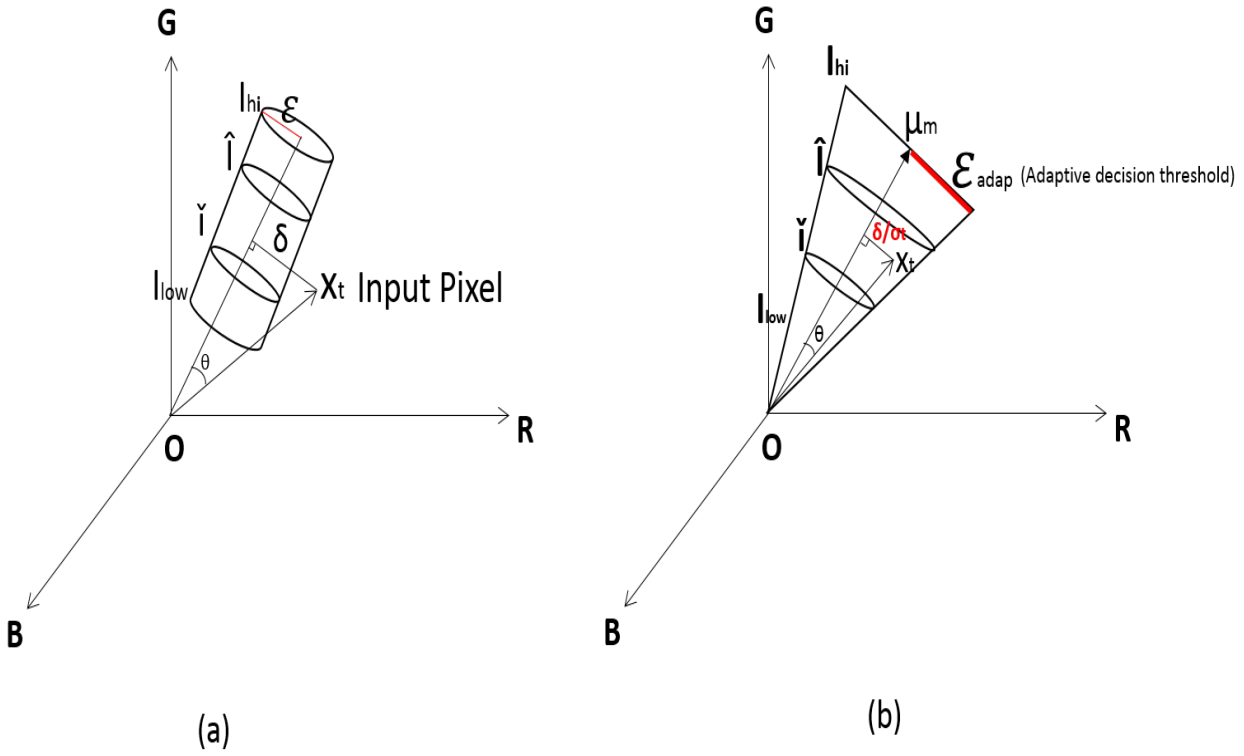


Figure 3.4: Representation of color distortion measure:(a) The cylindrical representation of mean color vector and distance measure. (b) The modified conic representation of mean color and distance measure as varying decision parameter threshold ε_{adap}.

$$\overline{b}_i = \gamma \overline{b}_i + (1 - \gamma)B \tag{3.9}$$

Where γ is the learning rate. It is used to update the color information of a codeword with a matching pixel to add it to the background. The γ is assigned value in the range between 0 and 1, a higher value results in a slow updation of mean color value and smaller value leads to faster color updation for the background. In order to adjust to changing background situations, an adaptive color model is proposed to habituate to variations in a scene, where the color distance is modified to deal with background dynamics. The mean color distance value is obtained as shown in Equation 3.10.

$$\overline{\delta}^2_{m,t} = \rho \overline{\delta}^2_{m,t-1} + (1 - \rho)\delta^2_{m,t} \tag{3.10}$$

Where $\overline{\delta}_{m,t}$ denotes mean color distance over pixel x at time t and $\delta_{m,t}$ represent current value of color distance as calculated in Equation 3.1. The ρ denotes learning rate. Similar to γ the range of ρ lies between 0 and 1.

Different lighting condition and global illumination variations over a pixel result in deviation in the color distance even without any moving object that may cause false detection or miss actual targets. The variance $\sigma^2_{m,t}$ in the color distance, is calculated as the difference between mean color distance and color distance over current pixel x_t as defined in Equation 3.11.

$$\sigma^2_{m,t} = (\overline{\delta}_{m,t} - \delta_{m,t})^2 \tag{3.11}$$

To adaptively update to the variance corresponding to matched codeword in background model the value of $\sigma^2_{m,t}$ of each codeword is formulated as defined in Equation 3.12.

$$\sigma^2_{m,t} = (1-\gamma)\sigma^2_{m,t-1} + \gamma\sigma^2_{m,t} \tag{3.12}$$

To make model adaptive by normalizing for zero mean and unit variance the color distance $\delta_{m,t}$ is modified as:

$$\delta_{m,t} = \frac{(\delta_{m,t} - \overline{\delta_{m,t}})}{\sigma_m} \tag{3.13}$$

Finally the decision threshold value for codebook m at time t, $\varepsilon_{adap}(m,t)$ is dynamically updated as follows:

$$\varepsilon_{adap}(m,t) = \tau \times \sigma_{m,t} \tag{3.14}$$

Here, τ is a fixed parameter, as decision threshold $\varepsilon_{adap}(m,t)$ is distributed normally around mean color distance $\delta_{m,t}$ and unit variance as σ. To classify foreground pixel τ is $> 2.5 \times \sigma$ (98.75% variance of the background pixels probability density function). Furthermore, the threshold is bounded by the upper and lower bounds as $T_{Lower} \leq \varepsilon_{adap}(m,t) \leq T_{Upper}$, so that, the value cannot go beyond explicit limits. For highly dynamic environments or illumination variations, the significant difference occurs in the color distance of pixels leading to false detection. This is taken care of by normalizing the color measure. Figure 3.5 depicts the value of adaptive parameters calculated over the pixel $(444; 325)$ across 3911 number of frames. This Figure represents the variation in mean color distance v_t, variance σ^2, adaptive color distance δ, and adaptive decision threshold $\varepsilon_{adap}(m,t)$ explained in Equation 3.10, 3.12, 3.13, 3.14 respectively. It can be clearly seen that

the $\varepsilon_{adap}(m,t)$ (decision threshold) varies across the mean color distance value in proportion to σ that denotes the deviation in color distance measure.

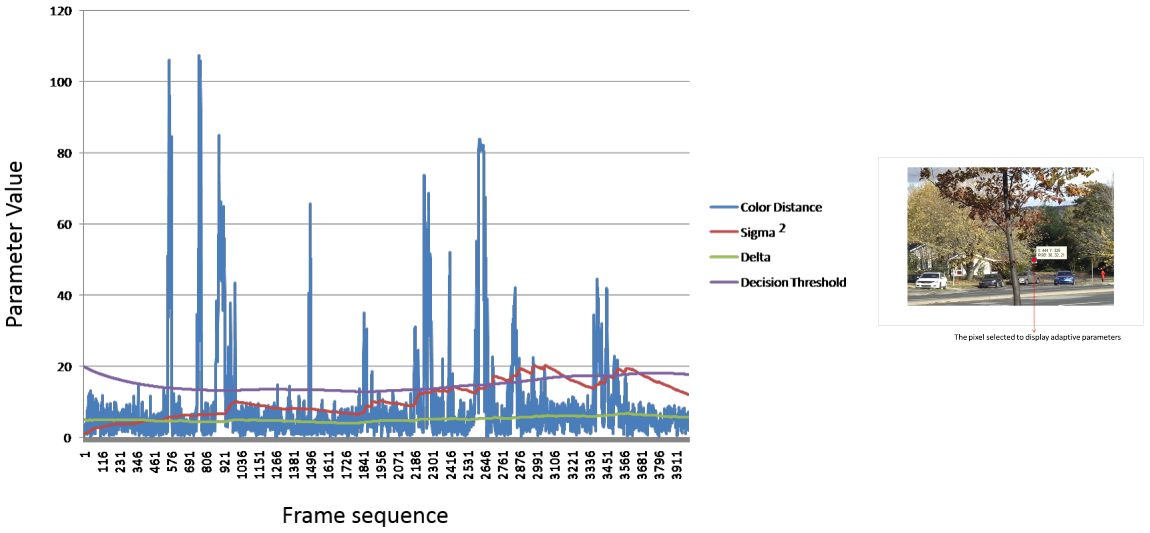


Figure 3.5: Adaptive parameters: color distance measure δ, *sigma* and decision threshold ($\varepsilon_{adap}(m,t)$) for pixel specified in image at location (444,325) across the frame sequence. The horizontal axis denote the frame sequence and vertical axis is showing the value of each parameters.

3.3.2 Adaptive background model updation

As it can be seen that background updation in primary codebook method is a simple linear process. The addition of codeword to background model depends upon the frequency of pixel matching with codeword or the longest time length for which the codeword is not updated. This linear approach of background learning solely depends upon the selection of threshold T_{add} and T_{delete}. In traditional methods empirical approach of increasing frequency $freq$ and negative run length λ is used, which is based on the assumption that objects will always show uniform motion or remain stationary and does not consider the instability in real world frame sequence. The real life situation differs from ideal behavior as slow moving objects may get included in the background model.

There is the tradeoff in a selection of threshold value. A small threshold value T_{add} may make the system vulnerable to noise and can also lead to the inclusion of small moving objects in the background, while a large value slows down the system, making it less responsive. To make the learning process adapt to the gradual background changes smoothly and not to be affected by noise and foreground objects an optimal value should be selected for T_{add}.

We present a novel formula based on sigmoid function to control the growth rate of a model. In order to make system adaptive to dynamic background environment, weight W_i is assigned to each codeword c_i which helps the system to adapt to changes by adding or deleting codewords from H (codewords belong to foreground region) and M (background codebook). A sigmoid function $sig(x, [l, m])$ as defined by Equation 3.15 is used to update weights assigned to codeword.

$$W_{c_i}^t = \begin{cases} W_{c_i}^{t-1} + Sig(freq, [l, m]) & \text{If codeword} c_i \text{match with pixel} x_t, \\ W_{c_i}^{t-1} - Sig(N - q, [l, m]), & \text{otherwise} \end{cases} \tag{3.15}$$

Parameter l and m are kept fixed which decides the rate of convergence of the system, where l and m are assigned the value as .1 and 25 respectively. It reflects many natural or real world systems where learning exhibits a slow start until it reaches the confidence cap after which it accelerates the learning, which finally stabilizes when the codeword stays for extended periods over time. For example, if an object is introduced in H, initially it will learn slowly, but if it remains there for a long time, then the rate of growth will increase rapidly, leading to the addition of codeword in the background model. As sigmoid function satisfy most of the moving object situations for updating weight, it can also be adjusted for other situations too.

If a codeword c_i with frequency $freq$ matches with pixel x_t, its weight gain is calculated using sigmoid function over frequency as defined by Equation 3.16.

$$Sig(freq, [l, m]) = \frac{1}{1 + e^{-l(freq-m)}} \tag{3.16}$$

The weight loss for a codeword without any match will be subtracted from its weight using Equation 3.17, which is computed using the length of time for which a match is not found $(N - q)$. Where N denotes current frame number and q is last frame number when codeword was matched with a background model.

$$Sig(N - q, [l, m]) = \frac{1}{1 + e^{-l((N-q)-m)}} \tag{3.17}$$

Figure 3.6 depicts the growth rate of codeword weight using our proposed approach. The parameter W is the learning rate of foreground codebook model H. When the updated weight value $W < (Th_{del})$, it is removed from the codebook.

When $W > Th_{add}$, move the codeword from cache codebook H to background codebook M optimally. The weight W is initialized with value 5, whenever a new codeword is added to the cache codebook H, where the value 5 is determined heuristically by performing experimental analysis over video sequence from dataset $CDnet-2015$, described in Section 3.4.0.1, with the aim of keeping the computational complexity of system optimal. Assigning a low value to W may lead to the removal of background pixel from codebook H and higher values keep foreground pixel in the codebook for a longer duration. Figure 3.7 shows the relationship of weight W, with parameters l and m used to decide the rate of convergence of the system.

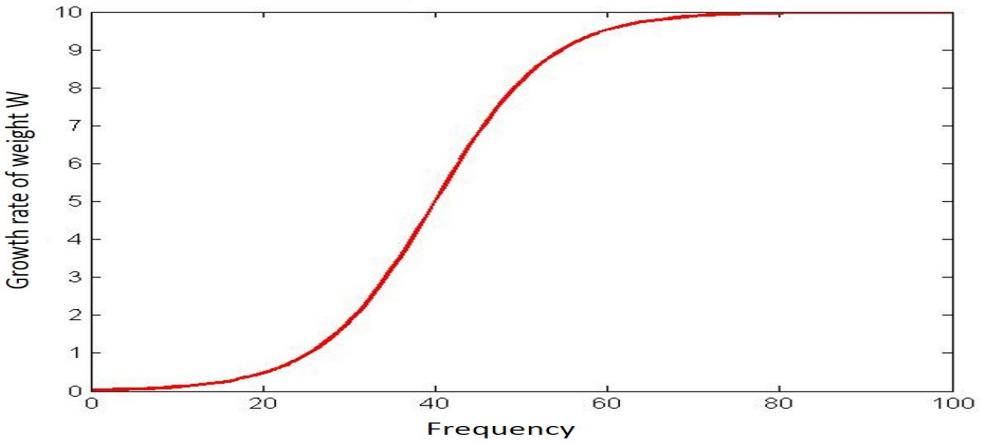


Figure 3.6: Graph representing weight gain using our proposed sigmoid function. It depicts the growth rate of parameter W as natural phenomenon (i.e., initially it will learn slowly, but if it remains there for a long time, then the rate of growth will increase rapidly, leading to the addition of codeword in the background model.)

By adjusting l and m values we can control growth rate of the learning system as shown in Figure 3.7. The l value is used to control how fast learning rate will be possessed by system, and m value is used as a stability criteria. For example when a moving object become stationary it should become part of background after minimum time, this situation is control by parameter m and l. The value of parameter l can be set between 0.1 and 0.5 and m lies between 10 to 80. For fast changing environment low value for l and m is preferred, while for stable environment high value is assigned to them.

It has been observed that some of the codewords in background model does not participate in the selection of background frequently. We need to decide the number of codewords sufficient to represent background efficiently. It is done by filtering

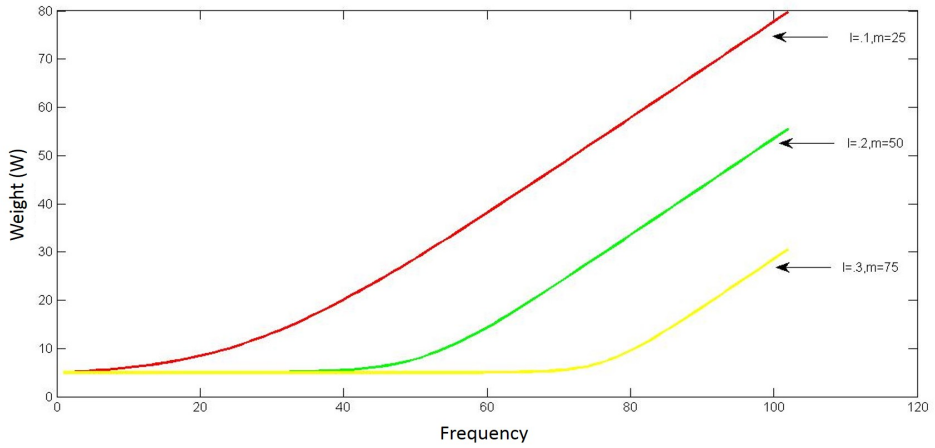


Figure 3.7: Effect of Weight W updation with parameters l and m to control the learning rate of background model. Different combinations of l and m can be used to regulate the learning of weight W.

the low ranking codewords, which also scales down the computational complexity significantly. Finally, all the codewords belonging to the background model M are normalized by dividing the weight of each codeword with the sum of all the codewords weight as defined by Equation 3.18.

$$W_k^t = \frac{w_k^t}{\Sigma_{k=1}^{c^M} w_k^t} \tag{3.18}$$

All the codewords c_k^M are arranged in decreasing order according to their normalized weights, the number of codewords to be linked with background model are the first N^t codewords that satisfy Equation 3.19.

$$C_k^M = argmin_n \left[\left(\Sigma_{k=1}^n W_k^t \right) > T_B \right] \tag{3.19}$$

Where W_k^t denotes weight of the normalized sorted codewords in background model and T_B is a threshold used to determine what fraction of codewords are enough to represent background efficiently. Here T_B is taken as 0.8 to maintain competent codewords in codebook.

3.3.3 Random Spatial codebook selection

It has been observed that in the dynamic background, pixels share space with their neighbors by oscillating around their region. So codebook of neighboring

pixels is locally dependent. Finding a match in pixel codebook with all of its neighbors increases the complexity of the system drastically. For example, a 4-connected neighborhood, with an average 3 codewords in their background model, requires 12 comparisons for color and intensity each. With an aim of incorporating spatial context for local changes, keeping the computational complexity low, a random neighbor selection policy is followed as explained in Algorithm 5. Where a foreground pixel x_t is randomly compared with codeword in background model M_y; if a match is found then, it is marked as background. Where M_y is the background model of the random neighbor of input pixel x.

Algorithm 1 Random neighbour pixel selection

```
for each input pixel x at frame t, x_t = (R, G, B), ||x|| = sqrt(R^2 + G^2 + B^2) do
   Find a matching codeword to x_t in background codebook M
   if found then
      Match=1, then update the codeword
   else
      match=0
      Select random(y) ∈ N_x
      for each codeword in M_y do
         Try to find match with x_t
         if found then
            Match =1
         end if
      end for
   end if
end for
```

3.3.4 Uncovered Background

It has been observed that the uncovered background region shows color similarity with their neighboring pixels over spatial domain. For example, when a moving object becomes stationary it covers the portion of the actual background. Depending upon learning rate of codebook model the codeword of that real background will be removed when a maximum negative run length exceeds the threshold value. Later on, when this temporary background leaves that space, it creates a set of holes as falsely detected foreground. In the following subsections, we explain improvements over fundamental codebook models that shows increase in performance.

We have improved the multi-layered modeling proposed in Kim et al. (2005) by introducing one more layer for uncovered background region for each pixel, where

permanent backgrounds are identified and marked when the background codeword count exceeds the threshold limit. Traditional methods do not determine stable backgrounds. In fundamental codebook approach when a codeword does not match for a long time, then, its negative run length λ will increase, when it reaches a threshold value, it is removed from background model codebook.

By analyzing various video datasets, it has been observed that when an object stops moving, it becomes part of the background and covers the permanent background region. That uncovered background region may have spatial properties similar to its neighbors. Thus, we check for a match in its neighbors codewords N_x. A match found means it shares spatial context with its neighbors and hence is added to uncovered background codebook. The transition of codewords between different codebooks is shown in Figure 3.8. Initially all pixels are classified as foreground , when a codeword is created it is assigned to moving background codebook (i.e., foreground model) denoted as $\{H_1, H_2, ...\}$. If the current pixel intensity matches with any of the codewords present in the moving background, its frequency is incremented. When the frequency is $> Th_{High}$, it is shifted to static background codebook model (i.e., background model) represented as $\{C_1, C_2, ...\}$. If a codeword presents in the static background model with significant amount of time, it is assigned to uncovered background codebook model as U_1.

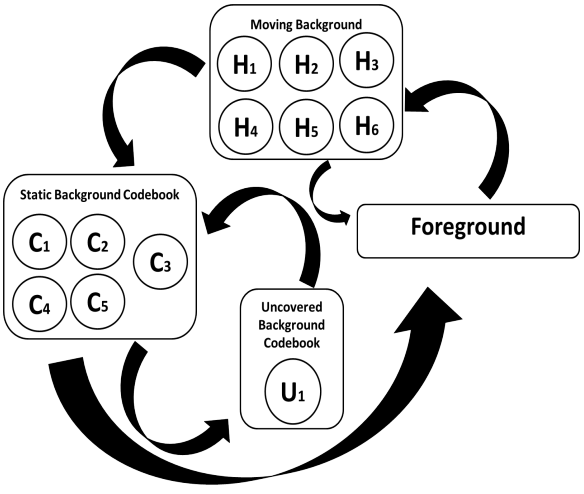


Figure 3.8: Transition diagram of codeword between different layers. Each pixel is initially consider as a foreground pixel. A codeword can move in or move out from one layer to other depends upon its maximum negative run length and frequency.

The procedure of including codewords belonging to uncovered background region in background model is outlined in Algorithm 6. It includes a codeword in static

background Codebook to uncovered background layer U if frequency of that codeword become greater than or equal to threshold value Th_{High}.

Algorithm 2 Identification of codeword belonging to uncovered background region.

for each input pixel x at frame t, $x_t = (R, G, B)$, $\|x\| = \sqrt{R^2 + G^2 + B^2}$ **do**
  Find a matching codeword to x_t in background codebook M
  if found **then**
    Match=1, then update the codeword
    if $freq_m \geq Th_{High}$ **then**
      for each codeword $M_y \in \{M_y | y \in N_x\}$ **do**
        Try to find matching codeword in M_y with x_t
        if found **then**
          Add match codeword in uncovered background layer U.
        end if
      end for
    end if
  end if
end for

For example, based upon the color information of neighboring pixels a region vacated by a car by moving out of a parking space will be added to the uncovered background as it shows similarity with background pixels in its neighbors as shown in Figure 3.9.

The assumption is that each uncovered background pixel shares color similarity with its neighbors and remains visible for a minimum time duration, represented by Th_{High}. Here, M_y denotes the codebook belonging to background model of pixel $y \in N_x$, where N_x are the neighbouring pixels of x. In this work, a 4-connected neighborhood is considered for keeping our method computationally efficient.

3.4 Experimental Results

This section details the experimental setup, the test sets, and the performance analysis metrics used in the analytical analysis of the proposed method and its comparison with some existing state-of-art techniques.

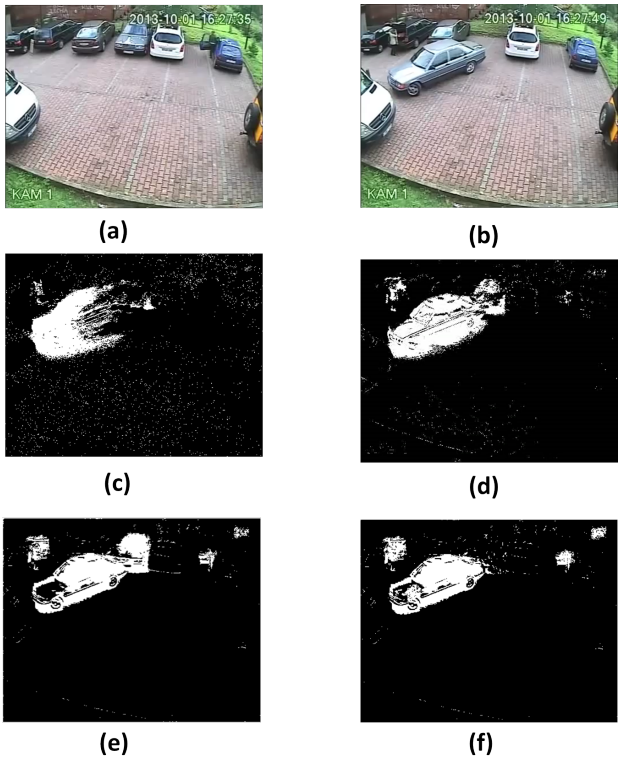


Figure 3.9: Figure shows an effect on number of false positive pixels detected behind a car when it moved out of a parking space. (a) Background image. (b) Current image. (c) GMM (d) ViBes (e) CB-Kim (f) Proposed method.

3.4.0.1 Dataset used for experimentation

The proposed method has been tested on a variety of video sequences taken from standard datasets used for change detection (*CDnet*) that are publicly available at Goyette et al. (2012). Testing is done on six video sequences belonging to dynamic background categories, where each frame sequence presents some challenges due to the dynamic background. The sequence has a resolution of 320×240 with many different background dynamics such as ripping water, waving trees and fountain. Manually annotated ground truth is available for all video sequences that is used for quantitative analysis.

3.4.1 Parameters

A number of parameters used are responsible for controlling the outcomes of proposed methodology. It is critical to understand and assign appropriate values to these parameter for generating better results. A detailed discussion about parameter used in this work is given here.

1. The number N of sequence frames for the training phase depends on how many static initial frames are available for each sequence. If no static initial frames are available, then a sufficiently high value for N should be chosen for initial background modeling. All the test video sequences have not used the initial 100 frames for quantitative analysis. These initial 100 frames are used by us for initial system training.

2. The initial weight vector (W) used for assigning newly created codeword is taken as 5 for all test sequences. This value is driven by experiments carried out to filter nonbackground objects entered into cache codebook H. The threshold value Th_{add} and Th_{del} to update background model is taken as 60 and 0 respectively.

3. The parameter value l and m used in proposed sigmoid function for weight updation is fixed to 0.1 and 25 for limiting learning rate of background model. The justification for which is illustrated in Figure 3.7.

4. In fundamental codebook approach the values for decision threshold ϵ_1 and ϵ_2 are taken as 25 and 15 respectively (as explained in Section 3.2). A high value for ϵ_1 limits inclusion of small objects in the initial background model during the training phase, similarly a low value of ϵ_2 is preferred for accurate foreground pixel detection after the training phase. In the proposed approach, we initialized both to 25 as our method is adaptive and finds the best values over time. The lower bound T_{Lower} and upper bound T_{Upper} for decision threshold is set as 15 and 35 respectively which has been determined empirically by studying the color distance vectors of dynamic videos as shown in Figure 3.11.

5. $\gamma = 0.005$ is used to control learning rate of decision threshold.

3.4.2 Methods considered for comparison

Proposed approach is compared with six other background subtraction methods as basic codebook model($CB - Kim$) Kim et al. (2005), Adaptive background mixture models ($GMM - Stauffer$) Stauffer and Grimson (1999), Improved adaptive Gaussian mixture model ($GMM - Zivkovic$) Zivkovic (2004), Multi-scale spatio temporal background model ($MS - STBM$) Lu (2014), Kernel density estimation($KDE-Mittal$) Mittal and Paragios (2004) and ($Vibes-Barnich$) Barnich and Van Droogenbroeck (2011).

3.4.3 Performance metrics

There are a number of performance matrices used in literature to compare segmentation quality and to evaluate the performance of background subtraction methods. The performance metrics often used in research papers are precision, recall, F-measure and Receiver Operating Characteristic (ROC) curve. In this thesis we report pixel-based evaluation method based on comparative measures, mostly comparing the foreground binary mask and ground truth reference image.The assessment of algorithms is based on the comparison of ROC curves. The primary aim of ROC curve is to focus on positive test results, both True positives and false positives. To generate the ROC curve true positive rate (TPR) and false positive rate (FPR) is calculated from six video sequences of category dynamic background from $CD-net$ database. TPR also termed as sensitivity or recall gives the ratio of detected true positives as compared to the total number of true positive in the ground truth as shown in Equation 3.20.

$$TPR = \frac{TP}{TP + FN} \tag{3.20}$$

Where TP is true positive, i.e., (the number of pixels where both ground truth and algorithm agree), FN is the total number of false negative i.e., (the number of pixels where ground truth marked as foreground are detected as background by the algorithm). FPR also termed as $(1 - specitivity)$ gives the ratio of detected FP as compared to the total number of background pixels identified in the ground truth as given in Equation 3.21.

$$FPR = \frac{FP}{FP + TN} \tag{3.21}$$

Here, FP denotes the total number of false positive i.e., (the number of pixels where ground truth marked as background, detected as foreground by the algorithm.),TN is the total number of actual negative i.e., (the number of pixels background pixels where both ground truth and algorithm agree).

The results of these pixel measures are reported in Table 3.1 used for performance evaluation. Higher values of TPR denotes that moving objects are segmented successfully. When we consider FPR the lower value is preferred as it shows the part of background detected as a foreground.

First and second videos (i.e. Fountain 1 and Fountain 2) shows the Quasi-periodic

Table 3.1: Performance metrics for evaluation of background subtraction algorithm

Video	Fountain 1		Fountain 2		Canoe		Boat		Overpass		Fall	
Methods\ Metrices	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
GMM1	0.80	0.013	0.87	0.00064	0.87	0.0036	0.76	0.002	0.83	0.00099	0.88	0.039
GMM2	0.75	0.012	0.84	0.00062	0.85	0.0027	0.70	0.0012	0.80	0.00074	0.86	**0.039**
ViBe	0.77	0.012	**0.88**	0.0007	0.92	0.003	**0.78**	**0.0007**	**0.88**	0.004	0.88	0.059
CB	0.75	0.015	0.88	0.00066	0.91	0.0036	0.70	0.0012	0.85	0.004	0.84	0.068
KDE	0.79	0.011	0.85	0.00047	0.83	**0.0019**	0.66	0.002	0.80	0.002	0.87	0.068
MSTM	0.49	**0.004**	0.85	0.00048	0.91	0.005	0.51	0.0039	0.82	0.0019	0.85	0.041
Pr. Ap.	**0.84**	0.005	0.87	**0.00041**	**0.95**	0.002	0.75	0.0011	0.82	**0.0007**	**0.91**	0.047

motion of fountains belongs to a background. Concerning TPR, the best result is achieved by proposed method. Higher TPR values and low value of FPR denotes that proposed method can handle Quasi-periodic motion and also detect foreground object successfully.

Learning rate of the model is an important issue in these situations so that model can adapt pixels in the background model. Proposed method uses sigmoid function to be used for learning background model that can successfully adopt periodic background motion.

Third and fourth video (i.e. Canoe and Boat) test algorithm performance in the presence of water waves that increases FPR, also in the fourth video (i.e. Boat) the boat color match with water surface result in camouflage that result in decrease in TPR value. It is observed that the proposed method gives the best result in the case of Canoe and performance of $Vibes$ is best in $Boat$ video. Adaptive threshold value increases chances of correct foreground detection with camouflage.

Next, two videos (i.e. Overpass and Fall) test algorithm performance with moving tree leaves over sufficient portion of the frame. Although $Vibes$ gives highest TPR value the performance of proposed method is better regarding FPR. The result of proposed method shows that small objects like tree leaves are filtered successfully.

3.4.3.1 ROC Curve

The *ROC* curve is a graph to visualize global classification performance of an algorithm. Any point on this curve is a relationship of classification response between pixel correctly classified and negative pixels incorrectly classified. It describes the relative balance between true positive and false positives. The measure of

specificity$(1 - FPR)$ and sensitivity TPR provide a general classification index, and both measures defines a single point in ROC curve for comparing the performance of different algorithms under the same conditions. The curve helps to compare the performance of the algorithms, a value located near to top left corner present better performance compared to other that are farther away. Figure 3.10 shows the ROC curve for all the algorithm. It is evident by area under curve AUC that the performance of the proposed approach is superior to other methods.

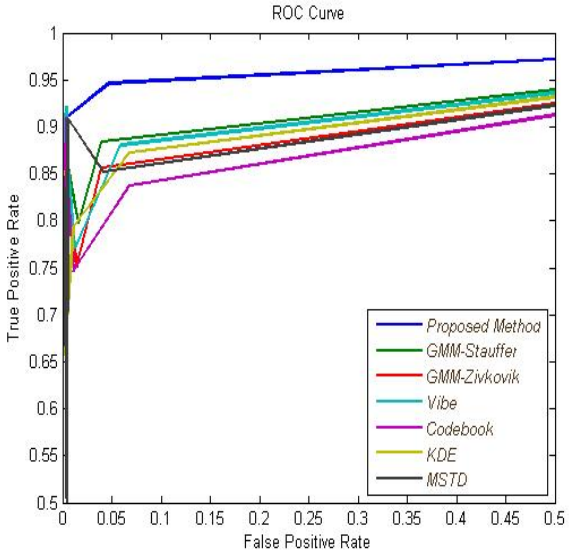


Figure 3.10: The ROC curve for comparison of performance analysis of background subtraction methods. It is evident by area under curve AUC that the performance of the proposed approach is superior to other methods.

Considering overall performance the Figure 3.11 shows graph by taking average of performance measure from all videos. The final result is compared for higher true positives and lower value of false positives. This can be verified by looking at point located near top left corner of the proposed approach.

Additionally, the performance of proposed method regarding $F - Measure$ is shown in Table 3.2. It can be verified that proposed method is better than other algorithms.

3.4.4 Qualitative Evaluation

For comparing qualitative results Figure 3.12 shows the change detection results for frame number 1500 belonging to video "$fall$".

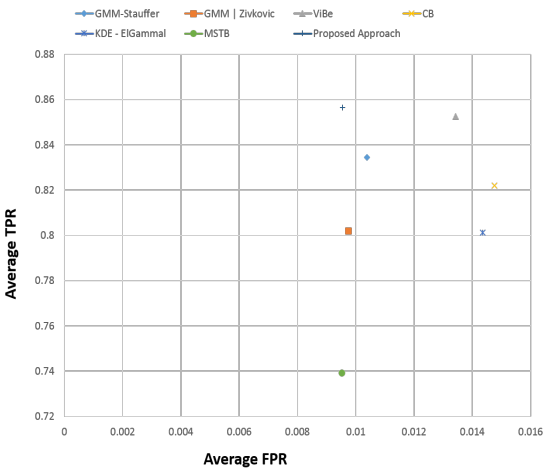


Figure 3.11: Plot of average value of FPR against TPR for overall performance analysis of background subtraction methods. The result is compared for higher true positives and lower value of false positives. Proposed approach outperform other methods and this can be verified by looking at point located near top left corner of the proposed approach.

Table 3.2: Performance metrics (F-measure) for evaluation of background subtraction algorithm

Video	Fount.1	Fount.2	Canoe	Boat	Overp.	Fall
GMM1	0.076	0.80	0.88	0.73	0.87	**0.44**
GMM2	0.081	0.79	0.89	0.75	0.87	0.42
ViBe	0.090	0.79	0.91	**0.83**	0.80	0.34
CB	0.099	0.81	0.90	0.74	0.79	0.29
KDE	0.105	0.82	0.88	0.63	0.83	0.31
MSTM	0.14	0.82	0.89	0.48	0.84	0.41
Pr. Ap.	**0.21**	**0.83**	**0.94**	0.78	**0.88**	0.39

3.5 Summary

As stated in the earlier sections moving object detection is a conventional issue for many computer vision applications that still need to be refined. This chapter presented an adaptive multi-layer codebook model for improving the quality of foreground segmentation in a video by proposing improvement in basic codebook model. These improvements are obtained by applying adaptive decision threshold to overcome challenges introduced due to background dynamics. By introducing adaptive decision threshold color distance is provided with a range information that makes this model robust against dynamic background in surveillance videos.

In this chapter we have stated that the cone shaped color distance measure instead of cylinder based color distance measure helps to achieve better accuracy against illumination variations due to its normalization by σ.

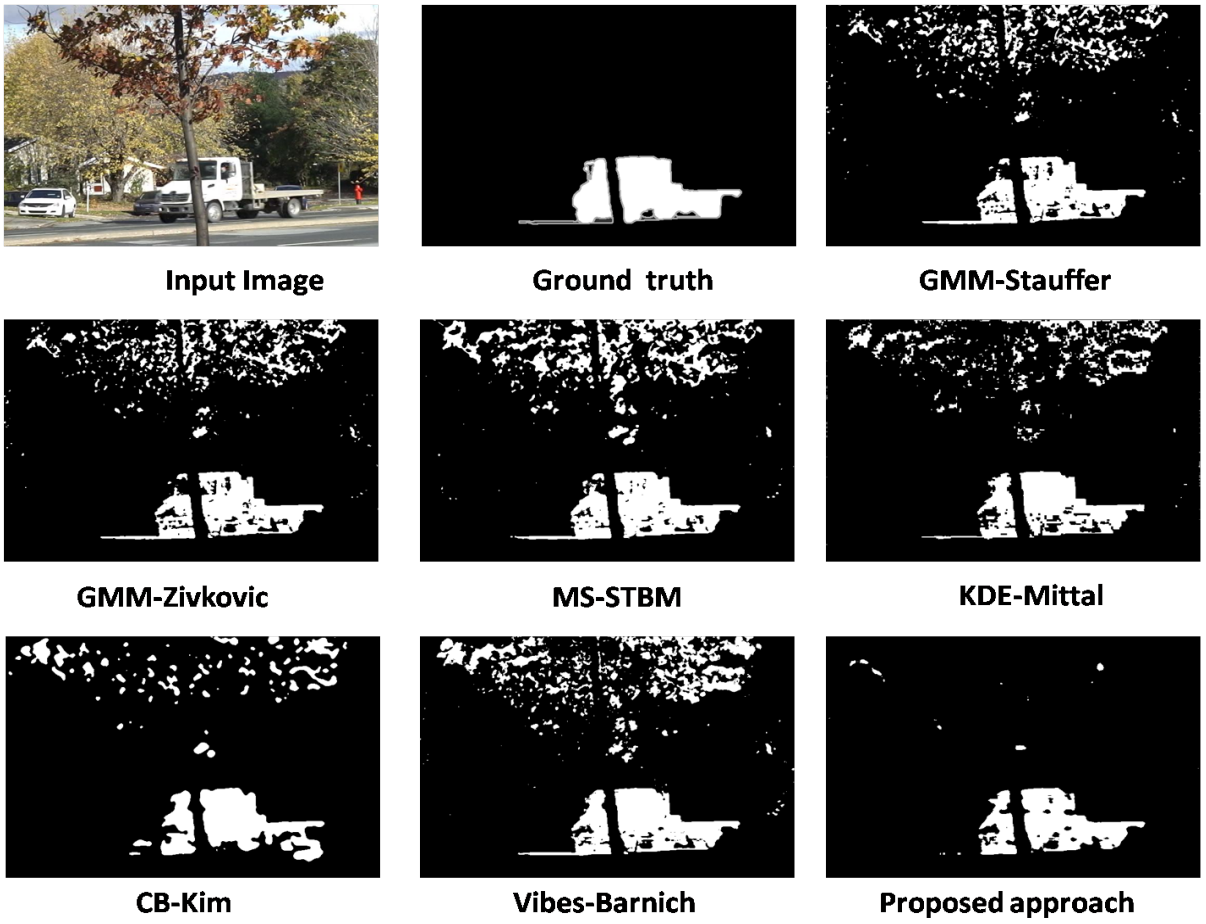


Figure 3.12: The qualitative results for frame number 1500 of *fall* video. Proposed approach is showing improvement by showing pixels belonging to tree leaves as background where other methods failed.

The sigmoid function is applied for assigning weights to each codeword for making decision about shifting codeword between different category of layers. Although linear function is appropriate for the scenario where objects move with constant velocity, but it fails in uncontrolled environment where dynamics of the background change continuously over a period of time. Therefore, we have applied sigmoid function to control learning rate of codewords that seems more effective in real world situation.

Also, Random neighbour selection policy is used in spatial context to avoid processing overhead in matching codewords using a 4-connected neighbourhood for deciding foreground pixels. Proposed method gives better results than standard codebook model and other state of the art methods by achieving high values of precision and recall. The proposed method achieves an overall better performance (as shown in Results section) when compared with other state-of-art methods.

The next chapter discusses a multiple object tracking, where we have described

our method of tracking multiple persons in challenging environment like long term occlusion and missed detection. The next chapter explains a coupling framework of position based and appearance based tracker for multiple object tracking.

Chapter 4

MULTI OBJECT TRACKING

Tracking may be defined as following the trajectory of a moving object across frames as it moves around the scene. Consistent labels are assigned to the tracked objects in each frame of a video. Further based on the tracking domain, a tracker can give useful information such as movement, shape, and orientation of the object under interest. Object tracking becomes a complex task due noise in images, complex object motion, articulated nature of non-rigid objects, objects occlude each other, objects occluded by a structure, and real-time processing requirements.

Tracking can be simplified by making some assumptions or imposing some constraints on the motion or appearance of the object. In almost all tracking algorithms, object motion is assumed to be smooth with no abrupt changes in between. Prior knowledge about the object size, number, appearance, shape and motion can also help in its tracking. A number of methods for object tracking have been proposed. In this thesis we focused on tracking moving objects in general and not on trackers tailored for specific objects, for example, human kinematics are used as the basis of parameters of tracker for implementation.

4.1 Object Tracking

Multi-object tracking is an inherent to many video based applications such as smart surveillance systems, augmented reality, crowd analysis and many more. Furthermore, trajectory analysis of target objects in videos is serving as a foundation tool for various other significant computer vision techniques used for knowledge discovery. Many real-time applications need tracking methods that give reli-

able result even in situations involving object with random movement, interaction between objects, scale variation, and occlusion. These parameters are the ones mainly responsible for variation in accuracy of different tracking methods.

The object of interest in a tracking application primarily depends on the requirement of analysis. For example, it may consider moving vehicles on the road for surveillance, face tracking for identification, motion structure of person for behavior analysis, etc. Proper feature selection in object detection and tracking is a crucial part that may influence the outcome of tracking significantly. By simply imposing motion velocity, direction, scale and structure of object the problem of feature selection can be overcome. Many recent advances in techniques allied with object detection and tracking consider motion structure, color cues and data association techniques to assign a consistent label to an object across the frames Chen et al. (2015); Milan et al. (2016).

With the recent advancements in computer vision, it can be claimed reasonably that there are feasible solutions available for addressing the robust tracking of the single target. However, simultaneous analysis of multiple targets in a video remains as one of the most challenging tasks in computer vision.

Although many tracking approaches operate on domain specific target representation that either determine manually or trained using the initial frame sequence Bao et al. (2013); Yang and Jia (2016), these methods tend to have challenges while tracking objects that show convincing variation in their appearance. It has been demonstrated that in many scenarios an adaptive appearance model, which evolves during the tracking process as the appearance of the object varies, is pivotal for achieving high performance. Another choice in the design of appearance models is whether to model only the object or both the object and the background. Numerous approaches have shown that application of discriminative classifier in training a model to separate the background from the target object generally results in superior performance Milan et al. (2015). As these procedures use object detection, they have been called "tracking-by-detection."

The tracking-by-detection approach use detector's output to associate current observations with existing trajectories. It is mainly categorized into the batch and online methods. Batch methods also termed as global optimization tracking, links fragmented tracks together by using detections of complete frame sequence. Here generally data association method is used to link short trajectories into a long trajectory Kamvar et al. (2004); Milan et al. (2016); Zhang et al. (2008). Although by analyzing complete frame sequences it may resolve some of the ambiguities

present due to detection failure and make tracking system robust, it also increases its computational complexity exponentially if there is a growth in the number of targets and that question the suitability of the system for real-time applications.

Online tracking methods sequentially connect detection in the current frame with existing trajectories or create new trajectory if they do not find a match with any of the previous trajectories. As they depends only on the current information, they are suitable for real-time applications. However, long-term occlusions and miss detections are two major issues with online object tracking methods that tend to generate fragmented trajectories Jacobs et al. (2007); Yang and Jia (2016).

Motivated by the above challenges we present an online object tracking method that can generate real-time solution even for complex scenarios like random motion and partial occlusion. A mechanism between data association and context based tracker which makes use of the color information as features is presented in this thesis. The pictorial representation of proposed approach is depicted in Figure 4.1.

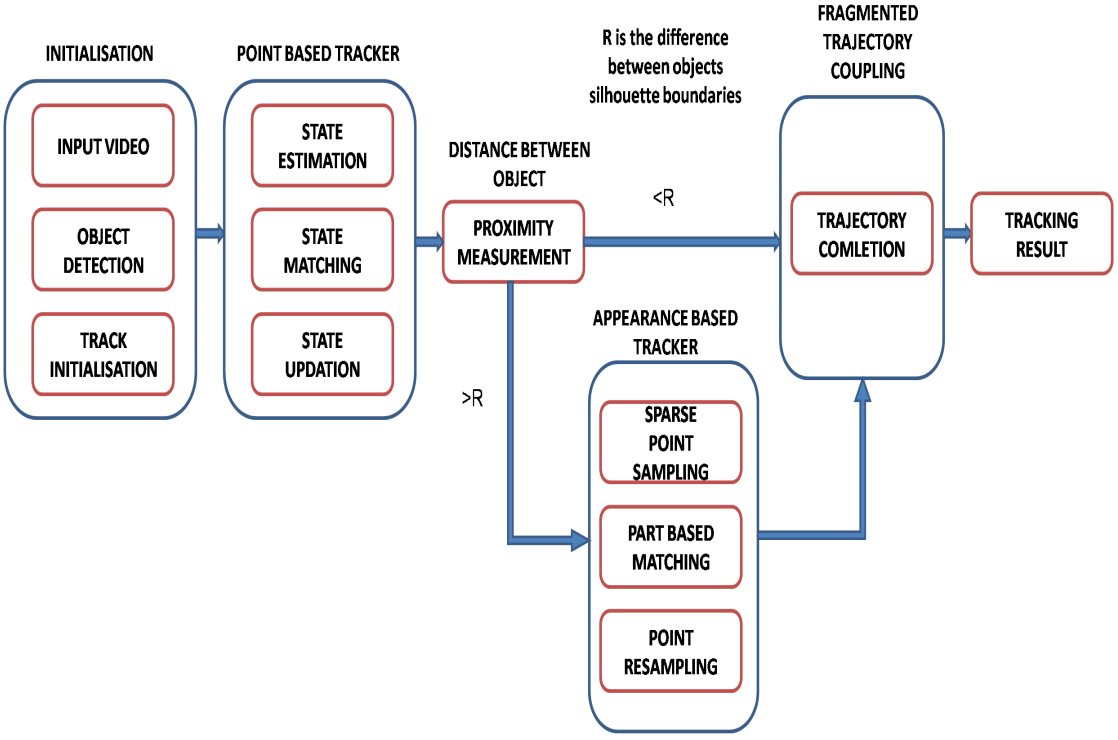


Figure 4.1: Schematic view of the proposed method for coupling point based and appearance based tracking. Moving objects are extracted and supplied to point based tracker. If the distance between object silhouette boundaries is $< R$ appearance based tracker is invoked. Trajectory coupling mechanism is used to join fragmented trajectories presented in closed proximity.

Principally, this figure shows a data association algorithm coupled with appearance based tracker with the ability to deal with a complicated scenario in multiple object tracking (e.g. associating one detection to many tracked objects (N-to-1), one tracked object to many detection's (1-to-N) that occur with the segmentation fault or partial occlusions).

The model proposed in this work has two strong points: first, low computational complexity accomplishes the requirement of designing a system that is suitable for real-time applications. Second, the structural representation of target is explicitly made adaptive based on their motion model and detector observations from the current image. By adopting background subtraction, we pull out the possibility of designing specialized tracker (for pedestrians only). Moreover, it does not require any training or particular learning process for classification. The process disseminated the coupling establishment to subproblems and compiled their local solutions to accomplish a global solution. In summary, our contributions are:

1. First, we formulate a data association task after track initialization; also it is intended to perform all associations with high probability.

2. Secondly, a efficient and local procedure is designed for coupling the subproblems of data association and appearance based tracking by formulation of proximity measurement between interacting objects.

3. Finally, a new sparsity-driven target specific proposal distribution that takes segmented foreground region as input to select features belonging to a target.

Our proposed coupling scheme is highly successful in eliminating error propagation due to miss detection and track overlap, which is a primary reason for decreasing the accuracy of traditional "tracking-by-detection" approaches.

As the proposed scheme is simple and does not demand high computational resources, it can be used for online multi-object tracking in real time applications. The primary objective of this work is to overcome "difficult-to-predict" nature of data association task. It is accomplished by finding the contiguous hypothesis using instance based target specific classifier. Furthermore, the essential aspect of proposed approach is taking advantages of background subtraction and data association sub problems.

The remaining sections of this chapter are organized as: Section 4.2 describes proposed approach for multiple moving object tracking and trajectory association.

Section 4.3 provides implementation details and performance evaluation of the proposed method finally summary of this chapter is given in Section 4.4.

4.2 Proposed Approach

Tracking multiple moving objects across frames is a highly challenging task when performed in imperative situations, as loss of object presence due to detector failure or occlusion between objects occurs frequently. Our goal is to track each object between a given frame sequence with complete or partial occlusion. The primary objective is to keep number of observations equal to the number of tracks present in a given video and also detection should be assigned to the track belonging to the corresponding object only.

4.2.1 Moving Object Detection

In multiple objects tracking approaches, the segmentation of area of interest is the primary task that needs to be performed in an initial stage. It separates target object from the background of a scene. The segmented regions are treated as target region for further tracking.

An adaptive background subtraction method Badal et al. (2015) has been applied to each frame to obtain the superpixel region of moving objects present in a video. Figure 4.2 shows the object detection result as a binary image after background subtraction. An input video as shown in Figure 4.2(a) is processed to extract frame sequence as depicted in Figure 4.2(b), then after applying background subtraction resulting a binary image represent the segmented moving object as shown in Figure 4.2(c).

The background subtraction may show false detection or miss detection due to long-term occlusion and interacting objects. With occlusion or miss detection, there is uncertainty in prediction that even non-maximal suppression technique require additional information to be given to the tracker to overcome these challenges.

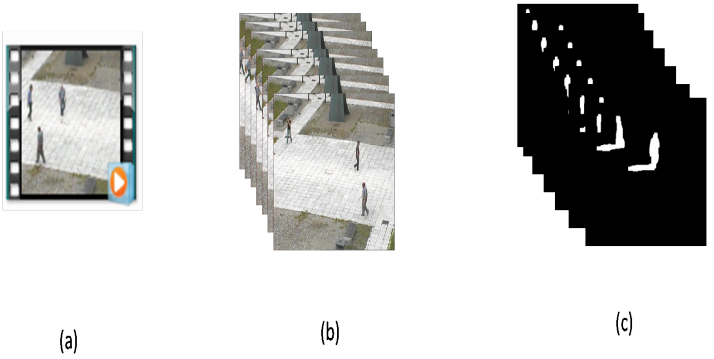


Figure 4.2: Result of moving object detection. (a) Input video. (b) Frame sequence. (c) Resulting binary representation of segmented moving objects.

4.2.2 Data Association

The tracking method applied here depends primarily over the distance between the objects. In the situation where object are distant apart a low cost point based technique is used. When objects move closer to each other the possibility of missing corresponding tracks increases, a more effective technique of tracking (i.e. appearance based tracker) is used. The proposed technique is adaptive and generalized in the manner that it is easy to substitute each sub problem used by other classic approaches. We adopt the Kalman filter Julier and Uhlmann (1997) tracking formulation to model the data association sub problem. Although many variants of this method are available in the literature, because of low computational cost the original form of this approach is utilized here. Again, the methodology is designed to keep the implementation simple and to be formulated as a sub problem, but there always exists a possibility to incorporate another optimization techniques. Proposed structure is based on binary image interpretation, and is not constrained by the specifics of the image resolution of the target objects.

A constant velocity model of Kalman filter is applied in estimating the current state of the corner points and the height of the bounding box of the object. The distance between the Kalman filter estimated current state $X_{kal} = (x_{kal}, y_{kal}, h_{kal}, w_{kal})$ and a rectangle position of object detection $X = (p_x, p_y, s_h, s_w)$ is defined as

$$d_{kal}(X_{kal}, X) = \sqrt{\frac{(x_{kal} - p_x)^2}{\sigma_x^2} + \frac{(y_{kal} - p_y)^2}{\sigma_y^2} + \frac{(h_{kal} - s_h)^2}{\sigma_h^2} + \frac{(w_{kal} - s_w)^2}{\sigma_w^2}} \tag{4.1}$$

where σ_x^2, σ_y^2, σ_h^2 and σ_w^2 are posterior error variance of x_{kal}, y_{kal}, h_{kal} and w_{kal}

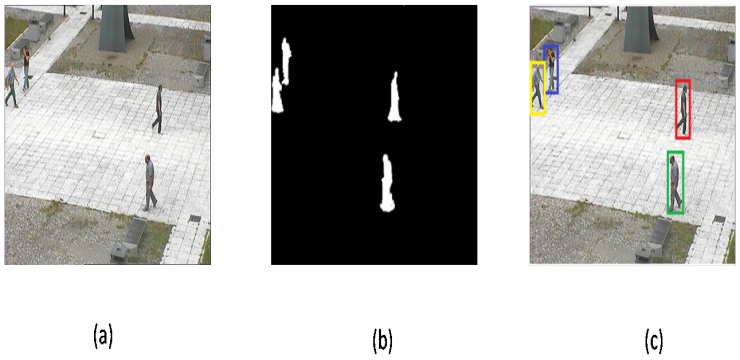


Figure 4.3: Result of point based data association tracker. (a) Input frame. (b) Mask image representing region belonging to the detected objects as foreground. (c) Tracked objects are assigned label as bounding box of different colors.

respectively.

Figure 4.3 (a)-(c) shows the result of point based data association applied for multiple object tracking. Where background subtraction is used to estimate the region belonging to the moving object in each frame of the video 4.3 (b). Track initialization is performed by assigning the track ID to each region detected in the earlier step. The Kalman tracker is used to estimate the position of consistent tracks in next frame as shown in 4.3 (c).

While tracking multiple targets the distinctiveness of the particular targets might be missed when targets move closer to each other. A state estimation tracker can be formulated for each observation in multiple objects tracking, only when all targets are adequately distant apart. When multiple targets move close to each other an ad-hoc appearance model based approach is applied to distinguish adjacent tracks. The appearance model is coupled with a state estimation model to estimate variation in position and to achieve smooth trajectory.

The spatial distance between object detection is examined to consider the indication of interaction between two objects. We formulate this as the distance between parameters used in data association step as given below in Equation 4.2.

$$I_t(i,j) = \begin{cases} 1 & if \ \ d_t(i,j) \geq R \\ 0 & else \end{cases}$$

where $d_t(i,j) = T_i^t - T_j^t$ is the spatial distance between current state of track i

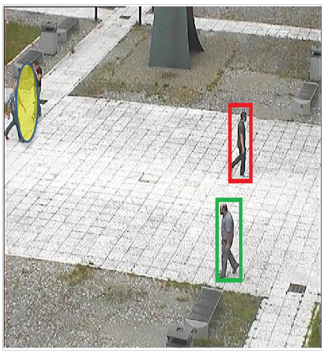


Figure 4.4: Object silhouette proximity measurement. Here the spatial distance R between object silhouette is examined to considered as an indication of interaction between two objects (i.e., $R < 25$).

and track j at time t and constant R as minimum distance used for computing the interaction. Here R is assigned a value 25 which means that there is an interaction between tracks if they are 25 pixels apart. The value of $R = 25$ is determined heuristically by experimenting on six videos of $2DMOT2015$ dataset Leal-Taixé et al. (2015b), that is also used for system performance analysis. This distance is used for target initialization for the appearance model in the presence of continuous overlapping of detected object positions marked as a yellow oval shape in Figure 4.4.

This decision characteristic of our coupling procedure is computationally efficient and is straight forward for further extension to higher order applications. It helps in avoiding extraneous evaluation of the appearance model at improbable positions by keeping the size of search window around estimated position only. Also, it benefits us in a selection of correct target with resembling appearance.

4.2.3 Appearance Model

As occlusion cannot be resolved entirely by object detection and data association techniques collectively, a natural expansion is to consider an ad-hoc representation of the occluded object into a sampling procedure by taking advantage of the multi-modality in tracking. In this situation, the sparse feature based tracker Gordon et al. (2004); Czyz et al. (2005) has been extensively practiced and has vital importance in occlusion handling and pose variance in multiple targets tracking. An online multi-instance based tracking approach is formulated that can be used in the case of failure in primary tracker. The coupling mechanism is also significant for multi-modal scenario as it applies to different modes of situations like: sparsely

occupied scene, scenario having interacting objects and scene with many objects. Only point based tracker is applicable for sparse situation as it integrates with an appearance based technique when object starts interacting.

This tracker is triggered when there is no match found for an existing trajectory and which also satisfy the boundary conditions. The color features are used to represent an object appearance and to match likelihood of observations. This structural information helps to resolve ambiguities whenever they are missed by motion model to establish correspondence with any detection.

The objective is to linearly perform simultaneous state estimation in a frame sequence $Z_k = \{z_1, z_2, \cdots, z_k\}$. Where z_k denotes the frame at discrete time instant k, so that it can be associated with the track that possesses similar color feature values $F = \{f_1, ..., f_N\}$ in corresponding channels c_i with reference to target appearance model. The features $f_i = (x_i, y_i, w_i, h_i, c_i)$ evaluate the sum of the feature values for each channel inside the rectangle at position (x_i, y_i) with width w_i, height h_i. For each feature f_i the expected feature $f_i(X)$ of the observed object X (label $Y = 1$) values are modeled using a Normal distribution.

Under weak assumptions and using the strong law of large numbers, it is possible to show that at each iteration the estimation of the mean given by the particle filter is asymptotically unbiased. Furthermore, if the variance of this estimator is finite, the central limit theorem justifies an asymptotic normal approximation for it. In case of particle distribution without prior information about target region sampling operation may generate a proposal that can vary the result falsely.

Okuma et al. (2004) has supposed non-Gaussian target distribution under the consideration of inclusion of biased distribution in case of pose variation of the target. While selecting the particle over target space, we have considered the information from background subtraction phase. So the distribution of particle over the pixel set in mask image sequence reduced the possibility of biased selection of particle. This is the main reason behind considering the Normal distribution in target selection.

We adopt the Gaussian density for the likelihood function of the measured color histogram as follows:

$$p\left(f_i\left(X\right) \mid Y = 1\right) \propto N(D_k; 0, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{D_k^2}{2\sigma^2}} \tag{4.3}$$

where D_k is the distance between the histogram computed from the current ob-

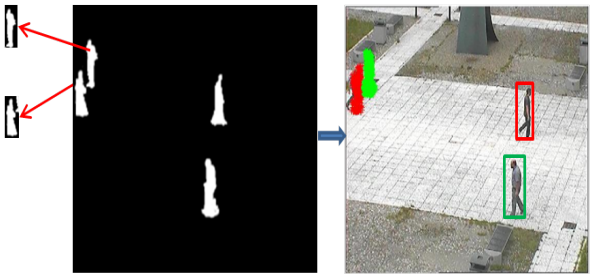
servation X_k and reference histogram X_k^* of objects to be tracked. If the two histograms are calculated over U bins, The Bhattacharya coefficient Okuma et al. (2004) is used as likelihood measure to find the relative similarity D_k between reference object template and current observation points as:

$$D_k = \sqrt{1 - \sum_{u=0}^{U} \sqrt{X_{k,u}^* X_{k,u}}} \tag{4.4}$$

where $\sum_{u=0}^{U} \sqrt{X_{k,u}^* X_{k,u}}$ geometrically represents the cosine similarity between the m-dimensional unit vectors $X_{k,1}^*, X_{k,2}^* ... X_{k,U}^*$ and $X_{k,1}, X_{k,2} ... X_{k,U}$. The Bhattacharya coefficient is more efficient then Mahalanobis distance, and it is suitable for the situation where two classes show same mean but different standard deviation.

The performance of the appearance based tracker depends mainly on the technique used for generating the proposal distribution. The state evolution model $p(X_k|X_{k-1})$ is considered here for assigning weights to corresponding features according to the likelihood measure. The sparse criteria are maintained by selecting hypotheses constrained over foreground region generated by the detector. It significantly reduces the number of false assumptions chosen due to the absence of non-maxima suppression technique. The process may lead to select hypotheses shared by neighboring objects because of occlusion. However, it can be suppressed by assigning weight to correct assumptions. The Figure 4.5 (a) and Figure 4.5 (b) shows the mask image and feature distribution respectively after point estimation based on pixel filter using the result of background subtraction. The region belonging to an object present in proximity is segmented out using mask image generated by background subtraction step as shown in Figure 4.5 (a). This segmented region is used in feature distribution for applying appearance based mapping. The particle distribution for two objects are represented in Figure 4.5 (b) using red and green color respectively.

The state space approach Okuma et al. (2004) fundamentally depends upon a state estimation strategy $p(X_k|X_{k-1})$ and a motion model to measure the movement of the target in a spatial domain, i.e., the association between state and recent measurement $p(z_k|X_k)$. While tracking the silhouette belonging to target object, initial state $p(X_0)$ is assigned some random distribution feature values present in the target region. Afterwards, the Bayesian state estimation strategy is formulated in calculating filtering distribution $p(X_k|z_k)$ for next frame. In Bayesian state



(a) (b)

Figure 4.5: Pictorial representation of feature distribution for objects in closed proximity. (a) Mask image used for distributing features in restricted region. (b) Feature distribution respectively after point estimation.

estimation, two steps are followed in computing the filtering distribution. The first step is prediction step where filtering distribution is calculated through the Bayes' rule as given in Equation 4.5.

$$p(X_k|Z_{k-1}) = \int p(X_k|X_{k-1})p(z_k|X_k)dX \tag{4.5}$$

This prediction step depends upon the specification of a dynamic model characterizing the state evolution, $p(X_k|X_{k-1})$ and a motion model that gives the probable occurrence of any state with the current observation, $p(z_k|X_k)$. In the second step, the objective is to estimate posterior distribution in the presence of prior and current distribution as given in Equation 4.6.

$$p(X_k|Z_k) \propto p(X_k|X_{k-1})p(z_k|X_k) \tag{4.6}$$

Once the combination of filtering distribution is generated, point estimates for proposal distribution can be decided after applying any appropriate estimation criteria, most widely used functions are Maximum a Posteriori (MAP) estimate, $argmax_{X_k}p(X_k|Z_k)$, and the Minimum Mean Square Error ($MMSE$) estimate, $\int X_k p(X_k|Z_k)$.

Initially a weighted set of samples $\{X_{k-1}^{(i)}, w_{k-1}^{(i)}\}_{i=1}^{N}$ approximately distributed according to $p(X_{k-1}|Z_{k-1})$, new samples are generated from a suitable proposal distribution, which may depend on the previous state and the new measurements, i.e., $X_k^{(i)} \sim q_p(X_k|X_{k-1}^{(i)}, z_k), i = 1, \cdots, N$. To maintain a consistent sample, the

new importance weights are set to:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(z_k|X_k^{(i)})p(X_k^{(i)}|X_{k-1}^{(i)})}{q_p(X_k|X_{k-1}^{(i)}, z_k)} \tag{4.7}$$

with $\sum_{i=1}^{N} w_k^{(i)} = 1$. The new feature set $\{X_k^{(i)}, w_k^{(i)}\}_{i=1}^{N}$ is then approximately distributed according to $p(X_k|Z_k)$. At time instant k a subset $F_i \subset F$ of selected features are used to build the model:

$$p\left(F_k\left(X\right) \mid Y = 1\right) = \prod_{f_i \in F_k} p\left(f_i\left(X\right) \mid Y = 1\right) \tag{4.8}$$

Assuming that $p_k(Y = 0) = p_k(Y = 1)$ the log odds ratio is used for inference:

$$H_k(i) = log\left[\frac{p_k\left(Y = 1 \mid F_k\left(X\right)\right)}{p_k\left(Y = 0 \mid F_k\left(X\right)\right)}\right] \tag{4.9}$$

The final appearance distance $d_{app}(x)$ of the appearance model given a query X is normalized to $[0, 1]$ by

$$d_{app}(X) = 1 - 0.5\frac{H_k(X). \mid F_k \mid^{-1}}{1 + \mid H_k(X). \mid F_k \mid^{-1}\mid} \tag{4.10}$$

with $\mid F_k \mid$ is the number of features in F_k.

To obtain a compact model for faster evaluation, discriminative samples $F_k \subset F$ are selected while updating the model at time k. The online Multiple-Instance Learning Avlonitis and Chorianopoulos (2014) and Okuma et al. (2004) are used for object classification. Multiple-Instance Learning uses bags of samples $X_i = \{x_{i1}, ..., x_{in}\}$. The bag is labeled by $Y_i = max_j(Y_{ij})$, so that at least one positive sample is sufficient for the bag to assign it a positive label. This instance based labeling is applicable in handling occlusion where only some portion of the object is visible. While updating, a positive sample is added to a bag by selecting target instance near estimated position of a target to maximize log likelihood in the posterior state. Only the positive samples are collected to maximize the log likelihood of all instances:

$$L = \sum_i log\left(p(Y_i \mid X_i)\right) \tag{4.11}$$

with $p(Y_i = 1 \mid X_i) = max_j(p(Y_i = 1 \mid x_{ij}))$ and $p(Y_i = 1 \mid x_{ij}) = 1 - d_{app}(x_{ij})$. Hence for a positive bag one sample with high likelihood is enough to get an

overall high bag likelihood, but all samples need to have a low likelihood for a low bag likelihood. Figure 4.6 shows multiple instance based feature distribution. Features distribution is performed for each object in proximity and after that matching of multiple instances is perform by comparing histogram of each instance independently.

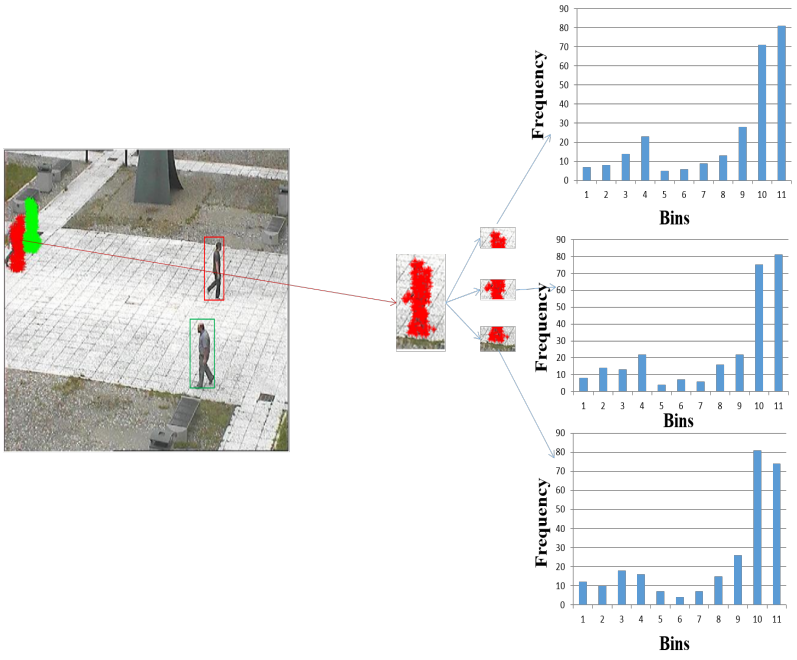


Figure 4.6: Pictorial representation of multiple instance based feature distribution. Objects in closed proximity are tracked using appearance based tracker. Multiple instances of each object are matched with the template of the object instance by finding the difference between their color component.

Our method selects M features in a greedy fashion:

$$f_m = argmax_{f_i \in F} L(H_{m-1} + h_i) \tag{4.12}$$

with

$$h_i = log\left[\frac{p_k\left(Y = 1 \mid F_i(X)\right)}{p_k\left(Y = 0 \mid F_i(X)\right)}\right] \tag{4.13}$$

After each step the selected features are added to the model:

$$F_{k,m} = F_{k,m-1} \cup \{f_m\} \tag{4.14}$$

The performance of the particle filter depends on the quality of the proposal distribution. In this work, state evolution model $p(X_k|X_{k-1})$ as a proposal distribution has been formulated which makes the new importance weights proportional to the

corresponding particle likelihoods. and H_m is updated for each sample as

$$H_m(X) = H_{m-1}(X) + h_m(X) \tag{4.15}$$

The final set of selected features is given by $F_k = F_{k,m}$.

4.2.4 Proposed Tracking Procedure

The primary objective of the multi-object tracking step is, to combine detection response d from given set of current detection D_t with the track b from existing tracks T, where distance measure between them is minimum according to a given set of parameters. At time instance t each track $b \in T_t$ is represented by assigning a set of six tuples as $X_t(b) = (p_x, p_y, s_w, s_h, K_t(b), F_t(b))$. Where p_x and p_y represent position vector on x-axis and y-axis respectively, s_w and s_h are used to denote size of an object. An object appearance template and Kalman filter is represented by $K_t(b)$ and $F_t(b)$ respectively. For associating a detection $(d \in D_t)$ with track $(b \in T)$, first Kalman filter $K_t(b)$ is used to estimate position $(p_x(b), p_y(b))$ and size $(s_w(b), s_h(b))$ of each track at time t. The Kalman filter may estimate erroneous position which are refined by searching their positions using appearance model. The feature distribution for generating the appearance model $F_t(b)$ is performed by creating the search window r around the position predicted by Kalman filter as given in Equation 4.16.

$$\left(\frac{p_x - p_x(b)}{s_w}\right)^2 + \left(\frac{p_y - p_y(b)}{s_h}\right)^2 < r^2 \tag{4.16}$$

Further, position with highest matching score with appearance model is found as:

$$X_t(b) = argmin_{X \in N_r(K_t(b))} d_{app}(X) \tag{4.17}$$

$N_r(X)$ denotes the neighborhood of X within the proximity area measure by r, with

$$(p_x, p_y, s_w, s_h) \in N_r(X(b)) \Leftrightarrow s_w = s_w(b), s_h = s_h(b) \tag{4.18}$$

and finally, a detection $d \in D_t$ is associated with a track $b \in T_t$ by using the following minimum distance function:

$$dist(b, d) = argmin\left(d_{app}(b, d) + d_{kal}(b, d)\right) \tag{4.19}$$

Our method greedily associates detections to tracks by minimizing $dist(b, d)$.

4.2.5 Trajectory Completion

After track formation, a system may generate fragmented tracks belonging to the same trajectory due to miss detection and occlusions. Hence, the tracklets can be merged by imposing linear motion model by applying velocity continuity constraint over the gap. The trajectory interpolation in this work is based on the following two constraints:

1. The distance between the terminating position of one trajectory with the initial position of other is less than R. Where R is assigned a value 25 in our implementation. Here the primary consideration is that the persistent track should be initiated and terminated near the boundary of the region only. If a track is started in the area that is away from the boundary region, it is considered to be a segmented part of a previous track. If the terminating position of one track is less than 25 pixels than the starting position of other track, they are considered to be part of the same track.

2. The corresponding affinity scores between two trajectories should be higher than a threshold.

4.3 Experiments

In Section 4.2 we introduced a coupling strategy that has been formulated with the primary objective of precisely reflecting the continuous action of multiple objects in proximity. Our claim is that by distributing the coupling formulation to sub-problems and by integrating their local outcomes to accomplish a comprehensive, result will achieve higher tracking efficiency. To establish this claim analytically, we apply our approach on the distinct databases and evaluate those experimental outcomes. In the following subsections, we analyze the influence of the particular sub problem in coupling procedure, examine tracking performance while varying parameter values to achieve efficient outcomes. Next, we state different matrices used for performance evaluation and discuss results of proposed method by showing a quantitative analysis of the various database.

4.3.1 Implementation

This section discusses the experimental setup and implementation details of the work proposed.

Tracking area: to implement valid track initialization and termination during entry or exit of objects, constraint over boundary conditions is applied. It is necessary to enforce boundary around tracking area for generating persistent trajectories and analysis of tracking result. A rectangular region is marked on the frame to simplify the distance estimation of an object from the boundary. An object outside of this rectangular area is treated as extraneous and is not considered for further processing.

Run time: For comparing the execution time of tracking only, we exclude the processing time spent in detection. Present implementation processes approximately 20 frames/second with precise occlusion handling. If computationally expensive occlusion reasoning is left out the program runs faster and accomplishes real time execution.

Parameters: The initial proposal distribution of particle features $p(X_k^{(i)})$ was uniformly selected around positions estimated by Kalman filter as explained in Section 4.2 (Combining the prior information to choose the most likely position where an object may appear). Finally, the color and gradient features were computed for each color channel independently over $8 \times 8 \times 8$ bins. The target template is generated using training frames calculated by proximity formula in Equation 5.16. The uniform search space for proposal distribution and histogram computation around detected targets is computed using Equation 4.16. The multiple instances of target histogram is generated for a frame to overcome partial occlusion.

The number of features required by an appearance model is determined by the number of objects with the similar appearance in a proximity and the prior information on the probable position of objects in the current frame. For distant or divergent objects up to $N = 200$ features are sufficient to achieve acceptable performance for detection to track assignment. If two or more identical objects are present in proximity, it becomes necessary to increase feature size as $N = 500$ in order to accomplish adequate accuracy and to generate smooth tracks of interacting objects. By applying prior estimation of position and suitable feature distribution strategy the number of features required can certainly be reduced acutely. We have applied some constraints over image boundary to deal with entry and exit conditions like: (a) Each frame is marked with a boundary region

which is considered to start processing of tracking an object. It is required to identify when an object enters the scene; it is necessary for ensuring the complete appearance structure of the object. (b) Every scene has entry and exit from the boundary of the frame only.

4.3.2 Parameter study

To avoid the risk of over-fitting we have learned parameters in proposed algorithm from sample videos, it becomes possible due to the availability of annotated ground truth videos. Although these videos show considerable variation in target motion structure and their appearances, it is required to use only efficient values for each parameter and apply this value set for all test videos. To determine the outcome of the distinct parameters of each sub problem in Section 4.2, we run our tracking algorithm and adjust the respective parameter while keeping all the other ones fixed. In Figure 4.7(a)-(c), it shows the plot of the relative change in performance for each term against the parameter value.

The strongest deterioration can be noticed when N the number of particles for appearance model is set too low. This once again established our confidence on explicitly modeling the localized proposal distribution to avoid situations of overlapping targets.

Although, assigning a high value to the number of bins result in high accuracy value, but it also increases computational complexity exponentially with increase in the number of targets. The greediest strategy always chooses the best possible combination of speed and particles. To maintain processing speed in real time, we select the number of particles to 500 as given in Figure 4.7(a). The system shows the highest accuracy when assigning 8 bins for each color space as shown in Figure 4.7(b). Proximity threshold value is used to couple data association based tracker with appearance based tracker. While a higher value of the threshold for proximity will yield high accuracy rate, it will also increase learning time for proposal distribution. The optimal choice for distance parameter is 25 as shown in Figure 4.7(c). Moreover, regardless of change in scenario, tracking performance is not affected much by varying parameter value over the particular range and results remain stable for the determined respective combinations.

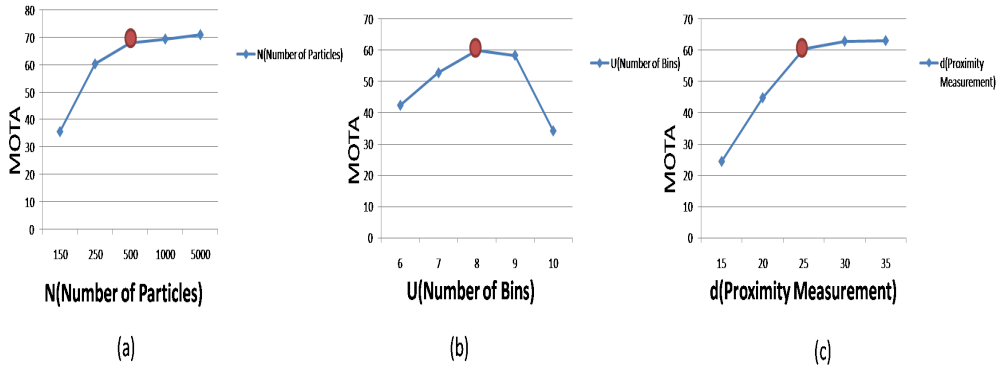


Figure 4.7: Influence of individual parameters on tracking performance. Each plot shows the relative change in performance (measured by MOTA) by changing the value of a single parameter while keeping the other ones fixed. The parameter value used in our experiments is marked with a circle.

4.3.3 Datasets

The performance of proposed approach is evaluated and discussed on six video sequences of publicly available Benchmark data sets MOTChallenge Leal-Taixé et al. (2015b). As these videos cover various challenges like variability in a number of objects, long-term occlusion, and dynamic motion behavior, etc. While considering multiple object tracking techniques, this dataset is best to perform a quantitative evaluation. These videos are suitable to be considered for surveillance scenario as the recording has been done from the elevated perspective. Note that there is a substantial disparity in targets appearance due to illumination variation and shadow.

4.3.4 Metrics

Due to different evaluation metrics and dataset used in tracking it is not easy to quantitatively evaluate different multi-target tracking methods. Comparative analysis of different tracking approaches using a single objective function is not feasible because usually tracking methods are application specific. Furthermore, assigning a stable value to various parameters is also a primary issue in evaluating tracker performance over precision and correctness. The most widely followed metrics for multiple objects tracking CLEAR MOT Leal-Taixé et al. (2015b) is used in this research work for quantitative evaluation. It computes the distance between targets to the manually annotated ground truth on the ground plane and the hit/miss threshold is assigning a value as 1m. The Multi-Object Tracking

Accuracy (MOTA) incorporates three categories of failures while tracking: false positives (FP), missed targets (FN), and identity switches (ID). All categories are assigned equal weights, and then normalization is performing such that the score of 100% resembles no errors. Parameters used for evaluation as shown in Table 4.1 are: Multi-object tracking accuracy (MOTA) and multi-object tracking precision (MOTP).

The MOTA evaluate tracker performance by combining three types of errors (explained in Table 4.1) as given below in Equation 4.20:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{4.20}$$

where t is the frame index and GT is the number of ground truth objects. The MOTP is used to compute bounding box overlap between true positives and their corresponding ground truth targets. It measures the average dissimilarity between correctly matched hypothesis and their respective targets as given below in Equation 4.21:

$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{4.21}$$

where c_t denotes the number of matches in frame t and $d_{t,i}$ is the bounding box overlap of target i with its assigned ground truth object. It basically returns the localization precision value of detector. For providing better understanding and comparison of results over different error categories individual values along with number of fragmented trajectories (Frag) according to Ellis and Ferryman (2010) are given in Table 4.1.

Finally, four more metrics are mostly tracked (MT), mostly lost (ML), partially tracked (ID) and how many times each track is fragmented (FM) are also measured. Evaluation script is available at Leal-Taixé et al. (2015b) so we can evaluate our result over the same parameters. The proposed tracker is compared with baseline method and other particle-based methods for which result is available on the same dataset.

4.3.5 Quantitative evaluation

- *GMPHD* Song and Jeon (2016), a method with a hierarchically adopted filter using motion and appearance.

Table 4.1: Parameters used for evaluation.

Parameter	Preferred Value	Ideal Value	Description
MOTA	higher	100%	Multiple Object Tracking Accuracy.
MOTP	higher	100%	Multiple Object Tracking Precision.
FAF	lower	0	The average number of false alarms per frame.
MT	higher	100%	Mostly tracked targets.
ML	lower	0%	Mostly lost targets.
FP	lower	0	The total number of false positives.
FN	lower	0	The total number of false negatives.
ID Sw.	lower	0	The total number of identity switches.
Frag	lower	0	The total number of fragmented trajectory .

- *MotiCon* Leal-Taixé et al. (2014), it generates interaction based feature string based on motions of targets.
- LP_SSVM Wang and Fowlkes (2016), learns parameter using target-specific loss function in predicting complete set of model parameter.
- $RMOT$ Yoon et al. (2015), considers relative movement between objects to define the spatial relationship between objects.

Table 4.2 shows the quantitative analysis of tracking results which include our method and other four methods. In data association based $GMPHD$ tracker; it generates fragmented trajectories in the case of occlusions. We also compared our approach to LP_SSVM, it uses network flow model in association with contextual information to estimate pairwise cost between tracks. The main issue with LP_SSVM approach is that if multiple objects are present in close proximity, it increases the number of ID switches and losses track. The relative motion based tracker $RMOT$ is distinct to our approach in that it estimates position by applying data association only, which may lead to an increase of false positive in the absence of object appearance cue. However, our method takes advantage of knowledge from background subtraction in generating proposal distribution from region belonging only to the target. By applying the target specific appearance cue, proposed method is able to track most of the targets throughout their presence. Proposed method outperformed the other trackers in both accuracy and efficiency which is higher.

The average experimental results over all benchmark video sequences in comparison with other appearance based methods are reported in Table 4.2.

By analyzing results in Table 4.2 the following observations can be drawn:

Table 4.2: Performance analysis of the proposed method with other appearance based tracker on sequence six publicly available benchmark video sequences.

Dataset	Method	MOTA	MOTP	MT	ML	FP	FN	ID Sw	Frag
TUD-Crossing	*GMPHD*	50.5	72.3	15.40%	15.40%	41	485	19	29
	MotiCon	58.2	70.8	23.10%	15.40%	32	403	26	32
	LP_SSVM	60	74.2	30.80%	15.40%	48	375	18	20
	RMOT	62.8	73	30.80%	15.40%	34	**362**	14	**19**
	Proposed	**63.1**	**74.2**	**38.50%**	**15.40%**	**20**	370	**4**	26
PETS09-S2L2	*GMPHD*	31.9	69.1	0.00%	31.00%	**467**	5,965	**131**	315
	MotiCon	**46.6**	67.6	9.50%	14.30%	560	4,354	238	264
	LP_SSVM	41.5	**70.5**	7.10%	16.70%	629	4,803	212	**249**
	RMOT	37.2	67.7	9.50%	14.30%	1,126	4,743	190	320
	Proposed	42.7	69.4	**9.10%**	**7.50%**	834	**4,205**	160	417
AVG-TownCentre	*GMPHD*	**16.3**	70	4.00%	65.00%	413	5,542	**26**	103
	MotiCon	11.9	**70.3**	0.90%	69.90%	**353**	5,872	74	**75**
	LP_SSVM	14.7	70.1	2.70%	61.50%	459	5,515	123	141
	RMOT	5.5	66.9	0.90%	59.70%	1,260	5,424	74	171
	Proposed	16.1	66.3	**4.70%**	**40.90%**	1,379	**4,197**	93	407
ADL-Rundle-3	*GMPHD*	15.3	**73.2**	0.00%	40.90%	1,374	7,174	64	108
	MotiCon	18.1	71.8	4.50%	**20.50%**	2,755	**5,355**	217	140
	LP_SSVM	28	72.9	**9.10%**	22.70%	1,855	5,388	81	**83**
	RMOT	20.6	71.6	**9.10%**	22.70%	2,574	5,388	112	115
	Proposed	**31.1**	72.1	**9.10%**	27.30%	**1,215**	5,478	**63**	150
KITTI-16	*GMPHD*	27.5	73.2	0.00%	29.40%	154	1,061	**19**	52
	MotiCon	38.8	70.1	0.00%	**11.80%**	142	863	36	48
	LP_SSVM	**39.2**	**73.6**	0.00%	**11.80%**	**90**	924	20	**29**
	RMOT	37.6	70.8	0.00%	17.60%	182	858	21	51
	Proposed	36	71.4	0.00%	17.80%	256	**845**	29	64
Venice-1	*GMPHD*	13.2	71.5	0.00%	41.20%	728	3,204	29	70
	MotiCon	18.2	72.9	0.00%	**29.40%**	820	2,838	74	72
	LP_SSVM	17.8	**73**	0.00%	41.20%	696	3,032	**23**	**27**
	RMOT	**18.8**	71.2	**11.80%**	35.30%	893	**2,781**	33	74
	Proposed	17.3	71.6	5.70%	51.30%	**532**	2,976	28	121

The six video sequences are used for testing. While considering accuracy ($MOTA$) parameter different methods are achieving the best result in different scenarios. Proposed method gives the best accuracy in two videos and also in others sequences the average difference between proposed method and best one is 2.2% only. The reason behind low value in $PETS09-S2L2$ and $KITTI-16$ sequence is overlap region generated by background subtraction region, it results in the distribution of particle in between the different objects. While considering overall tracking accuracy ($MOTA$) as reported in Table 4.3 the proposed method outperformed the other methods.

The number of the most tracks tracked is also significantly high in comparison to others; also the desirable level value of mostly lost (ML) tracks is achieved by our method. The main reason behind the high value of mostly track (MT) and low most lost (ML) value is the coupling of fragmented trajectories by incorporation of

Table 4.3: The average experimental results over all six video sequences.

Method	MOTA	MOTP	MT	ML	FP	FN	ID Sw	Frag
GMPHD	25.78	71.55	3.23%	37.15%	**530**	3905	**48**	113
MotiCon	31.96	70.58	6.33%	26.88%	777	3281	111	106
LP_SSVM	33.53	**72.38**	8.28%	28.22%	630	3340	80	**92**
RMOT	30.41	70.2	10.35%	27.50%	1012	3260	74	125
Proposed	**34.38**	70.83	**11.18%**	**26.70%**	706	**3012**	62	128

low-level appearance cue. In $Venice-1$ where most of the methods fail in tracking the target for complete frame sequence of its presence in video, a 5.70% of MT is achieved by proposed approach. In $KITTI-16$ the frame sequence contains pedestrians crossing a street captured from a car. The lights of the scene are low, and the object can not be segmented from background clearly when captured through the car. Due to the small size of objects across the frame sequence, no target is tracked across the complete frame sequence of its presence in the video. Due to the reasons mentioned above, no approach in the analysis can track objects completely. It is shown as 0.00% under MT column.

Proposed target Specific instance based strategy distinctly surpass the other trackers particularly in an environment with random movement between targets. By modeling target appearance it keeps track targets distinctively. The proposed coupling scheme performs better than the other online tracker concerning accuracy. Through various experiments over different datasets, our tracker is comparably efficient than other state-of-the-art methods based on evaluation parameters used for comparison. Qualitative analysis is demonstrated using ten frames in Figure 4.8 and Figure 4.9 with partially occluded dense crowd. It shows performance of proposed tracking approach over successive frame sequences with successful tracking of multiple moving objects.

4.4 Summary

In this chapter, we presented a coupling procedure between data association and target specific appearance based tracker for multi-target tracking, which included explicit occlusion reasoning and appearance modeling. It combines the outcomes of object detection with data association discriminative tracking for estimation of promising particles, taking advantage of the often complementary nature of the two subproblems. Prior estimation of the particle through detection result introduces natural sparse behavior in selection. We use particle filter-based tracker with color

Figure 4.8: Qualitative analysis of tracking using proposed method over $TUD - Crossing$ video of $PETS2009$ dataset from frame 89 to frame 98. Occluded objects are tracked successfully using proposed approach.

and gradient as a feature to map posterior state to calculate the distance between reference particle color histogram with particles belonging to current observation.

In this chapter we further demonstrate that, our coupling based tracking approach enable us to model the motion structure of targets that helps in driving heuristics for the likelihood of appearance using non-maxima suppression. Moreover, incorporating detection result in generating proposal distribution also reduces the chance of selecting particles from background region and distributing them around target region only.

The proposed coupling procedure is characterized by a small number of false detection, significant reduction in identity switches and join segmented trajectories. Furthermore, in a less crowded environment when targets are well separated only data association can achieve the competitive result that leads to low computational complexity.

As reported in Section 4.3.5 quantitative and qualitative results on benchmark video sequences shows that proposed coupling model surpass state-of-the-art method used for multiple moving object tracking. These promising results motivate and enable us to perform important activity classification from surveillance video, which is detailed in next Chapter.

Figure 4.9: Qualitative analysis of tracking using proposed method over $PETS09 - S2L2$ video from frame 86 to frame 95. Proposed method has shown significant tracking result with the objects showing random motion structure.

Chapter 5

VIDEO SYNOPSIS AND INDEXING

One of the primary driving applications of motion analysis has been an automated analysis of surveillance video which has been partly motivated by the focus on security and prevention of terrorist attacks increased in recent years. In literature, researchers have proposed various techniques to generate a compact representation of video data that require respectively less time to monitor and also take less storage space. While we study video analysis, we divide methods developed in the literature broadly into two categories: Static image based summarization to generate a sketch of all activities in original videos and dynamic content based video summarization.

In static image based methods, each shot represented by key frames is selected to generate a representative image. Some of the examples of static image based summarization are video mosaic in which video frames are found using region of interest and stitched together to form a resulting video. Another form is video collage in which single image is generated by arranging region of interest on a given canvas. Storyboards and narratives are some more basic form of image based summarization. However, static image based methods generate summary in less space but here it does not take care of temporal dependencies between important events and researchers also want to maintain the resulting summary visually more appealing than watching static images.

As an example of dynamic content based video summarization method, Video synopsis condenses video content in both spatial and temporal dimensions and presents a short video that helps in fast browsing. Figure 5.1 illustrated the

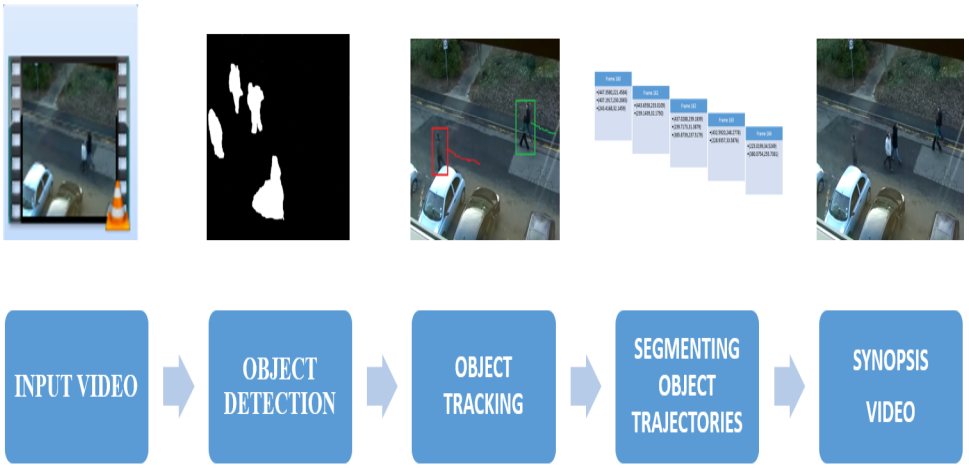


Figure 5.1: An overview of approach for generating synopsis video. Motion is extracted and tracked across the frame sequence and information related to tracked object is stored in the form of an array. Synopsis video is generated by arranging segmented trajectories over temporal and spatial domain.

general structure for generating synopsis video.

Video synopsis presents some limitations as it requires large memory area to store foreground and background regions. While video synopsis save space it does not maintain consistency between different objects, also the pleasing effect of the video is highly dependent upon the length of the final synopsis. Some more examples of dynamic methods are video fast-forward, video skimming, spacetime video montage, video narrative etc., where selected frames are arranged to form a highly condensed video.

Although the existing approaches of video synopsis work well in condensing activities present in video over space, they do not preserve interaction between objects. While going through the various surveillance videos, it is observed that the object interaction in video possesses vital information such as information exchange, accidents, and theft.

We present an approach to produce video synopsis while preserving motion structure and object interactions. While condensing video, object appearance over the spatial domain is maintained by considering its weight that preserves important activity portion and condenses data related to regular events. The spatiotemporal tubes form a distinction between objects by separating their motion structure that helps in producing semantic information about the distinct objects. The semantic

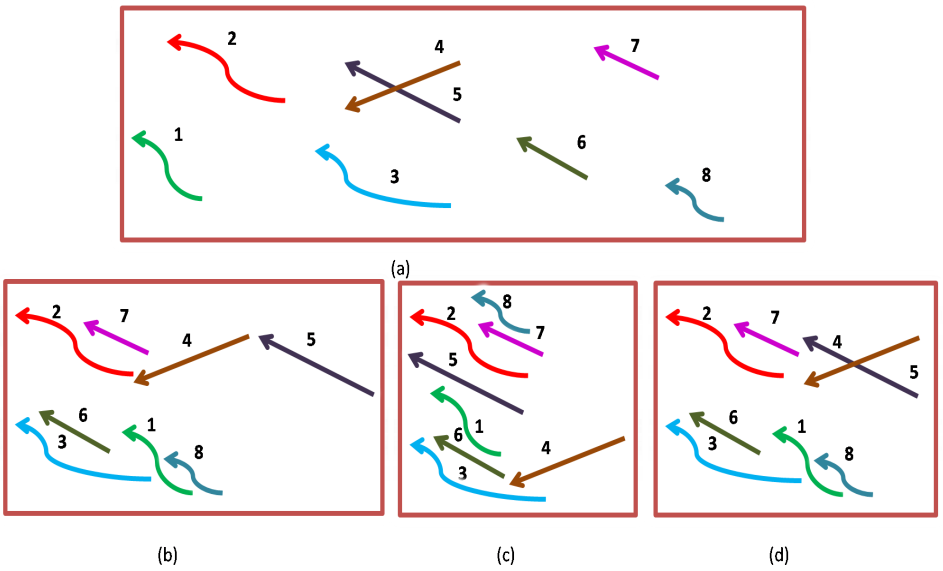


Figure 5.2: Representation of different video synopsis approaches. (a)Trajectories from original video (b) Synopsis video with time shift only (c) Synopsis with time as well as space shift (d) Synopsis using proposed approach preserving interaction between object 4 and 5.

information about moving objects present in a video not only helps in generating a summary of the activities in a video, but it also avoids spatial overlap between objects while generating the synopsis video. The approach is tested in the context of condensation ratio while maintaining the interaction between objects. Experimental results over six video sequences show high condensation rate up to 2%.

The segmented object trajectories are stored as spatiotemporal tubes which are a set of 3D tuples (x_i, y_i, t). Tube represents region belonging to an object i in frame t. Tubes store the motion structure of individual moving objects so that they can be arranged over the spatial domain to generate video synopsis. Segmenting the motion structure of different objects helps in separating important activities and producing the synopsis of those activities only. We refer object motion structure and tube interchangeably in this thesis. These tubes are further used to generate video synopsis by arranging them over space. It can also be used for activity-based video indexing.

While arranging the spatiotemporal tubes over the spatial domain, the time shift between tubes may destroy interaction between objects as shown in Figure 5.2(c) for object 4 and 5. Proposed methodology keeps this interaction in synopsis video by merging the interacting tubes and consider them as a single tubeset as illustrated in Figure 5.2(d).

A method is proposed for generating video synopsis, condensing as much information as possible while preserving the interaction between the objects. The object interaction is maintained by finding the interaction point between two objects and merging their tubes. Energy minimization method is used for arranging the object tubes over spatial domain. While arranging the tubeset over space, the cost is normalized using the length of tubes so that the participation of tube in cost function is proportional to the tube size.

5.0.1 Merge Interacting Object Tubes

The difference between the spatial overlapping tubes is considered here as the indication of intersection between the objects in an original video. We find the interaction between tubes by measuring the difference between tubes as given below in Equation 5.1.

$$I_t(i,j) = \begin{cases} 0 & \text{if } d_t(i,j) > k, \\ k - d_t(i,j) & \text{otherwise} \end{cases} \tag{5.1}$$

Where $d_t(i,j) = T_i^t - T_j^t$ is used to compute the distance between tube i and j at time t and constant k is used for considering the minimum distance for interaction. Here, k is taken as 25 which means that the tubes are considered as interacting if they are 25 pixels apart. The tubes having $I_t(i,j)$ other than 0 are merged to form a tubeset.

5.0.2 Background updation strategy

While arranging the object tubes to generate video synopsis it is required to arrange them over the background. To maintain the synchronization of objects with the background scene, it is essential to maintain variations occurring in a background. The background updation strategy used in this work is given in Equation 5.2 which immediately reflects variation in the background.

$$B_k(i,j) = \begin{cases} I_{k-1} & \text{if pixel(i,j) not belongs to motion region,} \\ B_{k-1}(i,j) & \text{otherwise} \end{cases} \tag{5.2}$$

where $B_k(i,j)$ and $B_{k-1}(i,j)$ represent the pixels belonging to k^{th} and $(k-1)^{th}$ background frames respectively. I_{k-1} represents $(k-1)^{th}$ frame of the original video sequence. Figure 5.3 shows an example background image for three different videos. A region of an image is considered as background when there is no changes in pixels belong to that region for a significant amount of time. In Figure 5.3 the background does not contain any object in the second and third image, so it considered as empty. In the first image, vehicles are present into the scene, but they have not shown movement for a significant amount of time so that they will be considered as a part of the background.

Figure 5.3: Background images of three different videos used in generating synopsis video. Initial image sequence without any moving object is selected as a background image for temporal shift of object from original video.

5.0.3 Video Synopsis by Energy Minimization

Energy minimization Rav-Acha et al. (2006); Fu et al. (2014) is widely used and is a traditional technique for generating video synopsis. The objective of energy minimization is defined as a function that assigns a cost for all possible solutions and finds the solution with the lowest cost. While shifting the input pixel to synopsis pixel with temporal shift M. The temporal shift, is assigning the segmented trajectory over different time domains in synopsis video. It is not necessary to attach an activity over same time stamp; an activity can be placed over different time stamps depending upon the availability of space over that time period. Temporal shift allows a pixel to be assigned to a location in synopsis video with different time domain, while shortening the length of synopsis this may lead to information loss when some set of pixel from particular frame may be ignored in synopsis video. The temporal shift M denotes the loss of information while ignoring the information from particular frame. Energy function is formulated to

assign a cost to activity loss and occlusion as follows:

$$E(M) = E_a(M) + \alpha E_o(M) \tag{5.3}$$

Where $E_a(M)$ and $E_o(M)$ are used to represent the activity loss and occlusion across the frames respectively. α is used to assign a relative weight of occlusion. The activity loss of an object is the difference between pixels belonging to object tubes in input video and synopsis video. The occlusion cost represents the area that is shared by tubes in a frame in synopsis video.

Activity cost: As the length of an object tube depends upon its appearance in the video it does not participate equally in synopsis video too. While calculating the activity loss weighted average of pixels belonging to input video and synopsis video is considered as given in Equation 5.4.

$$E_a(M) = \frac{\sum_{t=SFrame_i}^{EFrame_i}((x_i, y_i, t)_o - (x_i, y_i, t)_s)}{Length_i} \tag{5.4}$$

where $(x_i, y_i, t)_o$ and $(x_i, y_i, t)_s$ represent super pixel region belonging to object i in original video and synopsis video respectively. $Length_i$ represents the number of frame where an activity i has occurred as given in Equation 5.5.

$$Length_i = EFrame_i - SFrame_i \tag{5.5}$$

where $SFrame_i$ and $EFrame_i$ are the starting and ending frames of i^{th} activity.
Collision cost: While condensing the activity in an even shorter video it is required that some pixels are shared between tubes. Collision cost is computed by finding the total number of pixels belonging to an object in consecutive frames share space in synopsis video as given in Equation 5.6.

$$C_i = \begin{cases} 1 & \text{if } (x_i, y_i, t)_s = (x_i, y_i, t+1)_s, \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

It is also used to allow the user in defining the number of pixels in tubes that can overlap. Collision cost is normalized by the length of the object tube to ensure equal participation of each object as given in Equation 5.7.

$$E_o(M) = \frac{\sum_{i \in Q} \sum_{j=1}^{S} C_{i,j}}{Length_i} \tag{5.7}$$

The higher value of $E_o(M)$ results in pixel overlapping between two objects that can affect the smoothness of object appearance. Moreover, the smaller value keeps objects well separated, but it generates a longer synopsis video.

The synopsis video generation requires various, pixel based as well as object based computations in controlling the activity and collision cost. It is required to mention a number of frames in original and synopsis video. While considering the object tubes it is important to mention about the number of tubes, starting and terminating frame of each tube and pixels belongs to a tube at frame i. Table 5.1 explains the various notations used in this work.

Table 5.1: Notations used in generating synopsis video

Symbol	**Description**
N	Number of frames in original video
S	Number of frames in synopsis video
Q	Set of tubes
K	Number of tubes
(x_i, y_i, t)	Pixel area of tube i at frame t
$SFrame_i$	Tube i starting frame
$EFrame_i$	Tube i ending frame
$Length_i$	Length of i^{th} tube

Our procedure to generate synopsis video of important activity primarily considers, keeping the maximum information present in original video as the primary objective. The interaction between the objects and abnormal activities is supposed to have maximum information. In this work, we maintain both the interaction between objects and all the important activities. The procedure of summarizing the activity in the synopsis video is explained in Algorithm 3. The synopsis video is generated using five steps. The primary consideration while generating synopsis video is preserving the activities present in the video. The interaction between objects is maintained by merging the tubes related to interacting objects in the first step. In the second step, it is considered that maximum space utilization is achieved by assigning space to the tube to longest to shortest tube length.Further, in fourth and fifth step space has been assigned to arranged tubeset while maintaining the energy minimum. If it generates shorter video by changing the order of video, it is performed by computing the energy cost for corresponding arrangement.

When arranged object activity tubes over a background image, the sum of energy

Algorithm 3 Algorithm to generate synopsis video.

Input: An array of K tubes as $A_i = (x_i, y_i, t)$ where $t = SFrame_i.....EFrame_i$.
Output: Synopsis video S with minimum energy $\sum_{n \in Q} E_n$.
Initialization: $S \longleftarrow \phi$;
1. Merge tube having interaction using Equation 5.1.
2. Arrange important tubes in descending order according to their Length.
3. Process each tubes T_i from ordered list.
4. Find space and time shift for each tube in synopsis video.
5. $S \longleftarrow S \cup T_i$. with $E(M) \in min\{E(M)\}$
6. End.

terms belonging to the individual tube is considered as global energy of a synopsis video. Energy minimization of global energy is performed using greedy optimization. The energy term corresponding to all temporary arrangements is calculated while choosing the local optimal. As outlined in Algorithm 3 it is considered that by treating the tubes in ascending order of their length it helps to keep the global energy term minimized.

5.0.4 Experimental Evaluation

The performance analysis of proposed method is reported on a number of publicly available datasets. These videos have been selected to analyze the effect of different challenging situations occurs while generating the synopsis video. Video1 and Video2 are from the camera installed for outdoor surveillance, Video3 and Video4 containing varying lighting effect in the portion of a scene. Video5 and GroupVideo show activities in the group including the motion of vehicles to illustrate that system are not category specific. CarVideo shows traffic surveillance video having movement of cars of different size so that we can analyze the effect of different object size in generating synopsis video.

Table 5.2 covers the description of these videos and condensation rate (CR) as comparison parameter. CR denotes the percentage at which synopsis video compress the original video as given in Equation 5.8.

$$Cr\% = \frac{S}{N} \times 100 \tag{5.8}$$

where S and N is used to denote the number of frames in synopsis video and original video respectively.

Table 5.2 shows the comparison of our method with Rav-Acha et al. (2006) and Fu

Table 5.2: A summary of video description and result.

Video	**Frames**	**Tubes**	**Rav-Acha et al. (2006)**	**Fu et al. (2014)**	**P1**	**P2**
Video1	21450	223	9.3	7.6	5	2.1
Video2	5408	87	11	6.5	4.4	2.3
Video3	19841	395	13.2	7.3	6.6	3.2
Video4	7983	121	9	8.2	7.7	4.1
Video5	10900	117	12	9.8	6	2
GroupVideo	420	13	11	6.5	4.8	3.6
CarVideo	493	7	12.8	3.2	2.1	1

et al. (2014). We have stated our results as $P1$ and $P2$, where $P1$ is used to represents the result of proposed approach for generating video synopsis while considering the activity loss and occlusion cost normalized according to object length as given in Equation 5.4 and Equation 5.7. The $P2$ represents the result of proposed method for generating video synopsis with interaction preserving criteria.

The proposed method achieved low condensation ratio in comparison to other methods in analysis. The average condensation ratio we obtained is between 2.1 to 7.7 for a typical video having 10 to 17 tubes with no activity loss, with an occlusion of total 400 pixels between the tubes. We can further reduce the condensation ratio by allowing some activity loss and an increase in pixel overlapping.

Figure 5.4, 5.6, and 5.5 are used to represent the frame sequence from three synopsis videos. Here frame sequence in Figure 5.4 shows the frames having cars in the compressed domain from a synopsis video. Figure 5.5 and Figure 5.6 shows frame sequence form synopsis videos depicting simultaneous presence of persons. In all these examples bounding box is used to represents the pixel region belonging to objects. Figure 5.6 show sequence of frames by representing the pixels belongs to the objects using bounding box. The path of the objects in original video is also shown as the trail of points.

5.1 Video Indexing

There is an application of video synopsis that is known as video indexing Rav-Acha et al. (2006). In video indexing, every object is marked with its actual time of appearance in the original video. In a video synopsis, the spatial and temporal information of an object is lost. Sometimes it is required to have a time stamp of

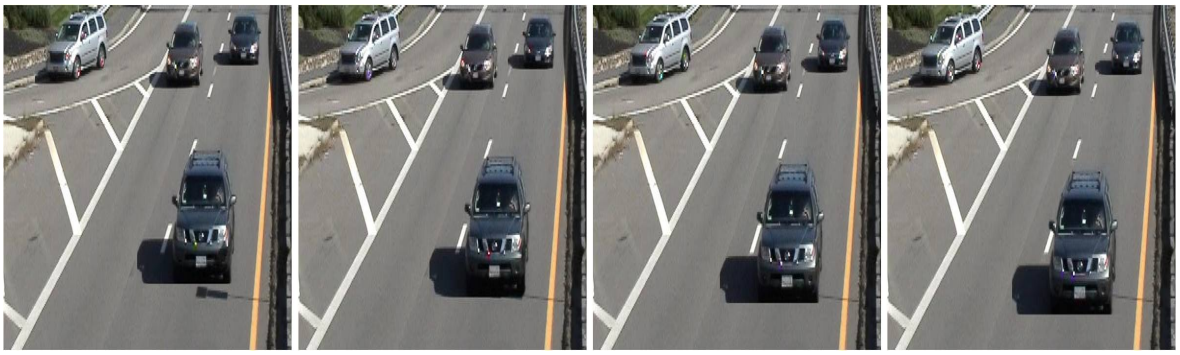


Figure 5.4: Four frames from resulting synopsis video generated from car video. It shows simultaneous occurrence of several car occur at different time instance in original video.

Figure 5.5: Four frames showing activities in synopsis video as individual persons also persons present in group in original video. The resulting synopsis video is generated from video3.

an object appears in the original sequence. Video indexing is an effective way of selecting activities happened during a particular time duration.

While considering the video indexing, when we select an object in video synopsis it redirects us to the original frame sequence assigned to that object as an index. It is required to store the corresponding original frame sequence with each activity present in a video synopsis. In this work the region belonging to each object in a frame is mark as a bounding box. By selecting this region in a synopsis frame, it takes the user to the position in the original video where the corresponding object occurred. It can also be used for generating the activity belonging to the individual object.

The activity related to an object in a video is considered as motion structure. Tubes of different objects are segmented out by separating the motion structure of distinct objects using multiple moving object detection and tracking methods. A three tuples structure represents an indexed object also termed as a tube. In

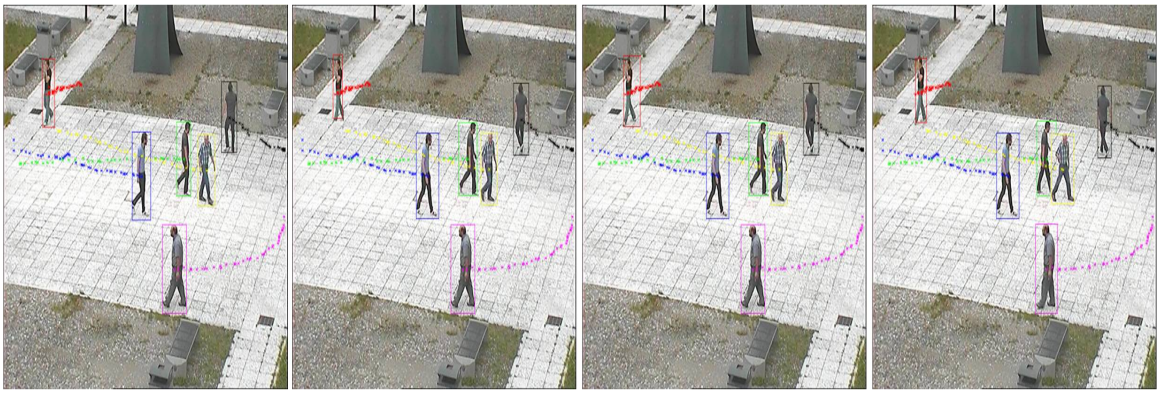


Figure 5.6: Sequence of four frames from synopsis video generated for video5. The path of objects are shown as trailing points and bounding box representing the region belongs to an object.

video analysis based applications, tubes of distinct moving objects are the primary processing components. Here we represent a tube as A_i for an object i, and three tuples represent motion structure of this (i, f, b). Where i represents the index assigned to an object when it first appears into the frame, f represents original frame number, and b is used to store the bounding box as $b = \{x_1, y_1, x_2, y_2\}$ of the region belonging to the object. Where x_1 and y_1 are the location of the top left corner of the bounding box and x_2 and y_2, represent the position of the right bottom of the bounding box. The bounding box helps in finding the location of the object in a particular frame. Each tube A_i is a union of bounding boxes belonging to object i represented by b_i from frame j to frame k as given in Equation 5.9.

$$A_i = \bigcup_{f=j}^{k} T_{(i,f,b_i)} \tag{5.9}$$

where the object i is tracked between frame number j to frame number k. Figure 5.7(b) gives an example of the segmented region belonging to two targets. A tube belongs to an object contains the information about the region it belongs to the sequence of frames. The primary processing is performed over the set of pixels belong to the object. As shown in Figure 5.7(c) an array is required to store the set of pixels belongs to each object. Maintaining the pixel information related to each target increases memory requirement and also make system computationally inefficient. To make system memory efficient and computationally effective only the coordinates of a bounding box are stored in an array instead of storing the set of pixels belonging to a target. The representation of segmented object is depends upon the problem domain. For example, point based tracking or flow segmentation of moving object required the information about centroid of bounding box only

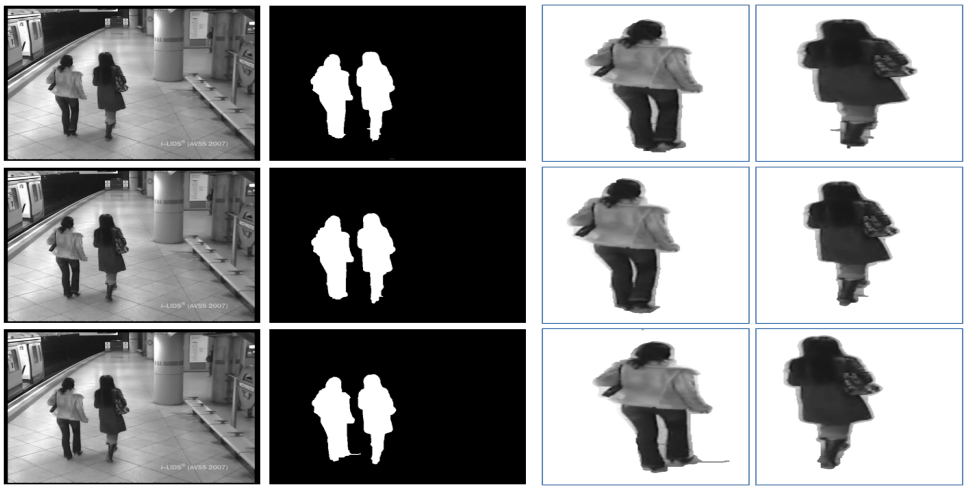


Figure 5.7: (a) Original frame sequence, (b) Foreground segmented frame sequence, (c) Set of pixel belongs to the first moving object, (d) Set of pixels belongs to the second moving object.

as shown in Figure 5.8. It keeps coordinates of the centroid of multiple bounding boxes corresponding to each object present in a frame as shown in Figure 5.8. In Figure 5.8 array is represented as a structure that contains coordinate values of object centroid for five consecutive frame sequences. The structure of frame number 160 is storing the location of three objects in an array.

The video indexing step requires the location and segmented region of an individual object across the frame sequence in the original video. This information is maintained by assigning an array to store the coordinate of bounding box of each object as shown in Figure 5.9. By keeping the information in this manner, it becomes easier to segment activity of individual object. Also, while generating the video indexing, this structure helps in redirecting users to the section where the object appears in the original video. Figure 5.10 shows a sequence of four frames from synopsis video. A frame number labels each object. It represents the frame where that object is belonging in the original video. It is used as indexing of object by redirecting the user to corresponding marked frame in the original video. Figure 5.11 frame sequence from synopsis video by assigning time stamp as an index to each object. The time stamp is representing the occurrence of the object in an original video.

Figure 5.8: Array representation of multiple objects present in five consecutive frames. It shows coordinates of the centroid value of multiple bounding boxes corresponding to each object present in frame sequence.

5.2 Activity Analysis

One of the most important applications of video synopsis is activity classification. As we have already segmented the activities of an individual object in the form of it motion structure tube, it can be applied to classification of activities. By applying the information from video indexing, the activity classification is performed as an extension of this work and the synopsis comprising specific activities is produced.

Automatic activity classification methods from videos for production of metadata is of grave importance for indexing of videos and activity specific segmentation. It may aid in effective indexing of online videos for efficient browsing Chen et al. (2015) and retrieval of high-quality video synopsis, thereby enhancing user experience Money and Agius (2008a). In this work, various parameters are explored for their efficacy in detection of events in videos that are notably significant. The motivation for our work comes from recent attempts of video synopsis generation by analysis of key activities that are present in videos. We make our contribu-

Object ID		
Frame Number	Bounding Box	Centroid
• 160	• (373,159,521,283)	• (447,221)
• 161	• (365,167,521,298)	• (443,233)
• 162	• (359,175,514,303)	• (437,239)
• 163	• (347,178,517,318)	• (432,248)
• 164		
• 165	• (341,193,521,340)	• (431,267)
• 166	• (337,204,516,347)	• (427,275)
• 167	• (323,214,507,360)	• (415,287)
• 168	• (326,221,506,360)	• (416,290)
• 169	• (300,238,525,360)	• (412,299)

Figure 5.9: Structure containing the information of an object across the frame sequence of original video. It stores the location of each object in a frame as bounding box and centroid of that bounding box.

Figure 5.10: Frame sequence from synopsis video assigned a frame number to each object. The frame number used as indexing of object in original video.

tion to this research stream by depicting detecting activities implicitly coherent to events that are of importance in videos. To the best of our knowledge, our work is a pioneer in studying the relationship between abnormal activities and significant events information present in the video.

Although the existing approach of video synopsis works well in condensing activities present in video over space, they do not differentiate normal and significant movement found in the video. While going through the various surveillance videos, it is observed that the particular portion of the video possesses vital information such as information exchange, accidents, theft, and other important activity. This work presents an automatic approach of condensing the specific activities in surveillance video by considering the spatial and temporal relationship between them. In this work, we aim to generate the synopsis of the parts in a video that is seen as interesting. To this end, we interpret and discuss the ability

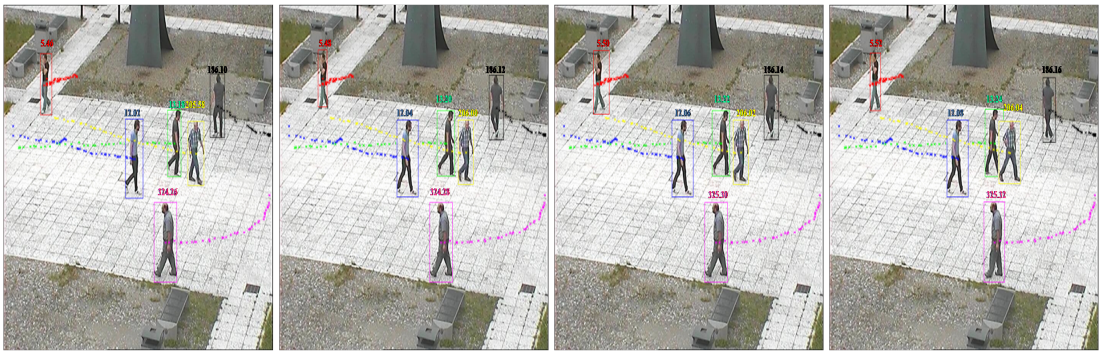


Figure 5.11: Frame sequence from synopsis video assigned a time stamp to each object. Here, time stamp is used as indexing of object in original video.

of computational techniques to spot these interesting events automatically.

The main advantages of detecting important activities using our method are that it is efficient, generalizable as well as scalable. Most of the existing approaches are content based and comprise of expensive extraction of audio visual features followed by machine learning for estimation of their importance level Grabner et al. (2013); Gygli et al. (2014); Ito et al. (2012).

Our approach is computationally efficient in comparison to Grabner et al. (2013) which is appearance based, and requires computation of object structure while we work on the object size that reduces the dimensions, additionally, we have the knowledge of object size and its speed pre-computed (tracks have been already segmented in tracking), which is used in object classification.

Further, our method preserves interaction between objects, so it maintains information intact that is a crucial part in generating synopsis. Last but not the least, it applies to any video genre since generic criterion is chosen for making assumptions about activity importance, unlike most previous works Kim et al. (2014); Mazloom et al. (2015) which are content driven.

5.2.1 Important Activity Model

Our implementation of activity classification, segments object motion structure using moving object detection and tracking and several features are extracted from the accumulated information. This section describes our procedure in detail.

5.2.1.1 Hypothesis on Abnormal Activity

Following hypothesis is formulated for an abnormal activity of user: When a user is following normal behavior, no significant movement will occur in the video content. On the other hand, when a user starts to deviate from the average normal behavior on the video, it will be the indication of an important event. We consider these movements when collecting activities to be included in the synopsis, which we describe below. **Normal Activity Collection:** Features are selected for recording the average normal behavior of the objects throughout the video session. Session starting and ending corresponds to the user entering a specific region in a video and leaving that region respectively in the video.

Processing of the data performed as follows:

1. During video playing, comparison of current activity is made with the normal one.
2. If there is a change in energy, the event registration is done as "important movement".
3. Otherwise, the event registration is done as "normal movement".

Conversion of all instances to their average value within that particular video is done to ensure important events are not associated with any particular genre which results in a global representation of important events. Computation of the final important activity score is done by combining scores obtained using different features. The considered features are linearly combined to obtain the final interestingness score.

5.2.2 Notation and Preliminaries

An introduction is provided here about the notations and general structure used throughout the thesis for the ease of understanding. The state vector S of all N targets in a sequence of F frames consists of the three tuples structure as explained in section 5.1. The various notations used in the thesis are summarized in Table 5.3.

Table 5.3: Notation used in activity analysis.

Symbol	Description
S	State of all targets in all frames
S_i	State of target i in all frames i.e. $\{C_i, s_i, e_i\}$
C_i	Location of target i in all frames
C_i^t	Location of target i in frame t i.e. $\{x_i^t, y_i^t, W_i^t, H_i^t\}$
x_i^t	Minimum x coordinate of target i in frame t
y_i^t	Minimum y coordinate of target i in frame t
W_i^t	Width of target i in frame t
H_i^t	Height of target i in frame t
s_i	Start frame of target i
e_i	End frame of target i
F, N	Total number of frames and targets

5.2.3 Continuous energy model

Energy minimization methods have gained popularity in applications in a way or other. In general, a function is set to determine cost/energy of possible solutions, and afterward, lowest cost state is approximated. This feature is in many ways in relevance to the application under consideration.

Two problems are associated with video/image processing applications:

1. Robust models are required for noisy input data;
2. Capturing all relevant features of actual real time situations for obtaining a function that is an accurate representation of the actual results in a highly complex function.

The second alternative is explored in our work in the analysis of surveillance videos for detecting activities of multiple objects. The function of energy is proposed which has been developed emphasizing on a precise description of various spatial and temporal features. This approach is particularly successful in analysis of situations having multiple targets.

Our energy function is a linear combination of five individual terms:

$$E = E_{Per} + E_{Reg} + E_{spatial} + E_{Interaction} + E_{Dyn} \tag{5.10}$$

where E_{Per} denotes the energy term assigned to each track on the basis of its closeness to boundary region. It ensures that only those tracks are considered as

activities that satisfy the condition of significant tracks given in Equation 5.11. It is used to keep solution closer to the tracking observations. E_{Reg} represents the energy belongs to the track of significant length as given in Equation 5.12. It is used to enforce physical constraints. Usage of energy term E_{Reg} is for capturing the temporal information of different objects for unambiguous object motion. The term $E_{spatial}$ represents the energy that depends on the size of the object which captures the variation in object size from average object size. The energy of interaction $E_{Interaction}$ is used to keep a record of the important activity sections in video and E_{Dyn} is used to provide a generic solution by assigning the energy to objects proportional to the variation in velocity. Figure 5.12 illustrates the model of binary classification of a track as important activity.

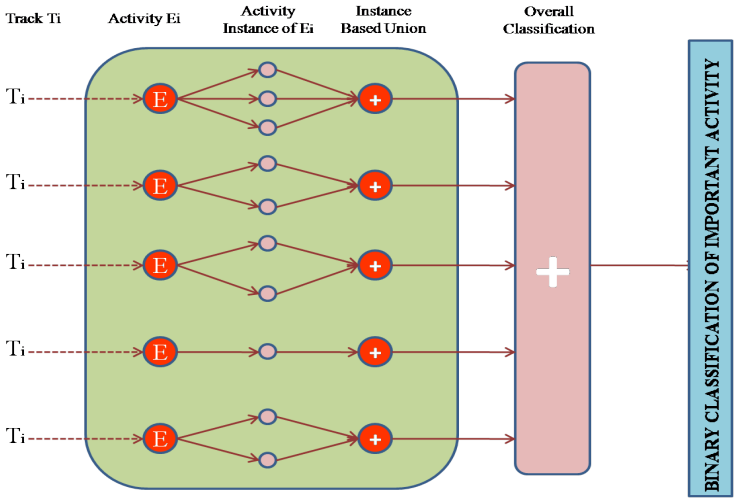


Figure 5.12: Our model for binary classification of a track as normal or important activity using multiple instances of different features.

5.2.4 Important Activity Tracking Model

The focus of our work is on moving objects as activity targets. The dynamic behavior term is used to converge trajectories towards normal observations. The selection of reliable track is done by applying the boundary condition over starting and ending location of track i i.e., over s_i and e_i.

5.2.4.1 Significant Tracks

There can be fragmentation of a track or a track may terminate abruptly in the middle of the tracking area if there are missing evidence. For example, trajectories

starting and ending along image borders or a predefined perimeter are shown in Figure 5.13.

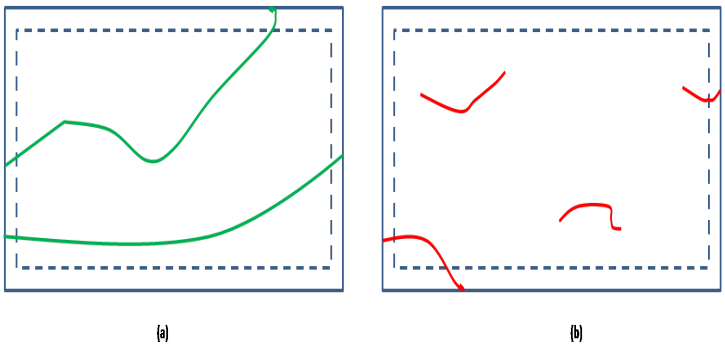


Figure 5.13: (a) Depiction of persistent tracks satisfying boundary conditions. (b) Fragmented tracks are not to be considered to be included in the synopsis video.

In the tracking area, a sigmoid having center on the border is used to keep the term smooth and robust and smooth:

$$E_{Per}(X) = \sum_{i=1,\ldots,N, t\in(s_i,e_i)} \frac{1}{1 + exp(-q \times b(X_i^t) + 1)} \tag{5.11}$$

where $b(X_i^s)$ and $b(X_i^e)$ measure distance of first and last known location of i^{th} target respectively to the border that is closest in the tracking area; and the parameter q is used for representing the thin entry margin. This is set to $q = \frac{1}{s}$, where s represents target size.

There is a need for a regulator for preventing insignificant or incomplete tracks to be excluded from the synopsis as fragmented tracks are not considered as useful information. In order to attain this, the incomplete targets are penalized by imposing criteria over starting and ending of a track. The inclusion of trajectory length in significance term provides better performance, and many short track solutions are less likely. These two terms are combined to compute $E_{Reg}(X)$

$$E_{Reg}(x) = Length_x \times \phi \tag{5.12}$$

Where $Length_x$ represent the length of a track as given in Equation 5.5 and ϕ is used to assigned weight to each track with respect to their length. The range of ϕ is 0 to 1. The weight of the ϕ can be adjusted individually for accommodating trajectory length importance. Empirically it can lead to a slight improvement in performance on some video sequences. For our experiments, it has been set to

1, with the premise that, keeping less number of parameters expedites: searching better parameters while avoiding fitting.

5.2.4.2 Spatial Classification

2-Dimensional object classifier segments objects in an image based on a division of a scale defined by a single measurement feature, in units defined by user-specified spatial calibration. An adaptive spatial measurement attribute is used for classification of the uncommon object. The system is trained with an initial sequence to set the range for the specified measurement attribute or a previously saved input file can be loaded to enter the scale limit values within the input fields automatically. The output (fully automatic) from this object classifier includes the mean and standard deviation of measurement values for all objects for a given region, with a segmented object, analyzed according to its assigned classification. Figure 5.14 illustrates an example of an image analysis using this 2-D object classifier performed according to the size (area) of an object.

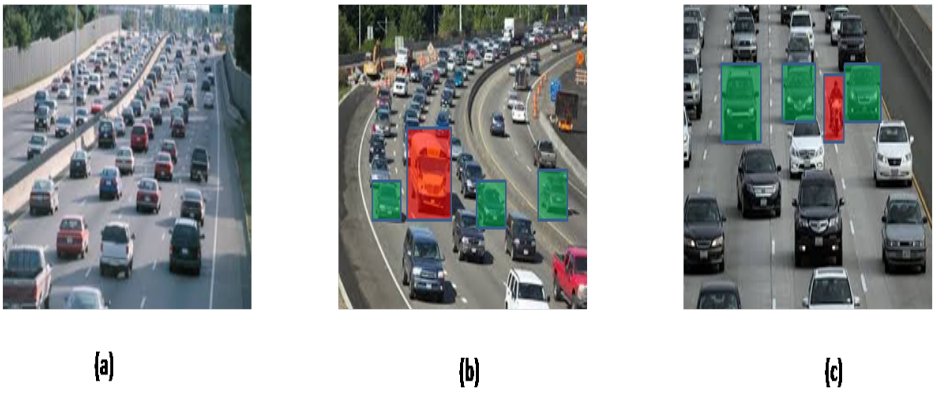


Figure 5.14: (a) Frame with same size objects showing normal activity. (b) Segmented objects marked with the red color having a size greater than mean object size. (c) A segmented object marked with the red color having a size smaller than mean object size.

Due to the more consistent appearance of targets in the image, we focus on the target size models and use them for object size normalization of all objects. To obtain a target size sample, the user can use the region based segmentation to add an object model at the desired location in the image such that the camera perspective effect at that image location is properly approximated.

The specified object size sample is denoted as Om, $m \in [1, N]$, where N is the total number of object size samples. For effectiveness, it is required to specify at

least five samples across the image. Once the system segments sufficient object size models, the predicted size distribution of image sequence is initialized by interpolating the specified size of samples. Given a video sequence, its predicted mean object size is denoted as S computed as,

$$\overline{S} = (1 - \alpha)\overline{S} + \alpha O_i \tag{5.13}$$

where α is used to control learning rate of mean object size $\overline{S}$. Variance is calculated as

$$\overline{\sigma}^2 = (O_i - \overline{S})^2 \tag{5.14}$$

To classify objects according to their scale feature each object size is compared with mean object size and if the difference is more than $2.5 \times \sigma$ it is considered as abnormal.

5.2.4.3 Temporal Features

We considered evidence arising from per-object velocity. Average velocity of each object in each time segment (T) is estimated. Temporal descriptors are constructed for each tracker at time t for an empirically chosen fixed length duration $T = [t - 31, t + 32]$ (roughly 2 seconds $\approx$ 64 frames).

Each segment of a track is classified independently. The velocity of moving object in the sequence frames is defined in pixels/second. We determine the velocity of an object over a segment of 64 frames so that for different positions of a frame we have a separate feature value to compare with. As an object may change its velocity over various regions of a frame, we divide the frame into segments to compute object velocity for each segment individually. The temporal features is represented by E_{Dyn}.

5.2.4.4 Interacting Object Tubes

Interaction in between the objects is a crucial aspect when tracking and analyzing behavior of multiple targets. In our model a continuous weight is applied to configurations where two targets come too close to each other as:

$$d(X_i, X_j) = \sum_{t=1}^{F} \sum_{i,j \neq i}^{N(t)} \| X_i^t - X_j^t \|^2 \tag{5.15}$$

where $d(X_i, X_j)$ represent the distance between track i and j. X_i^t and X_j^t is used to denote the location of track i and j in frame t respectively.

The difference between the overlapping spatial tubes is considered here as the indication of intersection between the objects in an original video. We find the interaction between tubes by measuring the difference between tubes as given below in Equation 5.16.

$$E_{Interaction}(X_i, X_j) = \begin{cases} 0 & \text{if } d(X_i, X_j) => k \ , \\ k - d(X_i, X_j) & \text{otherwise} \end{cases} \tag{5.16}$$

Here k is taken as 25 which means that the tubes are considered as interaction if they are 25 pixels apart. The tube X having $E_{Interaction}(X_i, X_j)$ other than 0 is included into the synopsis of important activities.

5.2.5 Important Activity Scores

The important activity score is calculated by averaging the values obtained for each term explained in Equation 5.10. Each value for corresponding activity criteria is first normalized to have zero mean and one standard deviation. To classify each activity as important or normal a threshold is applied to the average of this normalized score. A non-zero value belonging to any of the energy terms is considered as an important activity and added to synopsis video.

5.3 Summary

An approach to generate video synopsis of surveillance video is presented in this chapter. While existing methods condense activity in the video using shifting of object tubes over time and space, they do not maintain interaction between the objects.

The interaction between object tubes was calculated using distance measure and there after those tubes were merged. The cost function for activity loss and occlusion is computed for energy minimization problem. A weighted average of the cost function is taken to compute participation of each object. The experimental results generated by the proposed approach over the different video sequences

provide promising results. This chapter also explains an array representation of object motion structure. This structural information is used for video indexing.

Furthermore, an automatic approach to generate video synopsis of important activities present in the surveillance video is also explained in this chapter. We presented a procedure for motion trajectory-based statistical modeling and classification of activities captured from any form of object tracking. The strength of this technique is its robustness and independence from a video specific category. In the next Chapter 6 we conclude the thesis.

Chapter 6

Conclusions and Future Work

This thesis work presents a step towards a better integration of functional stages in generating synopsis video. In this research work, we study various state-of-art methods used in the different stage of synopsis video generation and find their limitation in various challenging environments. Further, we formulate techniques which aim to generate better synopsis by optimizing results at each step. These techniques can be implemented in any language that supports image processing, and it is not prescriptive of any particular platform or tool.

In this Chapter, we present the summary and conclusions drawn from our research study and the possible improvements as future work to extend this work.

6.1 Summary of The Work

In Chapter 2, a detailed literature survey of existing work for different functional blocks in synopsis video generation technique is provided. The chapter explores the various challenges present in the complete process of video synopsis. It also addresses the limitations of the widely used methods for each of these phases. The chapter illustrated the current video-based activity analysis techniques in multidisciplinary research such as surveillance systems, border security, activity classification, etc.

In Chapter 3, we described the various applications and necessity of moving object detection in video analysis system. The complete description of the background subtraction model for moving object detection has been presented. This chapter further introduced the detailed description of basic Codebook model for foreground

segmentation. The observation also brought in focus that learning parameters used in this model can be made adaptive for getting better accuracy. An improved Codebook model is implemented for foreground segmentation with a dynamic background. This chapter also reported the comparative analysis of our technique with state-of-the-art methods.

In chapter 4, we employed the procedure for multiple object tracking to achieve better performance than the existing methods in challenging environments like crowded sequence and miss detection. The procedure starts by proposing a coupling technique between point based and appearance based tracker. Motivated by object detection and tracking studies, a target specific multiple instance appearance model is developed for a particular scenario where only a portion of an object is visible.

Chapter 5 concerns with the final process of generating synopsis video. The process of implementation of the methodology describes the activity cost and collision costs as decision criteria of smoothness of resulting video. Furthermore, the chapter showed the functionality and usability of the our implementation for the efficient and fast retrieval of stored information.

In this chapter we also explored the methodology and the technique that is used for generating video indexing. Video indexing is performed by representing the motion structure of an object as an array. Further, we have formulated a method for activity classification by trajectory analysis. The event classification is performed via similarity measure with normal behavior. The normal activities are modeled as a set of features defined for each spatial, temporal, consistency and interaction behavior. Our techniques can be applied to any surveillance video and does not limit to specific video class.

6.2 Conclusions

In this work, we present an efficient system to generate synopsis video. We have explored different phases involved in the generation of synopsis video and optimization is performed over each step.

The major contributions of our research work can be summarized as follows:

1. We have found that the task of moving object detection and segmentation remains challenging when there are moving/changing a background, varia-

tion in illumination, shadow effects and multiple moving objects (crowded sequence). Codebook model is a widely used pixel-based background subtraction method for segmenting moving object. However, it often engenders erroneous positive results in the case of a dynamic environment. We formulate an adaptive multi-layer codebook model for foreground segmentation in a video by implementing the improvements in basic codebook model. Our method gives better results than standard codebook model by achieving high values of precision and recall.

Following implementations obtain these improvements:

(a) Adaptive weights assigned to each codeword used for deciding the threshold for deleting or shifting codeword category between layers.

(b) Adaptive decision threshold is calculated to overcome challenges introduced due to background dynamics.

(c) A Layer is added to maintain information about static uncovered background which helps in eliminating false positives termed as ghost region.

(d) Random neighbor selection policy is used in spatial context to avoid processing overhead in matching codewords using a 4-connected neighborhood for deciding foreground pixels.

(e) The cone-shaped color distance measure instead of the cylinder based color distance measure is used which helps to achieve better accuracy against illumination variations due to normalization by σ. Our method achieves an overall better performance when compared with other state-of-the-art methods.

2. We have found that feasible solutions are available for addressing the robust tracking of the single target. However, simultaneous analysis of multiple targets motion structure in a video stays as one of the most challenging tasks in computer vision. The accuracy of different tracking methods reduces in situations like object with random movement, an interaction between objects, scale variation, and occlusion. To achieve high performance in multiple object tracking, we have applied following enhancement:

   (a) We have presented a coupling procedure between data association and target specific appearance based tracker for multi target tracking, which includes explicit occlusion reasoning and appearance modeling.

   (b) It combines the outcomes of object detection with data association discriminative tracking for estimation of promising particles, taking advantage of the often complementary nature of the two sub problems.

(c) Prior estimation of the particle through detection introduces sparse natural behavior in selection. We use particle filter-based tracker with color and gradient as a feature to map posterior state to calculate the distance between reference particle color histogram with particles belonging to current observation.

(d) Our coupling based tracking approach enables us to model the motion structure of target that helps in driving heuristics for the likelihood of appearance using non-maxima suppression.

(e) By incorporating detection result in generating proposal distribution also reduces the chance of selecting particles from background region and distributing them around target region.

(f) The coupling procedure is characterized by a small number of false detection, significant reduction in identity switches and joins segmented trajectories. Furthermore, in a less crowded environment when targets are well separated only data association can achieve competitive results that lead to low computational complexity.

(g) Quantitative and qualitative results on benchmark video sequences show that our coupling model surpass state-of-the-art methods used for multiple moving object tracking.

3. Traditional video analysis methods generate a summary of day long videos, but maintaining the motion structure and interaction between objects is a grave concern to researchers. In this work:

   (a) The interaction between object tubes was calculated using distance measure to merge the tubes.

   (b) The cost function for activity loss and occlusion is computed for energy minimization, where a weighted average of the cost function is applied to identify participation of each object.

   (c) The experimental results generated by our method over the different video sequences give promising results.

4. We have used the information from multi-object tracking step in generating the array representation of object motion structure. This structural information is used for video indexing. The reason behind using this array representation is low memory requirement and faster access. While storing the pixels values belonging to each object in a video require a lot of memory,

we represent the location and frame number of an object in an array, which saves a lot of memory.

5. Although the existing approaches of video synopsis work well in condensing activities present in video over space, they do not differentiate normal and important movement found in the video. While going through the various surveillance videos, it is observed that the particular portion of the video possesses vital information such as information exchange, accidents, theft, and other important activity. In this work, we have presented an automatic approach of condensing the specific activities in surveillance video by considering the spatial and temporal relationship between them.

   We have implemented a effective procedure for motion trajectory-based statistical modeling and classification of activities captured from any form of object tracking. The strength of this technique is its robustness and independence from a video specific category.

6.3 Limitations and Future Work

There are many current research opportunities and new challenges open with this work for future action. The techniques implemented in this work for different stages of video synopsis can be used as basic building blocks for many video analysis tasks. In continuation of the above research work done, some of the future research perspectives are as follows:

- In our method we generate multiple codewords for each pixel to represent background model. Any background pixel may have shared some similarity with its neighboring pixels. Incorporating similarities of neighboring connected pixels can be a significant extension of our work.
- For dynamic background, it can be a challenging task to generate background model due to uncertain variations in spatial geometry. Pyramid based codeword matching scheme can be a possible solution to overcome this.
- Object detection can be coupled with tracking task instead of treating them independently to counter miss detections occuring due to occlusion.
- To improve the performance of tracking methodology, curve fitting mechanism can be used along with the existing technique.

- Tracking methods can be further extended for multiple camera tracking scenarios.
- A user driven query based event selection to classify important activities is another attractive future extension.
- An interactive synopsis video can be a future direction in generating synopsis videos, with the capability to select or query about a particular event using touch screen monitors to increase user experience of event driven activity analysis systems.

Appendix A

Implementation of The Design

This appendix provides an introduction to the tools and software packages used in this dissertation work. It also gives the detailed description of the program implementation of methods proposed in previous chapters to understand the input and output structure of various phases. The choice of Matlab as a programming environment and functions used to implement multiple modules in this work is given in this chapter.

This chapter details the design and experimental details of four module involving: object detection, multiple object tracking, synopsis video generation and activity analysis. We preferred a four-level design in this work instead of a design where only one stage implementation is performed. It gives us the opportunity to optimize each module independently without affecting the others.

A.1 Platform Used

The implementation of the proposed methodology in this thesis is done in MATLAB. MATLAB is preferred for experimentation because of the following reasons:

- A huge repository of built-in algorithms for image processing and computer vision applications.
- Its interpretative nature allows you to test algorithms immediately without recompilation.

- The MATLAB Desktop environment, allows you to work interactively with your data, helps you to keep track of files and variables and simplifies common programming/debugging tasks.
- The ability to read in a wide variety of both common and domain-specific image formats.
- The ability to process both still images and video.
- The ability to auto-generate C code, using MATLAB Coder, for a large (and growing) subset of image processing and mathematical functions, which you could then use in other environments, such as embedded systems or as a component in other software.

A.2 Toolbox Used

This work primarily deals with data in the form of video and images. For this purpose, we have used mainly Image Processing Toolbox and Computer Vision System Toolbox. Image Processing Toolbox provides a comprehensive set of reference-standard algorithms and workflow applications for image processing, analysis, visualization, and algorithm development.

A.3 Description of Functions Used

Details of the functions used in the implementation of object detection are as follows:

- **videoFReader:** vision.VideoFileReader(FILENAME) returns a video file reader System object, videoFReader. The object can sequentially read video frames and/or audio samples from the input video file, FILENAME. Every call to the step method returns the next video frame.
- **size:** Return the dimensions of array. Mainly used to get the size of image.
- **step:** Advance simulation by one more step. To read next frame from video.
- **min:** Function returns smallest element in an array.
- **max:** Function returns largest element in an array.

- **cat:** Concatenate array along specified dimension.
- **xlswrite:** write Microsoft Excel spreadsheet file.
- **sprintf:** Format data into string.
- **imwrite:** Write image into graphics file.
- **imread:** This function reads a grayscale or color image from the file specified by the string filename.
- **zeros:** Create an array of all zeros.
- **sqrt:** This function returns the square root of each element of the array X.
- **randi:** This function returns an n-by-n matrix containing pseudorandom integer values drawn from the discrete uniform distribution on the interval [1,imax].
- **histc:** This function counts the number of values in vector x that fall between the elements in the edges vector (which must contain monotonically nondecreasing values).
- **struct:** This function creates a scalar (1-by-1) structure with specified fields.
- **load:** This function loads the variables from a file into a structure array, or data from an ASCII file into a double-precision array.
- **insertmarker:** This function returns a truecolor image with inserted plus (+) markers.
- **insertshape:** This function returns a truecolor image with shape inserted.

A.4 Object Detection

This section provides the detailed description of program implementation related to object detection. This procedure is used to implement the adaptive multi-layer background subtraction method by implementing various improvements over fundamental codebook model. The detailed description of the proposed method for background subtraction is given in Chapter 3. The dataset used for experimentation of moving object detection is given in Section 3.4.0.1. The parameters used in this procedure are explained in Section 3.4.1.

A.4.1 Pseudocode: Object Detection Using Adaptive Background Subtraction Technique

The pseudocode of proposed methodology used in this thesis for moving object detection is given in Algorithm 4. The input for this program is video sequence containing moving objects. The output is in the form of mask image of foreground segmented object in corresponding frames in input video. The detailed description of the methodology is given in Chapter 3.

Algorithm 4 Object Detection Using Background Subtraction

Input: (Frame Sequence)

Output: Mask Image

Procedure: Adaptive Codebook Model

Training Phase:

Initialize Auxiliary Variables

MAX=7 *% Max value of codewords in M*

H_Th=50 *% Threshold to filter codewords from H*

alpha=0.5; *% 0.4 to 0.7*

beta=1.3; *% 1.1 to 1.5*

$t_k = 100$ *%number of training frames*

e1=15;

e2=25;

for Read 100 frames as t=1 to 100 of input video **do**

% Defining codewords structure.... considering their indexes as their C_m codeword number : index 1 is C_1 codeword, index 2 is C_2 codeword. so index m is C_m codeword

Initialize codeword tuples

l=cell(ht,wt) % total number of codewords in M of pixel(i,j)

v=cell(ht,wt,3,1) % [pixel(i,j)],[1=r,2=g,3=b],[1=pointer to codewords-set of pixel-i,j]

Imin=cell(ht,wt,1) % Minimum Intensity Value

Imax=cell(ht,wt,1) % Maximum Intensity Value

f=cell(ht,wt,1) % Frequency of codeword

MNRL=cell(ht,wt,1) % lambda=Maximum Negative Run Length

p=cell(ht,wt,1)

q=cell(ht,wt,1)

$color_{dist} = cell(ht, wt, 1)$

brightness=cell(ht,wt,1)

l(:) = 0; % initializing total number of codewords for all pixel to zero

%end of initialization of M

% FIND matching C_m

for 1 to L % search all codewords in M for match **do**

% BRIGHTNESS calculation

if $Xtr \leq IhighandXtr \geq Ilow$ **then**

$brightnessi, j, 1(i1) = 1$ % TRUE

else

$brightnessi, j, 1(i1) = 2$ % FALSE

end if

UPDATION of C_m

if $color_dististi, j, 1(i1) <= e1)\&(brightnessi, j, 1(i1) == 1$ **then**

Update tuples belonging to the matching codeword in M

end if

% Refined background model to remove foreground objects during training
$Tu = \frac{t_k}{2}$ % Threshold: Half of the training frames
% Adding Codewords to the background model M having maximum negative run length less than threshold value
for Each pixel find codeword assign in initialization **do**
  for All codeword in M **do**
    if $(MNRLi, j, 1(m) <= Tu)$ **then**
      % if λ(MNRL) is less than some threshold, then only allow it in M.
    end if
  end for
end for
%End of training
%This structure is final training data in M
%Multi-Layered Approach
%Define structure for another cache codebook H.
hl=cell(ht,wt)
hv=cell(ht,wt,3,1)
hImin=cell(ht,wt,1)
hImax=cell(ht,wt,1)
hf=cell(ht,wt,1)
hMNRL=cell(ht,wt,1)% λ
hp=cell(ht,wt,1)
hq=cell(ht,wt,1)
$hcolor_dist = cell(ht, wt, 1)$
$hbrightness = cell(ht, wt, 1)$
hl(:) = 0; % Initially number of codewords in H is zero

FOREGROUND DETECTION STEP

```
fg = zeros(ht,wt) % output frame
for Read each frame in video after training frames do
  for Read each pixel do
    match=0 % Default value for match is zero.
    I = sqrt(double((R * R) + (G * G) + (B * B)))
    Call Procedure(RandomNeighborSelection)
    Find matching codeword in M.
    if Match found then
      fg(i, j) = 1
      update corresponding codeword C_n in M
    else
      Increase λ of non matching codeword
      Find matching codeword in H
      if Match Found then
        Update lambda value in H
      else
        Create a new codeword % No match found
      end if
    end if
    % move the cached codewords staying from enough time in H to M
    for All codeword in H do
      if Matching frequency is greater than threshold value then
        Check if M is full
        if Codebook M having codeword less than limit then
          Add codeword to M
          Update M
          Update codeword tuple values
        else
          Delete codeword having Maximum negative run length in M
          Add codeword to M
          Update M
          Update codeword tuple values
        end if
      end if
    end for

    for All codeword in H do
      if Matching negative run length is greater than threshold value then
        Delete codeword from H
        Update H
      end if
    end for
```

A.4.2 Pseudocode: Random neighbor selection

The procedure for selecting random neighbor is given in Algorithm 5. The detailed explanation about random neighbor selection is given in Section 3.3.3.

Algorithm 5 Random Neighbor Selection

Input: (x,t,M)% x is input pixel, t is the frame number and M is background codebook model at pixel x.

Output:Match

Procedure: RandomNeighborSelection

for each input pixel x at frame t, $x_t = (R, G, B)$, $\|x\| = \sqrt{R^2 + G^2 + B^2}$ **do**
  Find a matching codeword to x_t in background codebook M
  if found **then**
    Match=1, then update the codeword
  else
    match=0
    Select $random(y) \in N_x$
    for each codeword in M_y **do**
      Try to find match with x_t
      if found **then**
        Match =1
      end if
    end for
  end if
end for

A.4.3 Pseudocode: Uncovered background region

The procedure to include pixel into the uncovered background region codebook is provided in Algorithm 6. This procedure is explained in Section 3.3.4.

Algorithm 6 Identification of codeword belonging to uncovered background region.

Input: (x,t,M)
Output:Match
Procedure: UncoveredBackground
for each input pixel x at frame t, $x_t = (R, G, B)$, $\|x\| = \sqrt{R^2 + G^2 + B^2}$ **do**
  Find a matching codeword to x_t in background codebook M
  if found **then**
    Match=1, then update the codeword
    if $freq_m \geq Th_{High}$ **then**
      for each codeword $M_y \in \{M_y | y \in N_x\}$ **do**
        Try to find matching codeword in M_y with x_t
        if found **then**
          Add match codeword in uncovered background layer U.
        end if
      end for
    end if
  end if
end for

A.5 Multiple Object Tracking

The implementation details of online object tracking method is given in Algorithm 7. The input for this procedure is the color image sequence containing multiple objects need to be tracked and the mask image sequence generated by the procedure is given in Section A.4.1. It generates the structure as an output containing labeled object's location as the bounding box in each frame. The description of the proposed methodology used for multiple object tracking is given in Chapter 4.

Algorithm 7 Multiple Object Tracking.

Input: (Mask image sequence, Original image sequence)
Output:Structure of labeled bounding boxes in each frame
Procedure: MultiObjTracking

for Read frame from input video **do**

N = 400 % Number of particules
$N_{bins} = 6$ % Number of bins in each color domain
$N_{hist} = 256$ % Upper limit of Histogram
Bhattindex = 1 %Bhattacharya Index
$N_i = 4$ %Number of instance
%Interval of histogram
bin1 = (0 : range/Nr : range)
% Call Random Procedure to distribute particle over sample region
ranpixel1 = Procedure ranpix(Xmin1,Ymin1,H1,W1,N);
% Call procedure to compute histogram density over bins
C1 = Procedure ColorPDF(im, ranpixel1, Xmin1, Ymin1, H1, W1, N, bin1);
Compute Bhattacharya Coefficients of sample region
% Distribute particle over estimated position
ranpixel2 = Procedure ranpix(Xmin2,Ymin2,H1,W1,N);
% Call procedure to compute histogram density over bins
C2 = Procedure ColorPDF(im, ranpixel2, Xmin2, Ymin2, H1, W1, N, bin1)
%calculate bhattacharya coefficient for each sample set
for Each sample set i **do**
$BhattCoeff(1,i) = BhattCoeff(1,i) + \sqrt{C1(i,j) * C2(i,j)}$
$BhattDist(1,i) = \sqrt{(1 - BhattCoeff(1,i))}$
AvgBhattDist = BhattDist(1,1)+BhattDist(1,2)+BhattDist(1,3) % Mean Distance of color domain
% Select x and y index for which there is minimum Bhattacharya distance
if $AvgBhattDist < MinBhattDist$ **then**
Update parameters for corresponding sample
MinBhattDist = AvgBhattDist
MinXIndex = x
MinYIndex = y
Minranpixel = ranpixel2
C=C2;
end if
end for
Call procedure to display particle for illustration only
Procedure displayparticle(I,ranpixel,Minranpixel)
end for

Algorithm 8 Distribute particle.

Input: (Mask Image, Xmin, Ymin, H, W, N)

Output:Index of random particle distribution

Procedure: ranpix

Procedure [randpixel]=ranpix(BW, Xmin, Ymin, H, W, N)

num=1

Count total number if pixels in segmented region and assign it to num

r = randi(num,1,N) % Generate N random particle from num

randpixel=pixellist(r,1:2) % assign pixel list to randpixel

Algorithm 9 Procedure to compute the color PDF.

Input: (Input image, Xmin, Ymin, H, W, N, bin1)

Output:Weighted color PDF

Procedure: colorPDF

Procedure [C1] = ColorPDF(im , ranpixel, Xmin , Ymin , H, W, N , bin1)

q=ranpixel; s=size(q,1); im=Z; im=im./256;

% Histogram without adding weight to histogram bins

C1 = histc(im2(:,:,1), bin1)

n=size(bin1,2);

C1=zeros(3,n-1)

$a = sqrt((H/2)^2 + (W/2)^2)$

c=[(Xmin+(W/2)) (Ymin+(H/2))]

f=0

for i=1:s % s is number of random particles **do**

  $r2 = \sqrt{\frac{(q(i,1)-c(1,1))^2+(q(i,2)-c(1,2))^2)}{a}}$ % Distance between two bins

  for j=1:3 % for color RGB **do**

    for b=1 : n-1 % number of bins **do**

      if $im(q(i,1), q(i,2), j) \geq bin1(1,b) and im(q(i,1), q(i,2), j) \leq bin1(1, b+1)$ **then**

        $C1(j,b) = C1(j,b) + (1 - r2^2)$

      end if

    end for

  end for

  $f = f + (1 - r2^2)$

end for

$C1 = \frac{C1}{.f}$

A.6 Synopsis Video Generation

The procedure implemented to generate synopsis video of tubes is provided in Algorithm 10. This procedure takes structure t, input video sequence, and background sequence. Where structure t is generated as a result of procedure 7. The explanation of complete procedure is given in Chapter 5.

Algorithm 10 Procedure to generate Synopsis Video

Input: (Tube Structure t, Input video, Background sequence)
Output:Synopsis Video
Procedure: SynVideo

```
Procedure synopsis()
% Generating synopsis by arranging each tube start from first frame.

global t
load moving object structure in t
global s
load moving object structure in s
% structure M is used to assign same track ID to objects with overlapping
bounding box
M=struct('FrameNo',,'Bbox',[[],[],[],[]],'ObjectID',[])
M=s
Orgvid= VideoReader(videoSeq2.avi) % Read Original Video
nframes=orgvid.NumberOfFrames;
[ , col]=size(s);
max=s(1,1).ObjectID(1,1);
for i=1:col do
  [ , c]=size(s(1,i).ObjectID);
  for j=1:c do
    if s(1,i).ObjectID(1,j) > max then
      max=s(1,i).ObjectID(1,j)
    end if
  end for
end for
```

```
for o=1:max do
  for i=1:col do
    [ ,c]=size(s(1,i).ObjectID);
    for k=1:c do
      if s(1,i).ObjectID(1,k)==o then
        bbox=s(1,i).Bbox(k,:);
        frame=imread('backgroundseq2.png');
        filename = sprintf('SFrame');
        frame=imread(filename);
        originalframe=s(1,i).FrameNo;
        if originalframe <= nframes then
          orgframe=read(orgvid,originalframe);
        end if
        x1 = bbox(1,1) - 2
        if x1 <= 0 then
          x1=1
        end if
        if x1 > 704 then
          x1 = 704
        end if
        x2 = bbox(1,1) + bbox(1,3) + 2
        if x2 > 704 then
          x2 = 704
        end if
        y1 = bbox(1,2) - 2
        if y1 <= 0 then
          y1 = 1
        end if
        if y1 > 576 then
          y1=576
        end if
        y2=bbox(1,2)+bbox(1,4)+2
        if y2 > 576 then
          y2 = 576
        end if
        for Y = y1 : y2 do
          for X = x1 : x2 do
            frame(Y,X,:)=orgframe(Y,X,:)
          end for
        end for
        xcenter = bbox(1,1) + bbox(1,3)/2
        ycenter = bbox(1,2) + bbox(1,4)/2
        centroid2(SFrame,1:2)=[xcenter,ycenter]
```

Bibliography

Avlonitis, M. and Chorianopoulos, K. (2014). Video pulses: User-based modeling of interesting video segments. *Advances in Multimedia*, 2014:2.

Babenko, B., Yang, M.-H., and Belongie, S. (2011). Robust object tracking with online multiple instance learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1619–1632.

Badal, T., Nain, N., Ahmed, M., and Sharma, V. (2015). An adaptive codebook model for change detection with dynamic background. In *Signal-Image Technology & Internet-Based Systems (SITIS), 2015 11th International Conference on*, pages 110–116. IEEE.

Bao, X., Fan, S., Varshavsky, A., Li, K., and Roy Choudhury, R. (2013). Your reactions suggest you liked the movie: Automatic content rating via reaction sensing. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 197–206. ACM.

Barnich, O. and Van Droogenbroeck, M. (2011). Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image processing*, 20(6):1709–1724.

Benezeth, Y., Jodoin, P.-M., Emile, B., Laurent, H., and Rosenberger, C. (2008). Review and evaluation of commonly-implemented background subtraction algorithms. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.

Berclaz, J., Fleuret, F., Turetken, E., and Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1806–1819.

Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L. (2009). Robust tracking-by-detection using a detector confidence particle filter.

In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1515–1522. IEEE.

Brutzer, S., Höferlin, B., and Heidemann, G. (2011). Evaluation of background subtraction techniques for video surveillance. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1937–1944. IEEE.

Butt, A. A. and Collins, R. T. (2013). Multi-target tracking by lagrangian relaxation to min-cost network flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1846–1853.

Chen, L., Zhou, Y., and Chiu, D. M. (2015). Smart streaming for online video services. *IEEE transactions on multimedia*, 17(4):485–497.

Collins, R. T., Liu, Y., and Leordeanu, M. (2005). Online selection of discriminative tracking features. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1631–1643.

Czyz, J., Ristic, B., and Macq, B. (2005). A color-based particle filter for joint detection and tracking of multiple objects. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 2, pages ii–217. IEEE.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

Danelljan, M., Häger, G., Khan, F., and Felsberg, M. (2014). Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press.

Danelljan, M., Häger, G., Khan, F. S., and Felsberg, M. (2017). Discriminative scale space tracking. *IEEE transactions on pattern analysis and machine intelligence*, 39(8):1561–1575.

Dhar, S., Ordonez, V., and Berg, T. L. (2011). High level describable attributes for predicting aesthetics and interestingness. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1657–1664. IEEE.

Ellis, A. and Ferryman, J. (2010). Pets2010 and pets2009 evaluation of results using individual ground truthed single views. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 135–142. IEEE.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645.

Fortmann, T. E., Bar-Shalom, Y., and Scheffe, M. (1980). Multi-target tracking using joint probabilistic data association. In *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, pages 807–812. IEEE.

Fragkiadaki, K. and Shi, J. (2011). Detection free tracking: Exploiting motion and topology for segmenting and tracking under entanglement. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2073–2080. IEEE.

Fu, W., Wang, J., Gui, L., Lu, H., and Ma, S. (2014). Online video synopsis of structured motion. *Neurocomputing*, 135:155–162.

Führ, G. and Jung, C. R. (2014). Combining patch matching and detection for robust pedestrian tracking in monocular calibrated cameras. *Pattern Recognition Letters*, 39:11–20.

Gordon, N., Ristic, B., and Arulampalam, S. (2004). Beyond the kalman filter: Particle filters for tracking applications. *Artech House, London*, 830.

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

Grabner, H., Nater, F., Druey, M., and Van Gool, L. (2013). Visual interestingness in image sequences. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 1017–1026. ACM.

Guo, J.-M., Liu, Y.-F., Hsia, C.-H., Shih, M.-H., and Hsu, C.-S. (2011). Hierarchical method for foreground detection using codebook model. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(6):804–815.

Gygli, M., Grabner, H., Riemenschneider, H., and Van Gool, L. (2014). Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer.

Hung, M.-H., Pan, J.-S., and Hsieh, C.-H. (2014). A fast algorithm of temporal median filter for background subtraction. *Journal of Information Hiding and Multimedia Signal Processing*, 5(1):33–40.

Ito, Y., Kitani, K., Bagnell, J., and Hebert, M. (2012). Detecting interesting events using unsupervised density ratio estimation. In *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pages 151–161. Springer.

Jacobs, N., Roman, N., and Pless, R. (2007). Consistent temporal variations in many outdoor scenes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE.

Julier, S. J. and Uhlmann, J. K. (1997). New extension of the kalman filter to nonlinear systems. In *AeroSense'97*, pages 182–193. International Society for Optics and Photonics.

KaewTraKulPong, P. and Bowden, R. (2002). An improved adaptive background mixture model for real-time tracking with shadow detection. In *Video-based surveillance systems*, pages 135–144. Springer.

Kamvar, M., Chiu, P., Wilcox, L., Casi, S., and Lertsithichai, S. (2004). Minimedia surfer: browsing video segments on small displays. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1371–1374. ACM.

Kim, J., Nguyen, P. T., Weir, S., Guo, P. J., Miller, R. C., and Gajos, K. Z. (2014). Crowdsourcing step-by-step information extraction to enhance existing how-to videos. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pages 4017–4026. ACM.

Kim, K., Chalidabhongse, T. H., Harwood, D., and Davis, L. (2005). Real-time foreground–background segmentation using codebook model. *Real-time imaging*, 11(3):172–185.

Kuo, C.-H. and Nevatia, R. (2011). How does person identity recognition help multi-person tracking? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1217–1224. IEEE.

Lahraichi, M., Housni, K., and Mbarki, S. (2016). Bayesian detection of moving object based on graph cut. In *Intelligent Systems: Theories and Applications (SITA), 2016 11th International Conference on*, pages 1–5. IEEE.

Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., and Savarese, S. (2014). Learning an image-based motion context for multiple people tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3542–3549.

Leal-Taixé, L., Milan, A., Reid, I., Roth, S., and Schindler, K. (2015a). Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*.

Leal-Taixé, L., Milan, A., Reid, I., Roth, S., and Schindler, K. (2015b). Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*.

Lee, Y. J., Ghosh, J., and Grauman, K. (2012). Discovering important people and objects for egocentric video summarization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1346–1353. IEEE.

Lee, Y. J. and Grauman, K. (2015). Predicting important objects for egocentric video summarization. *International Journal of Computer Vision*, 114(1):38–55.

Li, W., Wu, X., Matsumoto, K., and Zhao, H.-A. (2010). Foreground detection based on optical flow and background subtract. In *Communications, Circuits and Systems (ICCCAS), 2010 International Conference on*, pages 359–362. IEEE.

Li, X., Wang, Z., and Lu, X. (2016). Surveillance video synopsis via scaling down objects. *IEEE Transactions on Image Processing*, 25(2):740–755.

Lipton, A. J., Fujiyoshi, H., and Patil, R. S. (1998). Moving target classification and tracking from real-time video. In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pages 8–14. IEEE.

Liu, G., Chen, Z., Yeung, H. W. F., Chung, Y. Y., and Yeh, W.-C. (2016). A new weight adjusted particle swarm optimization for real-time multiple object tracking. In *International Conference on Neural Information Processing*, pages 643–651. Springer.

Liu, H., Hong, T.-H., Herman, M., Camus, T., and Chellappa, R. (1998). Accuracy vs efficiency trade-offs in optical flow algorithms. *Computer vision and image understanding*, 72(3):271–286.

Lo, B. and Velastin, S. (2001). Automatic congestion detection system for underground platforms. In *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pages 158–161. IEEE.

Lu, X. (2014). A multiscale spatio-temporal background model for motion detection. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 3268–3271. IEEE.

Lucas, B. D., Kanade, T., et al. (1981). An iterative image registration technique with an application to stereo vision.

Mashak, S. V., Hosseini, B., Mokji, M., and Abu-Bakar, S. A. R. (2010). Background subtraction for object detection under varying environments. In *Soft Computing and Pattern Recognition (SoCPaR), 2010 International Conference of*, pages 123–126. IEEE.

Mazloom, M., Habibian, A., Liu, D., Snoek, C. G., and Chang, S.-F. (2015). Encoding concept prototypes for video event detection and summarization. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 123–130. ACM.

McFarlane, N. J. and Schofield, C. P. (1995). Segmentation and tracking of piglets in images. *Machine vision and applications*, 8(3):187–193.

Milan, A., Leal-Taixé, L., Schindler, K., and Reid, I. (2015). Joint tracking and segmentation of multiple targets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5397–5406.

Milan, A., Schindler, K., and Roth, S. (2016). Multi-target tracking by discrete-continuous energy minimization. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2054–2068.

Mittal, A. and Paragios, N. (2004). Motion-based background subtraction using adaptive kernel density estimation. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–302. IEEE.

Money, A. G. and Agius, H. (2008a). Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2):121–143.

Money, A. G. and Agius, H. (2008b). Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2):121–143.

Morris, R. J. and Hogg, D. C. (2000). Statistical models of object interaction. *International Journal of Computer Vision*, 37(2):209–215.

Okuma, K., Taleghani, A., De Freitas, N., Little, J. J., and Lowe, D. G. (2004). A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, pages 28–39. Springer.

Pirsiavash, H., Ramanan, D., and Fowlkes, C. C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1201–1208. IEEE.

Potapov, D., Douze, M., Harchaoui, Z., and Schmid, C. (2014). Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer.

Pritch, Y., Rav-Acha, A., and Peleg, S. (2008). Nonchronological video synopsis and indexing. *IEEE transactions on pattern analysis and machine intelligence*, 30(11):1971–1984.

Rav-Acha, A., Pritch, Y., and Peleg, S. (2006). Making a long video short: Dynamic video synopsis. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 435–441. IEEE.

Reid, D. (1979). An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854.

Shirazi, A. S., Funk, M., Pfleiderer, F., Glück, H., and Schmidt, A. (2012). Mediabrain: Annotating videos based on brain-computer interaction. In *Mensch & Computer*, pages 263–272.

Sigari, M. H. and Fathy, M. (2008). Real-time background modeling/subtraction using two-layer codebook model. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1.

Song, X., Cui, J., Zha, H., and Zhao, H. (2008). Vision-based multiple interacting targets tracking via on-line supervised learning. In *European Conference on Computer Vision*, pages 642–655. Springer.

Song, Y.-m. and Jeon, M. (2016). Online multiple object tracking with the hierarchically adopted gm-phd filter using motion and appearance. In *Consumer Electronics-Asia (ICCE-Asia), IEEE International Conference on*, pages 1–4. IEEE.

Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE.

Stauffer, C. and Grimson, W. E. L. (2000). Learning patterns of activity using real-time tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):747–757.

Stern, H. and Efros, B. (2002). Adaptive color space switching for face tracking in multi-colored lighting environments. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 249–254. IEEE.

Su, S.-T. and Chen, Y.-Y. (2008). Moving object segmentation using improved running gaussian average background model. In *Computing: Techniques and Applications, 2008. DICTA'08. Digital Image*, pages 24–31. IEEE.

Sun, I.-T., Hsu, S.-C., and Huang, C.-L. (2011). A hybrid codebook background model for background subtraction. In *2011 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 96–101. IEEE.

Sun, M., Farhadi, A., and Seitz, S. (2014). Ranking domain-specific highlights by analyzing edited videos. In *European conference on computer vision*, pages 787–802. Springer.

Syed, Y. A., Shetty, S., Wilkinson, N., and Brown, D. A modified codebook-based background subtraction technique to improve activity classification in highly variable environments.

Veltkamp, R., Burkhardt, H., and Kriegel, H.-P. (2013). *State-of-the-art in content-based image and video retrieval*, volume 22. Springer Science & Business Media.

Wang, D., Lu, H., and Yang, M.-H. (2013). Online object tracking with sparse prototypes. *IEEE transactions on image processing*, 22(1):314–325.

Wang, S. and Fowlkes, C. C. (2016). Learning optimal parameters for multi-target tracking with contextual interactions. *International Journal of Computer Vision*, pages 1–18.

Wren, C. R., Azarbayejani, A., Darrell, T., and Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):780–785.

Wu, S.-Y., Thawonmas, R., and Chen, K.-T. (2011). Video summarization via crowdsourcing. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pages 1531–1536. ACM.

Wu, Y., Lim, J., and Yang, M.-H. (2013). Online object tracking: A benchmark. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2411–2418. IEEE.

Xiao, J., Stolkin, R., and Leonardis, A. (2015). Single target tracking using adaptive clustered decision trees and dynamic multi-level appearance models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4978–4987.

Xu, J., Jiang, N., and Goto, S. (2011). Block-based codebook model with oriented-gradient feature for real-time foreground detection. In *Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*, pages 1–6. IEEE.

Yang, B. and Nevatia, R. (2012). Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1918–1925. IEEE.

Yang, M. and Jia, Y. (2016). Temporal dynamic appearance modeling for online multi-person tracking. *Computer Vision and Image Understanding*, 153:16–28.

Yoon, J. H., Yang, M.-H., Lim, J., and Yoon, K.-J. (2015). Bayesian multi-object tracking using motion context from multiple objects. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 33–40. IEEE.

Zen, G., de Juan, P., Song, Y., and Jaimes, A. (2016). Mouse activity as an indicator of interestingness in video. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 47–54. ACM.

Zhang, L., Li, Y., and Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Zivkovic, Z. (2004). Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31. IEEE.