N° d'ordre : 40317

Détection et suivi d'événements de surveillance

THÈSE

présentée et soutenue publiquement le 16 Juillet 2010

pour l'obtention du

Doctorat de l'Université des Sciences et Technologies de Lille
(spécialité informatique)

par

Md. Haidar SHARIF

Composition du jury

Président :	Sophie TISON, (Professeur)	Université de Lille I
Rapporteurs :	Zhongfei (Mark) ZHANG, (Professeur)	State University of New York (SUNY)
	Claude CHRISMENT, (Professeur)	Université de Toulouse III
Examinateurs :	Liming CHEN, (Professeur)	Ecole Centrale de Lyon
	Bernard GOSSELIN, (Professeur)	Université de Mons
Directeur de thèse :	Chabane DJERABA, (Professeur)	Université de Lille I

UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE
Laboratoire d'Informatique Fondamentale de Lille - UPRESA 8022
U.F.R. d'I.E.E.A. - Bât. M3 - 59655 VILLENEUVE D'ASCQ CEDEX
Tél. : +33 (0)3 28 77 85 41 - Télécopie : +33 (0)3 28 77 85 37 - email : direction@lifl.fr

Surveillance Event Detection and Monitoring

Dissertation

presented and publicly defended on the 16th July, 2010

for the achievement of

the Doctor of Philosophy (PhD) degree delivered by Université des Sciences et Technologies de Lille (specialization in Computer Science)

by

Md. Haidar SHARIF

Dissertation committee

Chair :	Prof. Sophie TISON	Université de Lille I
Reviewers :	Prof. Zhongfei (Mark) ZHANG	State University of New York (SUNY)
	Prof. Claude CHRISMENT	Université de Toulouse III
Examiners :	Prof. Liming CHEN	Ecole Centrale de Lyon
	Prof. Bernard GOSSELIN	Université de Mons
Advisor :	Prof. Chabane DJERABA	Université de Lille I

UNIVERSITÉ DES SCIENCES ET TECHNOLOGIES DE LILLE
Laboratoire d’Informatique Fondamentale de Lille - UPRESA 8022
U.F.R. d’I.E.E.A. - Bât. M3 - 59655 VILLENEUVE D’ASCQ CEDEX
Tél. : +33 (0)3 28 77 85 41 - Télécopie : +33 (0)3 28 77 85 37 - email : direction@lifl.fr

To my loving parents and exalted pedagogues

Résumé

Dans les systèmes de vidéosurveillance, les algorithmes de vision assistée par ordinateur ont joué un rôle crucial pour la détection d'événements liés à la sûreté et la sécurité publique. Par ailleurs, l'incapacité de ces systèmes à gérer plusieurs scènes de foule est une lacune bien connue. Dans cette thèse, nous avons développé des algorithmes adaptés à certaines difficultés rencontrées dans des séquences vidéo liées à des environnements de foule d'une ampleur significative comme les aéroports, les centres commerciaux, les rencontres sportives etc. Nous avons adopté différentes approches en effectuant d'abord une analyse globale du mouvement dans les régions d'intérêt de chaque image afin d'obtenir des informations sur les comportements multimodaux de la foule sous forme de structures spatio-temporelles complexes. Ces structures ont ensuite été utilisées pour détecter des événements de surveillance inhabituels au sein-même de la foule. Pour réaliser nos expériences, nous nous sommes principalement appuyés sur trois ensembles de données qui ont suscité notre réflexion. Les résultats reflètent à la fois la qualité et les défauts de ces approches. Nous avons également développé une distance pseudo-euclidienne. Pour démontrer son utilité, une méthodologie qui lui est propre a été utilisée pour la détection de plusieurs événements de surveillance standards issus de la base TRECVID2008. Certains résultats montrent la robustesse de cette méthodologie tandis que d'autres soulignent la difficulté du problème. Les principaux défis portent, entre autres, sur le flux massif de personnes, l'importance de l'occlusion, la réflexion, les ombres, les fluctuations, les variations de la taille de la cible, etc. Cependant, nos idées et nos expériences de ces problèmes d'ordre pratique ont été particulièrement utiles. De plus, cette thèse développe un algorithme permettant de suivre une cible individuelle dans le cadre de plusieurs scènes de foule. Les séquences vidéo de la base de PETS2009 Benchmark ont été prises en compte pour évaluer les performances de cet algorithme. Si on analyse ses avantages et ses inconvénients, celui-ci fait toujours preuve d'une grande exactitude et sensibilité vis-à-vis des effets de variation de la lumière, ce qui atteste de sa grande efficacité même lorsque la luminosité baisse, que la cible entre ou sort d'une zone d'ombre ou en cas de lueur soudaine.

Mots-clés : détection d'événement, entropie, matrice, distance, flux optique, point d'intérêt, distance pseudo-euclidienne, région d'intérêt, suivi.

Abstract

Computer vision algorithms have played a vital role in video surveillance systems to detect surveillance events for public safety and security. Even so, a common demerit among these systems is their unfitness to handle divers crowded scenes. In this thesis, we have developed algorithms which accommodate some of the challenges encountered in videos of crowded environments (e.g., airports, malls, sporting events) to a certain degree. We have adopted approaches by first performing a global-level motion analysis within each frame's region of interest that provides the knowledge of crowd's multi-modal behaviors in the form of complex spatiotemporal structures. These structures are then employed in the detection of unusual surveillance events occurred in the crowds. To conduct experiments, we have heavily relied on three thought-provoking datasets. The results reflect some unique global excellences of the approaches. We have also developed a *pseudo Euclidian distance*. To show its usage, a methodology based on it has been employed in the detection of various usual surveillance events from the *TRECVID2008*. Some results report the robustness of the methodology, while the rest gives evidence of the difficulty of the problem at hand. Big challenges include, but are not limited to, massive population flow, heavy occlusion, reflection, shadow, fluctuation, varying target sizes, etc. Notwithstanding, we have got much useful insights and experience to the practical problems. In addition, the thesis explores an individual target tracking algorithm within miscellaneous crowded scenes. Video sequences from the *PETS2009 Benchmark data* have been used to evaluate its performance. Viewing its pros and cons, the algorithm is still highly accurate and its sensitivity to the effects of diversity in noise and lighting, which ascertains its high-quality performance on disappearances, targets moving in and out of the shadow, and flashes of light.

Keywords: Event detection, Entropy, Matrix, Metric, Optical flow, Point of interest, Pseudo Euclidian Distance, Region of interest, Tracking.

Acknowledgments

I would like to take the opportunity to express my profound gratitude and sincere reverence to my advisor Prof. Chabane Djeraba for his guidance and relentless support. His commitment to hard work was a great source of inspiration over the last three years, and motivated me to always do my best. His worthy advice and scrupulous assistance made me more conversant in the research field of computer vision. I enjoyed and spent my valuable time by working in the demanding research environment provided by him. I would like to thank my first advisor in abroad, Dr.-habil. Christian Seidel, who welcomed me on the 10^{th} October 2002 at Max Planck Institute for Colloids and Interface, Golm, Germany to work on a project concerning computer architecture. Thereafter, by his continuous inspiration and instruction, I performed my MSc in Computer Engineering from Duisburg-Essen University. Needless to say, his valuable suggestion and recommendation helped me to start working with Prof. Chabane Djeraba.

The tedious work of reviewing this dissertation was performed by Prof. Zhongfei (Mark) Zhang and Prof. Claude Chrisment. I would like to give thanks for their valuable correction, suggestion, and acceptance. As well, thanks much the president, Prof. Sophie Tison, as well as the examiners, Prof. Liming Chen and Prof. Bernard Gosselin, of my dissertation committee.

I would like to give thanks all of my colleagues, with whom I worked on joint projects (e.g., MIAUCE, CAM4Home, etc.) and interacted on both scientific and non-scientific aspects, including, Nacim, Samir, Fred, Marius, Jean, Ismail, Ahmed, Emilie, Céline, Taner, Tarek, Thierry, Saja, Sahin, and Yassine. I would render thanks specially to Adel Lablack for his diversified both scientific and non-scientific suggestions and helps at the very beginning of this thesis. Likewise, thanks to all personages whose names I forgot to mention herewith.

Yet, this appreciativeness list will be uncompleted without acknowledging the ceaseless encouragement of my family members, to a vital degree, my loving parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Objective

The objective of this thesis is take some of the challenges in computer vision posed by interesting event/behavior detection and individual target tracking in diverse crowded video scenes obtained by surveillance video cameras. Both usual (normal) and unusual (abnormal) events detection in video surveillance is an important task for public safety in locations such as airports, malls, banks, subways, stations, town centers, hospitals, hotels, schools, concerts, cinema halls, parking places, sporting events, political events, and rallies. As huge amount of video data makes it an exhausting work for people to monitor and find events, an automatic system is badly needed for detecting specially suspicious events which would pose a potential threat. In spite of the concerted effort of computer vision research community, intelligent surveillance systems which process video feeds from real-world scenarios have not yet attained the desirable level of applicability and robustness. This is widely due to the algorithmic assumptions as well as the huge amount of video data analysis. This thesis develops approaches which address some of the critical aspects of handling surveillance event detection in crowded scenes. Fig.1.1 shows instances of crowded scenes containing objects of different modalities. Targeting at automatically detecting surveillance events should significantly improve the efficiency of video analysis, saving valuable human attention for only the most salient content for security and safety. It adopts several approaches and starts by performing a global-level *crowd-flow motions* analysis within each frame's region of interest that provides the knowledge of crowd's multi-modal behaviors in the form of complex spatiotemporal structures. These structures are then employed in the direction of detecting unusual surveillance events in the diverse crowded scenes. Upon analyzing the motion in another direction, this thesis develops a *pseudo Euclidian distance* and thereafter it employs a methodology based on the obtained distance for the direction of detecting various usual surveillance events. This thesis also explores an approach to tracking individual targets within the crowded scenes. The targets can be of a variety of types including but not limited to people, cars, etc. For instance, Fig.1.2 shows an example to track single targets in crowded scene. However, results of the approaches will be reported on diverse scenes to emphasize the generic nature of the techniques developed in this thesis.

1.2 Motivation

Safety and security have always been a major issue for shopping centers, banks, official buildings, enterprises, etc. Nowadays, nearly everybody looks for a way to keep its belongings safe and secure. Improvements in new technologies are making possible the development of many systems of safeguarding and surveillance for all necessities and budgets. Video surveillance is commonly used in security systems, but requires more intelligent and robust technical approaches. Such systems, which used in airports, subways, banks, concerts, cinema halls, sporting events, schools, supermarkets, parking places, hospitals, hotels, town centers or other private/public spaces, can bring security to a high level. The scientific challenge is to invent and implement automatic systems for obtaining detailed information about the activities and behaviors of people or vehicles observed by sensors (e.g., cameras). Automatic video surveillance is attractive because it promises to replace more costly option of staffing video surveillance

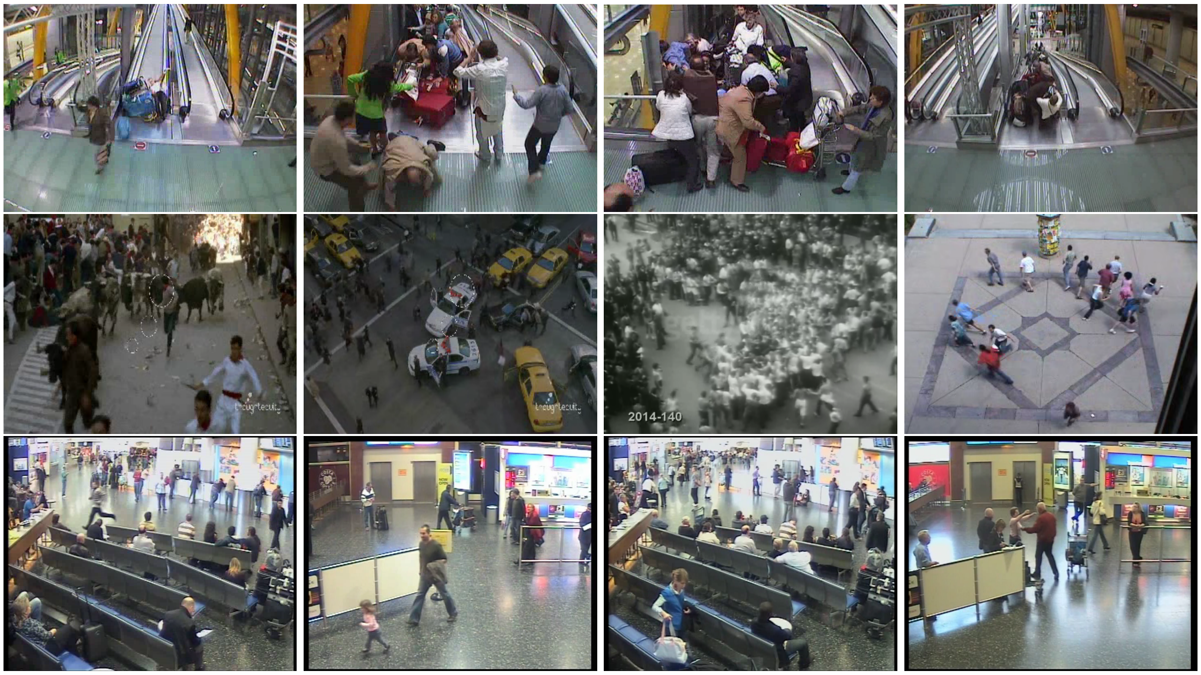


Figure 1.1: Some instances of crowded scenes containing objects/targets of different modalities.

monitors with human observers. Gathering of people at both public and private spaces pose significant challenges to public safety management officials from both normal and abnormal video event detection point of view. Often at event involving a gathering, people move through confined locations. A video event may be usual (normal) or unusual (abnormal), would vary greatly in duration, can be defined to be an observable action or change of state in a video stream that would be important for security management. Depending on the context, a usual event would be an unusual and contrariwise; e.g., suddenly a person starts running while others are walking or stops running while Marathon is running.

Unusual events are rare and occur infrequently and very hard to define. Fig.1.3 sketches a very simple definition. Unusual events include, but are not limited to, people fighting on the streets, people escaping from panic situations, person falling on the escalator, car violating the traffic rules on the high-way, etc. Some scenarios of abnormal situation are illustrated in the first and second rows of Fig.1.1. The first row in Fig.1.1 mainly concerns persons falling and rescuing the unfortunates who suffered from adverse circumstances on the escalator exit, whereas the second row depicts escaping in panics, fighting on the street, etc. It is quite obvious that to detect emergencies and provide useful

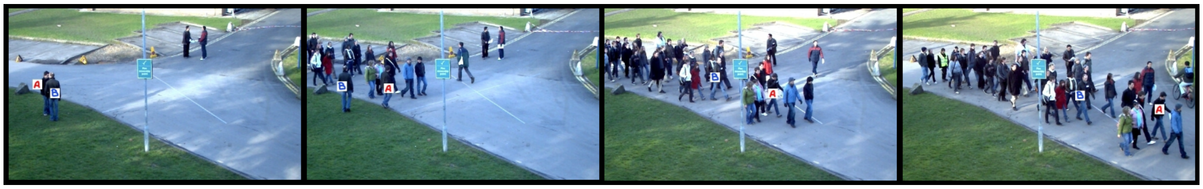


Figure 1.2: How can be tracked in dense crowd the individuals marked A and B in the video sequences?

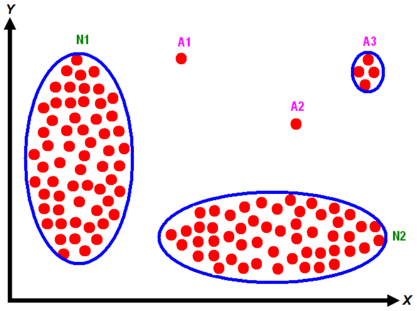


Figure 1.3: A simple example of usual and unusual behavior/event/situation. N_1 and N_2 are regions of *normal behavior*. Points A_1 and A_2 as well as points in region A_3 are *anomalies*, as their behavior or pattern differs from the usual e.g., N_1 and N_2. Explicitly, they are deemed as outliers in this context.

data from such videos is a daunting task normally due to the various behaviors of people as well as the shear number of people. There are many applications for vision systems that can detect emergencies and provide useful and informative surveillance. For example, escalators have become an accepted part of urban life. The United States Consumer Product Safety Commission estimates that there are approximately 7300 escalator-related injuries in the United States each year [39]. In 2000, the accident rate for escalator riding was about 0.815 accidents per million passenger trips through Taipei Metro Rapid Transit (MRT) heavy capacity stations [33]. Large-scale video surveillance of escalators would benefit from a system capable of recognizing perilous and inconsistent conditions and circumstances to make the system operators fully aware and attentive. More clearly, computer vision algorithms can make a significant contribution towards the security management.

Based on the context, e.g., *TRECVID2008* [43], usual events include, but are not limited to, person running, people meeting and splitting, opposing flow, embracing, object putting, etc. The third row in Fig. 1.1 displays few examples of such scenarios. Although these normal video events do not often show vital threat for the security management, it would still be enthusiastic for the security personnel to monitor the potential hazards if there would be certain information; e.g., a person may run suspiciously through a crowded public place, or may put an abandoned object in the long run, or may stroll in the opposite direction of the major flow of crowd, or may fight with another person, or may enter into the no entry zone. On analyzing and studying the behavior of the crowd through a vision system, it is possible to work in accordance with the public safety officials to ensure the safety of the public.

Individual target tracking in crowded video scenes is an eminent problem which arises in a wide variety of domains such as robotics, vehicular traffic, navigation, and communication systems. The main goal is to obtain a record of the trajectory of the moving targets over a space and time by processing sensor data. Reliable tracking methods are of crucial eminence in many surveillance systems to make possible human operators to remotely monitor activities across vast environments such as airports, railway transportation, maritime transportation, urban and highway road networks, banks, shopping malls, car parks, public buildings, industrial ambiences, military bases, prisons, strategic infrastructures, radar centers, and hospitals. For instance, a public and/or private space security personal watching the video-feed would be interested in tracking a few suspicious individuals within the crowd to keep an eye on their activities. In crowded situations it is quite common to lose track of target objects due to a severe occlusion arising from both the interaction of targets object with other members of the crowd and the structure of the scene. An example scenario of tracking individual targets in a dense crowd is demonstrated in Fig.1.2, where the individual targets marked **A** and **B** have to be tracked over crowded video sequences.

Still limited research efforts in this direction have been spent in building vision systems which can model various crowded scenes and provide useful information for public safety officials. One rational reason for the lack of these efforts in this direction is the complexity and challenges inherent in the problem.

1.3 Challenges

Successful techniques for handling a crowded visual scene will address a variety of problems, e.g.,:

- *Depiction of Abnormality*: An abnormal behavior or situation or event is difficult to be defined in a formal manner; a highly crowded scene often spreads with a speed, which makes it challenging to develop a general appreciation of abnormality by the gleaning information from the behavior of an individual. Also, a suspicious behavior in one scene can be regarded as normal in another ambiences. For instances, people running is abnormal if most of the crowd is walking or standing; people running in Marathon is normal but it is abnormal if a participating runner suddenly stops running while Marathon is running; a car moving in a different way than the most other traffic is also abnormal.

- *Few Pixels on Target*: In crowded situations, detection of an individual target becomes extremely hard when the number of pixels on the target decreases with the increasing density of the objects in the scene. The appearance information becomes further distorted due to the constant interaction among individuals making up the crowd (e.g., Fig. 1.2).

- *Appearance Ambiguity*: Ideally, one would like to track all the visible objects throughout the scene. Nevertheless, ambiguous appearance information resulting from too few pixels than recommendable on the target objects makes it arduous to persistently track the objects.

- *Selection of Good Features*: To detect surveillance events worthy of target/object scrutiny, it is necessary to represent video data in terms of features which allow us to reliably distinguish usual/unusual behaviors from the very ordinary occurrences. Feature or variable selection is the technique of selecting a subset of relevant features for building robust learning models. It serves two primary purposes. First, it makes training and applying a classifier more efficient by decreasing the size of the effective vocabulary; thus it minimizes the train expense. Second, it often increases classification accuracy by eliminating noisy features, which would increase the classification error. Consequently, a good feature selection method based on the number of features investigated for a sample classification is needed to speed up the processing rate and predictive accuracy, and to avoid incomprehensibility.

- *Handling of Occlusion*: Occlusion can result from the interactions of objects among themselves or the interactions of objects with the scene. Physical characteristics of a scene can act as sources of an occlusion resulting in the loss of observations of the target objects.

- *Effect of Lights and Shadows*: In outdoor installations illumination varies significantly, while in many indoor installations there may not be noticeable illumination variation but sometimes some light reflection can appear in the scene.

- *Estimation of Threshold*: To select the boundary, beyond which a radically different state-of-affair exists, is a crucial question in vision systems.

1.4 Nomenclature

Some terms which are engaged to depict the phenomenon related to crowded scenes in this thesis are used in some loose manner in the literature. To avoid confusion, some choice of words are:

- The term *segmentation* refers to the task of dividing a given crowded scene into dynamically distinct crowd regions or groupings.
- The term abnormal (unusual) *event*, *behavior*, and *situation* are used interchangeably, referring to a region of the scene where the behavior of the crowd is different from its usual (learnt) patterns.
- The term *context* refers to the contextual knowledge present in the crowded scene.
- The term *crowded scene* is used to refer to a video stream which contains a different density (low, moderate, and high) of objects.
- The term *motion history blob* (MHB) and *silhouetted region of motion component* (SRMC) are used interchangeably, referring to the silhouetted structure of more recently moving pixels of an object of interest.

1.5 Contributions

The contributions of this thesis is three folded as stated below.

- *First*, we have developed six algorithms for abnormal events detection.
- *Second*, we have proposed the *pseudo Euclidian distance* (PED). As an usage of the PED, we suggest a methodology for different kinds of usual *video event detection* (VED).
- *Finally*, an individual target tracking algorithm has been developed.

Unlike many traditional methods of processing a surveillance video, we start with computing a region of interest based on the main motion activities of the video frames which specially speed up the processing time. To detect abnormal events, we perform a frame based global motion analysis to generate a representation of each scene frame which captures the dynamics of the crowd on the frame basis. The global level analysis eliminates the need for low level change detection algorithms. To detect different normal video events, we have proposed a methodology based on PED. To track individual target, we have introduced a temporal-spatial domain algorithm which concerns the calculation of motion history blob regions and phase-correlation functions.

1.5.1 Detection of Unusual Video Events

We have investigated the conventional visual saliency, inspired by the fact that human attention may focus only on the most salient contents of the video, to single out abnormal or unusual events within streaming or archival videos. Saliency at a given location is determined primarily by how different this location is from its neighborhood in color, orientation, motion, depth, etc. However, the conventional saliency approach does not give us satisfactory results to analyze crowd behavior as a whole. Consequently, several algorithms have been developed in this thesis to perform crowd behavior analysis and to use those for the detection of abnormal events taking place in the crowd. The algorithms start with treating the spatial extent of a video where the main motion primarily exists as a form of *region of interest* (RoI) called *motion heat map* or *motion map* or *region of interest image map*. Points of interest (PoI) are defined within the RoI. Motion of PoI from one frame region to another is controlled by the optical flow. The significance of using the optical flow field to examine the temporal behavior of PoI is that the optical flow of a scene helps in revealing the characteristics of the scene as optical flow patterns vary in time inline with the crowd multi-modal behaviors. To distinguish any kind of crowd events/behaviors/situations either usual or unusual, it is very important to analyze this complex spatiotemporal structures exhibited by a moving crowd.

Spatiotemporal information takes into account motion as an informative feature to detect and segment interesting objects or targets by means of optical flow computation, block matching or other motion detection methods. On analyzing the complex spatiotemporal information in various ways, we have proposed the following approaches:

- Covariance Matrix 3.2([SID08a]),
- Normalized Continuous Rank Increase Measure 3.3([SD09a]),
- Mahalanobis Metric 3.4 ([SD09c]),
- Bhattacharyya Metric 3.5([SD10a]),
- Enumerated Entropy 3.6 ([SID08b],[SID10]),
- Shannon Entropy 3.7 ([SDb])

which have been annexed on the existing directional start-of-the-art. All the approaches have been tested on surveillance videos obtained by single cameras. To conduct experiments, we have heavily relied on the *Escalator dataset* [132], *UMN dataset* [137], and *Web dataset* [131]. Since the approaches are based on the analysis of the spatiotemporal information, there are some unique global excellences and breakages reflect on them.

1.5.2 Detection of Usual Video Events

In this direction of researches, we have lined up our efforts and successfully annexed the following contributions ([SD09b]):

- *First*, we developed a new method which generates automatically the *pseudo Euclidian distance* (PED) from the trigonometrically treatment of the *motion history blob* (MHB).
- *Second*, we proposed a methodology based on PED for various kinds of *video event detection* (VED), e.g., PersonRuns, OpposingFlow, PeopleMeet, Embrace, PeopleSplitUp, and ObjectPut.

The developed algorithm generates automatically PED from the trigonometrical treatments of *motion history blob* (MHB) obtained from *motion history image* (MHI). Given a point with its direction of motion where the point coincides the center of a circle. *How far the point can virtually travel inside the circle with that direction?* That virtual distance is called *pseudo Euclidian distance* (PED). The idea of PED remains one of the important contributions of this thesis and would be used in a wider variety of computer vision applications. If we use PED for normal VED, it is important to know the explicit information of motion history blobs which can be gained by tracking objects of interest and thereof can get more information that will be used for detecting specific video events. The results based on the detection of several events at *TRECVID2008* [43] in real videos have been exhibited. Some results show the robustness of the methodology, while the rest gives evidence of the difficulty of the problem at hand.

1.5.3 Individual Target Tracking in Crowded Scenes

We have also engaged many of our efforts in this research and contributed as the following.

- *First*, we have studied an extended method ([SMD08b]), which was originally proposed in [166]. The method is based on the spatial information. It follows the detection of a target in a video and the target is to be tracked in the subsequent frames using the *region covariance matric* method introduced in [166]. The method works in some extent as a single covariance matrix extracted from a region of interest matching the region in some other views and poses.

- *Second*, we have proposed an approach ([SDc]) based on *temporal-spatial* information suitable for tacking individual targets in a sparse crowd, medium density crowd, and dense crowd.

The approach is based on the estimation of a target (region of interest over frame in time) and candidate (region of possible target over next frame in time) regions from the silhouetted structures of more recently moving pixels of the object of interest by combining two techniques, namely *motion history image* MHI [22] and *Hu's moments* [80]. The MHI function, which uses temporal history of the position or motion, helps to create a *silhouetted region of motion component* (SRMC) while Hu's moments find the *center of mass* or *center of gravity* or *centroid* of each SRMC. A great advantage behind of this hybrid technique is that it is not necessary to search the possible target region everywhere in the candidate frame except for the candidate regions. Consequently, the searching process becomes extremely rapid. The target region and its most representative candidate region in the next frame normally give a distinct phase-correlation sharp peak as compared to the individual peaks between them. The height of the peak gives a good similarity measure for image matching which is good for tracking targets one by one in different crowded scenes. The video sequences of the *PETS2009 Benchmark data* [42] have been considered for performance evaluation of the approach. Deeming the favorable and the unfavorable factors of the proposed approach, it is still highly accurate with respect to the effects of mutations in noise and lighting, which assures high-quality performance on fades, targets moving in and out of the shade, and flashes of light.

1.6 Organization of the Thesis

The structure of the thesis is as follows. **Chapter 2** reviews the existing literature that focuses on handling different aspects of crowded scenes. **Chapter 3** presents the detailed abnormal video event detection frameworks. Results are shown for very challenging sequences gathered from a variety of resources. **Chapter 4** develops the algorithm of the PED which is used for normal video events detection. **Chapter 5** introduces the tracking algorithm that is specifically designed for tracking individuals in crowded scenes. The thesis is concluded in **Chapter 6** with a brief statement that presents the main points of the contributions in a concise form and some clues for a further investigation.

Chapter 2

Literature Review

Contents

2.1 Overview

In this chapter, we review the approaches which have been developed to handle aspects of crowd behavior analysis and target tracking. We have divided the chapter into two primary parts. The first part covers the algorithms and techniques which are used for crowd flow and behavior analysis in crowded scenes. The second part gives a recent overview of the target tracking approaches.

2.2 Crowd flow and behavior analysis

We make a classificatory division of the works of crowd flow and behavior analysis: first division relates to person count and density estimation, the second division concerns abnormal event detection in mob flows, and the third division depicts specific video event. A common view of all those methods is that they make interesting analysis for crowd surveillance, nevertheless they do not detect abnormal situations/events/behaviors.

2.2.1 Estimation of Crowd Density

In computer vision, person count and density estimation is gaining popularity for the sake of security and surveillance. The estimation of number or density of people in an area under surveillance is very important for the problem of crowd monitoring. To address the crowd density estimation problem, the initial research efforts ([40, 44, 35, 155, 34, 128, 129]) can be found in the early to late nineties. Global image features e.g., foreground pixels, textures, edges, optical flows, etc., were often put to use in this assembly of works. More recently, there are some crowd density estimation and counting works based on wavelet features and trajectory information. We have subclassified the works of crowd density estimation.

2.2.1.1 Pixel-based techniques

In [40], crowd density was estimate by extracting a set of features which included a number of edge points, a number of maxima in the edge point histogram, and the sum of the amplitudes of the maxima in the edge point histogram. Authors in [44] estimated the number of foreground pixels or number of edge pixels from the image and used them in a linear regression framework to estimate the number of people in the scene. In [35], the sizes of foreground regions and ratio of foreground to background regions as features were used to train a fuzzy classifier that classified the scene into one of five categories: no people, a few people, some people, many people overcrowding. In the same vein, in [155, 34], neural networks were trained to classify the level of a crowd. Crowd estimation using color density have been presented by [4]. However, these pixel-based techniques were simple and fast, but are not reliable when the crowd density is high.

2.2.1.2 Texture-based techniques

There are some texture-based works [128, 129] which used crowd images of different densities as different texture patterns, and estimated the crowd density by texture analysis schemes. Texture measures were extracted, in [128], from the images through gray level dependence matrices, straight line segments, Fourier analysis, and fractal dimensions. Crowd density estimations were given in terms of the classification of the input images into densities of very low, low, moderate, high, and very high. In a later work of [129], this method was extended where the Minkowski fractal dimension was used for density estimation. An automatic method of estimating crowd density using texture analysis and machine learning has been presented by [113].

2.2.1.3 Wavelet-based techniques

According to the definition of wavelet, a wavelet is an oscillating and attenuated function and its integrals equal to zero. It is a mathematical function useful in digital signal and image processing. However, the utility of wavelet features for density estimation was explored in the work of [116, 179]. In [116], the Haar wavelet transform (HWT) is used to extract the featured area of the head-like contour, afterwards support vector machine is used to classify these featured area as the contour of a head or not. At the end, perspective transforming technique is used to estimate crowd size. Authors in [179] addressed an algorithm to estimate the crowd density based on the combination of multi-scale analysis and a support vector machine. Using wavelet transform, the crowd image transforms into multi-scale formats. Statistical features at each scale of the transformed images are then extracted as density character vectors. A classifier based on a support vector machine is designed to classify the extracted density character vectors into different density levels.

2.2.1.4 Trajectory-based techniques

A trajectory is the path a moving object or target follows through frames. In [146] counted the number of people by segmenting the moving objects in a dense crowded scene. Counting is typically performed by clustering a set of extended tracked features where spatio-temporal conditioning was used to overcome the fragmented nature of the tracks. Authors in [10] addressed a trajectory clustering outline for crowd counting. They used some representations (e.g., independent component analysis, time series, maximum of cross correlation) and compared different distance/similarity measures (e.g., Euclidian, longest common subsequence, Hausdroff) under a common hierarchical clustering framework. Length clustering, spatial clustering, and pedestrian counting are the stages of the hierarchy.

2.2.2 Detection of Unusual Events

Events detection is a classical task in computer vision. From a surveillance point of view, it is specially important to detect unusual events and hence a wide variety of different approaches covering diverse applications have been proposed. We have discussed some works which is based on the approach that consists of modeling normal behaviors, then estimating the deviant behavior or attitudes between the normal behavior model and the observed behaviors. Those deviations are labeled as abnormal. The principle of the general approach is to exploit the fact that data of normal behaviors are generally available, and data of abnormal behaviors are ordinarily less available. Consequently, the deviations from examples of normal behavior are used to characterize abnormality. The major existing methods for abnormal behavior understanding are outlined as follows:

2.2.2.1 Trajectory major approaches

One of the earliest approaches to behavior classification, proposed by [91], identified unusual behavior by comparing new trajectories with a set of clusters representing typical sequences of typical local motion vectors in a given scene. A similar approach has been adopted by [62], where typical trajectories are modelled with a more complex hierarchical clustering strategy. In the algorithm of [62] for learning motion patterns, trajectories are clustered hierarchically using spatial and temporal information and then each motion pattern is represented with a chain of Gaussian distributions. Based on the learned statistical motion patterns, statistical methods are used to detect anomalies and predict behaviors. Experimental results of anomaly detection in the real traffic scene (mainly cars) and indoor model scene have been reported. In a different vein, the work of [47] has shown that unusual trajectories can also be identified using a rule-based approach, inspired by cognitive science, which quantifies the extent to which the movements of a given individual could be regarded as goal-directed. A spatial model to represent the routes in an image has been developed in [127]. One short coming of this method is that solely spatial information is used for trajectory clustering and behavior recognition. The system cannot differentiate between a person walking and a person lingering around, or between a running and a walking person. On the other hand, a method for detecting nonconforming trajectories of objects has been proposed in [92]. Authors in [162] detected events which have never occurred or occur so rarely that they are not represented in the clustered activities. The method includes robust tracking, based on probabilistic method for background subtraction. This system has been used to track people (from low crowded scenes) in indoor environments, people and cars in outdoor environments, fish in a tank, ants on a floor, and remote control vehicles in a lab setting. But the robust tracking method is not adapted to crowd scene, in which it is too complex to track objects. In [124], a fuzzy support vector machine based algorithm to detect the abnormal trajectory patterns of moving objects from surveillance video has been proposed. However, the algorithm does not consider the analysis of multi trajectories.

2.2.2.2 Statistical model & classifier major approaches

A hidden Markov model (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved state. An HMM can be considered as the simplest dynamic Bayesian network. A *Markov random field* (MRF) is similar to a Bayesian network in its representation of dependencies. A Markov random field, sometimes called as *Markov network* or *undirected graphical model*, is a graphical model in which a set of random variables have a Markov property described by an undirected graph. It can show certain dependencies that a Bayesian network cannot (e.g., cyclic dependencies); but then again, it cannot represent certain dependencies that a Bayesian network can (e.g., induced dependencies). The prototypical Markov random field is the *Ising model*; indeed, the Markov random field was introduced as the general setting for the Ising model [103].

Automatic behavior analysis can consist of decomposing video data in terms of some low level representational primitive, and modeling the sequential topology of behaviors in terms of such primitives. The low-level representational currencies which have been employed for the global representations of changes in scene content employed by [177, 178]. Sequences of such low-level primitives are typically represented using Hidden Markov Models (and variants thereof) or Bayesian Networks, which provide a powerful probabilistic framework for identifying anomalous behavior. A location-based approach for behavior modeling and abnormality detection has been addressed by [16]. The spatial and temporal dependencies between motion labels obtained with simple background subtraction. A Markov random field model is parameterized by a co-occurrence matrix, which contains the average behavior observed in a training sequence. Abnormal events can be detected by detecting traces which significantly differ from the normal model following a likelihood ratio test. The method was tested on various challenging outdoor videos, which primarily present few people.

Authors in [52] proposed probabilistic models corresponding to behavior clusters, and use these models to perform abnormal behavior detection. The method was tested by a highway video. For recognizing rare events in aerial video, authors in [32] used hidden Markov models to represent the spatiotemporal relations between objects and uncertainty in observations, where the data observables are semantic spatial primitives encoded based on prior knowledge about the events of interest. The effectiveness of the approach was demonstrated by using real aerial video and simulated data. In [189], a semi-supervised adapted HMM framework has been proposed, in which usual event models are first learned from a large amount of (commonly available) training data, while unusual event models are learned by Bayesian adaptation in an unsupervised manner. The framework is good for cases in which collecting sufficient unusual event training data is impractical and unusual events cannot be defined in advance. Experiments on audio, visual, and audiovisual data streams illustrate the effectiveness of the framework, but there is no golden hints how the approach would be handled in either few people or crowd situation. For detecting abnormalities in surveillance video, author in [89] proposed a multi-sample-based similarity measure, where HMM training and distance measuring are based on multiple samples. These multiple training data are acquired by a dynamic hierarchical clustering method. Experimental results have been reported on real surveillance video with few persons. The approach of [9] presented a set of theoretical and practical tools for the domain of behavior recognition, which have been integrated within a unified, automatic, bottom-up system based on the use of multiple cameras performing human behavior recognition in an indoor environment, without a uniform background. Their methodology classifies behavior as normal or abnormal, by treating short-term behavior classification and trajectory classification as two different classification problems. A support vector machine and a continuous HMM treated as an one-class classifier. However, the methodology would not be interesting for crowded scene.

By carrying out change detection and congestion estimation, an MRF-based based approach for real-time subway monitoring has been proposed by [141]. Their solution consisted of two steps. The first step was a change detection algorithm that distinguished the background from the foreground by using a discontinuity preserving MRF-based approach. In the MRF model, information from different sources (background subtraction, intensity modeling) was combined with spatial constraints to provide a smooth motion detection map. In the second step, the obtained change detection map was combined with a geometry module to perform a soft auto-calibration to estimate a measure of congestion of the observed area (platform). A space-time MRF model to detect abnormal activities in video has been proposed by [101]. The nodes in the MRF graph correspond to a grid of local regions in the video frames, and neighboring nodes in both space and time are associated with links. To learn normal patterns of activity at each local node, the distribution of its typical optical flow with a mixture of probabilistic principal component analyzers is captured. For any new optical flow patterns detected in incoming video clips, the learned model and MRF graph to compute a maximum a posteriori estimate of the degree of normality at each local node is used. The method would arouse interest for few people scene but would not be interesting for crowded scene.

Authors in [178] addressed the problem of modeling video behavior captured in surveillance videos for the applications of online normal behavior recognition and anomaly detection without any manual labeling of the training data set. The similarity between behavior patterns are measured based on modeling each pattern using a Dynamic Bayesian Network. The natural grouping of behavior patterns is discovered through a novel spectral clustering algorithm with unsupervised model selection and feature selection on the eigenvectors of a normalized affinity matrix. A composite generative behavior model is constructed from a small training set to accommodate variations in unseen normal behavior patterns. Finally, a runtime accumulative anomaly measure detects abnormal behavior, whereas normal behavior patterns are recognized when sufficient visual evidence has become available based on an online likelihood ratio test method. The effectiveness and robustness of the approach has been tested on data sets collected from both indoor and outdoor surveillance scenarios. Nevertheless, human behavior was monitored with one or few people at a time. Also there is no suggestions how the approach would be handled in either some people or crowd situations.

The problem of detecting irregularities in visual data, e.g., detecting suspicious behaviors in video sequences, or identifying salient patterns in images has been addressed by [24]. The method formulates the problem of detecting regularities and irregularities as the problem of composing (explaining) the new observed visual data (an image or a video sequence, referred to below as *query*) using spatiotemporal patches extracted from previous visual examples (the *database*). Regions in the query which can be composed using large contiguous chunks of data from the example database are considered likely. The larger those regions are, the greater the likelihood is. Regions in the query which cannot be composed from the example database (or can be composed, but only using small fragmented pieces) are regarded as unlikely/suspicious. Concisely, their method is posed as an inference process in a probabilistic graphical model and would be arousing the attention for low crowded scenes but needs learning process and/or training data. An algorithm which is based on multiple local monitors that collect low-level statistics has been proposed by [1]. Each local monitor produces an alert if its current measurement is unusual and these alerts are integrated to a final decision regarding the existence of an unusual event.

An approach to detect unusual events in terms of velocity and acceleration has been proposed by [86]. The moving objects in the scene are detected and tracked. A supervised support vector machine method is used to train the system with one or more typical sequences, and the resulting model is then used for testing the proposed method with other typical sequences (different scenes and scenarios). Experiments were carried out on two outdoor and two indoor video sequences. The algorithm proffered by [59] shown the capabilities of their proposed system for analyzing complex threat detection scenarios in thermal imaging e.g., detection of people lying down in a crowded environment. Their method was based on the detection and segmentation of individuals within groups of people using a combination of several weak classifiers in a boosting algorithm.

2.2.2.3 Optical flow major approaches

Optical flow or optic flow means tracking specific features (points of interest) in an image across multiple frames. It is the pattern of apparent motion of objects, surfaces, and edges in a visual scene caused by the relative motion between an observer (an eye or a camera) and the scene [29, 173]. In [23] proposed to model the scene dynamics under consideration for the prevention of crowd related emergencies in large crowds. They started by estimating the optical flow and clustered the optical flow vectors based on direction and magnitude to segment different crowds. Afterwards, they detected a number of events using a technique based on Hough voting space. The types of event detected by their method include circular flow paths close to site exits indicating trapped crowds; crowd flow diverging from a point to all directions, which might indicate a potential danger (e.g., fights, fire, etc.). Obstacles in the flow paths that might correspond to injured pedestrians or deliberate flow disturbances. Afterwards, in the work of [121], an automatic monitoring system was proposed for detecting overcrowding conditions in the platforms of underground train services. Using a block matching scheme, optical flow, and filtering, authors in [25] proposed an algorithm for the detection of abnormal individual or crowd motion in subway corridors. The underlying assumption of their algorithm was that for detection of any abnormal activity the knowledge about direction of crowd motion is essential. The optical flow vectors were filtered and then used for the construction of motion trajectories, which were finally used to detect counter flows in one way corridors.

To detect emergency or abnormal events in the crowd, authors in [7, 8] encoded optical flow features with HMMs. The crowd behavior was characterized at a global level by using the optical flow of the video sequence. During the learning stage, a reduced order representation of the optical flow was generated by performing principal component analysis on the flow vectors. The top few eigenvectors were used as the representative features and spectral clustering was performed to identify the number of distinct motion patterns present in the video. The features in the clustered motion segments were used to train different HMMs which were later used for event detection in crowds. The methods would be worth interesting but were not experimented on the real video data.

The mathematical framework, introduced by [5], used Lagrangian particle dynamics to detect the flow instabilities from video streams characterized by extremely high crowd density e.g., marathon, political rally, thousands of people circling around the Kabba in an anticlockwise direction, etc. They obtained one mean field by calculating optical flow fields over several number (experimented 5 to 10) of video frames. Several number (experimented 5 to 10) of mean fields are stacked together to obtain one block of mean fields. A typical size of the block is 16×16. Their framework is very suitable to detect flow instabilities from the events where thousands of people primarily present like religious festivals, parades, concerts, football matches, etc. However, the method would not be so interest-bearing in the context of a crowd scene like airport, shopping malls, and so forth to detect abnormalities; more precisely, in case the of escalators or narrow passages where people mainly go in one direction and the density of people is never so high. Authors in [83] presented an approach to detect abnormal situations in crowded scenes by analyzing the motion aspect instead of tracking subjects one by one.

Authors in [131] introduced a method by capturing the dynamics of the crowd behavior to detect and localize abnormal behaviors in crowd videos using social force model. In this model an individual is subject to long-ranged forces and the dynamics follow the equation of motion, similar to Newtonian mechanics. A grid of particles is placed over the image and it is advected with the space-time average of optical flow. By treating the moving particles as individuals, their interaction forces are estimated using social force model. The interaction force is then mapped into the image plane to obtain force flow for every pixel in every frame. Randomly selected spatiotemporal volumes of force flow are used to model the normal behavior of the crowd. By using a bag of words approach normal and abnormal frames are classified. The regions of anomalies in the abnormal frames are localized using interaction forces. The experiments were conducted on several challenging datasets. The results are interesting except several number of false alarms.

2.2.2.4 Discussion

The brief overview of the research literature underscores the fact that most of the aforementioned approaches require a learning period to estimate various parameters of the system, and consequently, reliable learning of unknown parameters is not always accurately possible. We have proposed several algorithms, fall into the category of optical flow major approaches, to detect abnormal behaviors/events in crowd videos without segmentation or tracking subject singly. Our proposed algorithms have several important differences from these body of works as: (i) they expect region of interest; (ii) they detect all events in videos where entropy variations are important as compared to previous events; (iii) they work all directional flow of movers without imposing a restriction of their numbers in the videos; (iv) they do not expect efficient learning process and training data but would look for a prior cut-off.

2.2.3 Detection of Usual Events

Action recognition in crowded environment is an important and challenging topic in computer vision, with many important applications including video surveillance, automated cinematography and understanding of social interaction. Analysis of crowd behaviors is an important problem. It can be dealt with at the individual level where the event of interest is defined in terms of individual objects, or it can be defined at a global level where the behavior of the crowd is modeled at an extended spatial scale. The analysis of the global level behavior is often carried out by using the motion information. Many promising strategies have been identified which can be directly or indirectly employed for events detection. We have broadly categorized into approaches based on tracking, spatiotemporal sequence, spatiotemporal volume, and spatiotemporal point.

2.2.3.1 Tracking-based methods

Tracking-based approaches can incorporate existing domain knowledge about the target event in the model and the system can support online queries since the video is processed a single frame at a time. Initializing tracking models can be difficult, particularly when the scene contains distracting objects.

Although the work of [148] has demonstrated significant progress in cluttered environments, tracking remains challenging in such environments, and the tracker output tends to be noisy. An alternate approach to tracking-based event detection focuses on multi-agent activities, where each actor is tracked as a blob and activities are classified based on observed locations and spatial interactions between blobs [60, 78]. These models are well-suited for expressing activities such as loitering, meeting, arrival and departure, etc. The algorithm proposed by [147] works by two steps: (i) tracking people in 2D and then, using an annotated motion capture dataset; (ii) synthesizing an annotated 3D motion sequence matching the 2D tracks. The 3D motion capture data is manually annotated off-line using a class structure that describes everyday motions and allows motion annotations to be composed. They showed smoothed annotation results for a sequence of jumping jacks (sometimes known as star jumps) from two such annotation systems. Detection of events (e.g., PeopleMeet, personRuns, PeopleSplitUp, etc.) in the surveillance video selected for *TRECVID 2008*[43] is an extremely difficult task. For example, people meeting event usually happens in complex and crowded environments, where many pedestrians, moving in different directions are present simultaneously. Also many pedestrians have similar appearances and occlude each other and occlusions by other scene objects are also common. In the work of [109] people meeting event was detected mainly by analyzing pedestrian trajectories. They detected and tracked people in the scene by using the method described in [81]. To get reliable pedestrian trajectories for people meeting event detection task, they proposed a detection-based hierarchical association method that was capable of robustly tracking multiple pedestrians under such challenging conditions. Their method generated pedestrian trajectories by means of progressively associating detection responses given by the pedestrian detector as introduced by [176]. A combination of trajectory and domain knowledge based subsystems can be found in [55]. The trajectory-based subsystem implements human detection and tracking to generate trajectory and three-level trajectory features are used to detect PersonRuns, PeopleMeet, PeopleSpiltUp, and Embrace. The domain knowledge-based subsystem constructs specific models for PeopleMeet, Opposingflow, and ElevatorNoEntry depending on domain knowledge.

2.2.3.2 Spatiotemporal sequence-based methods

These kind of methods for event detection operate directly on the spatiotemporal sequences, attempting to recognize the specified pattern by brute-force correlation without segmentation. Authors in [51] correlated flow templates with videos to recognize actions at a distance. In [98] trained a cascade of boosted classifiers to process the vertical and horizontal components of flow in a video sequence. An algorithm for correlating spatiotemporal event templates against videos without explicitly computing the optical flow has been introduced by [159]. Their approach can detect very complex behaviors in video sequences (e.g., ballet movements, pool dives, running water), even when multiple complex activities occur simultaneously within the field-of-view of the camera. However, the approach can be noisy on object boundaries. On the other hand, method for detecting events in crowded videos has been proposed by [99]. The video is treated as a spatiotemporal volume and events are detected using a volumetric shape descriptor in combination with flow descriptor of [159]. Their approach detected events in difficult situations containing highly-cluttered dynamic backgrounds, and significantly outperforms the baseline method of [159]. Experimental results e.g., picking up a dropped object or waving in a crowd have been showed. Yet, the biggest limitation of this work is that the model is derived from a single exemplar of the event, thus limiting the ability to generalize across observed event variations.

2.2.3.3 Spatiotemporal volume-based methods

These type of methods treat the spatiotemporal volume of a video sequence as a 3D object. Different events in video generate distinctive shapes, and the goal of such methods is to recognize an event by recognizing its shape. Human action in video sequences can be seen as silhouettes of a moving torso and protruding limbs undergoing articulated motion. Authors in [72] regarded human actions as three-dimensional shapes induced by the silhouettes in the space-time volume. They adopted the approach of [56] for analyzing 2D shapes and generalize it to deal with volumetric space-time action shapes. Their method utilized properties of the solution to the Poisson equation to extract space-time features such as local space-time saliency, action dynamics, shape structure, and orientation. They showed that these features are useful for action recognition, detection, and clustering. A new view-based approach to the representation and recognition of human movement has been presented by [22]. The basis of the representation is a temporal template, which is a static vector-image where the vector value at each point is a function of the motion properties at the corresponding spatial location in an image sequence. For computational efficiency and greater robustness to action variations, authors in [22] projected the spatiotemporal volume down to motion-history images, which authors in [175] extended to motion-history volumes. In the work of [175] introduced Motion History Volumes (MHV) as a free-viewpoint representation for human actions in the case of multiple calibrated, and background-subtracted, video cameras. They presented algorithms for computing, aligning and comparing MHVs of different actions performed by different people in a variety of viewpoints. Alignment and comparisons were performed efficiently using Fourier transforms in cylindrical coordinates around the vertical axis. Results indicated that their representation can be used to learn and recognize basic human action classes, independently of gender, body size and viewpoint.

A novel representation for actions using spatiotemporal action volumes has been proposed by [187]. When the object performs an action in 3D, the points on the outer boundary of the object are projected as 2D (x, y) contour in the image plane. A sequence of such 2D contours with respect to time generates a *spatiotemporal volume* (STV) in (x, y, t), which can be treated as 3D object in the (x, y, t) space. The STV is analyzed by using the differential geometric surface properties to identify action descriptors capturing both spatial and temporal properties. Several experimental results using video sequences including dancing, falling, tennis strokes, walking, running, kicking, sit-down, stand-up, surrender, hands-down, aerobics actions were presented. The hypothesis of [181] is that any instance of an action can be expressed as a linear combination of spatiotemporal action basis, capturing different personal styles of execution of an action, different sizes and shapes of people, and different rates of execution. Based on this hypothesis, they have developed a framework for learning the variability in the execution of human actions that is unaffected by the changes. Their test data included the actions e.g., sitting, standing, falling, walking, dancing, running, etc.

2.2.3.4 Spatiotemporal interest point-based methods

Space-time interest points [108] have become popular in the action recognition community. In [156], constructed video representations in terms of local space-time features and integrate such representations with SVM classification schemes for recognition. For the purpose of evaluation, they used a new video database containing 2391 sequences of six human actions performed by 25 people in four different scenarios. Authors in [48] showed that the direct 3D counterparts to commonly used 2D interest point detectors are inadequate, and hence they proposed an alternative. Anchoring off of the interest points, they devised a recognition algorithm based on spatiotemporally windowed data. The recognition results were presented on a variety of datasets including both human and rodent behavior. An unsupervised learning method for human action categories presented by [87]. A video sequence is represented as a collection of spatial-temporal words by extracting space-time interest points. The algorithm automatically learns the probability distributions of the spatial-temporal words and the intermediate topics corresponding to human action categories. This is achieved by using latent topic models such as the *probabilistic latent semantic analysis* (pLSA) model and *latent Dirichlet allocation* (LDA). Experiments were conducted on three datasets. From a new video sequence, the algorithm can categorize and localize the human action(s) contained in the video. For event detection in crowd scenes, authors in [14] used a global analysis of motion vectors, obtained from optical flow techniques. Authors in [15] presented an effective method for human action recognition using statistical models based on optical flow orientations. The event detection algorithm of [53] found interest points, clustered them into visual keywords and used a classifier to detect activities based on trained SVM models. In [61] used optical flow features and SVM to detect surveillance events. To detect events (e.g., PersonRuns, OpposingFlow, etc.), the optical flow was used for analyzing the motion of objects in [97]. To detect single surveillance events, authors in [58] proposed systems which relied on low level vision properties such as optical flow and image intensity as well as heuristics based on a given event and context.

2.2.3.5 Discussion

A large body of the aforementioned works have addressed that vast diversity of one event viewed from different view angles, different scales, different degrees of partial occlusion, few pixels on targets, etc., make challenge for performance of the event detectors. Therefore, it is necessary to greatly improve their effectiveness by further investigation. We have developed a new method which generates automatically *pseudo Euclidian distance* (PED) from the trigonometrically treatment of the *motion history blob*. Then we have proposed a methodology, falls into the category of tracking-based methods, based on PED for various kinds of *video event detection* (VED), e.g., PersonRuns, OpposingFlow, PeopleMeet, Embrace, PeopleSplitUp, ObjectPut, etc. Some results show the robustness of the methodology, while the rests give evidence the dimension of the difficulty of the problem at hand.

2.3 Target Tracking in Crowds

Detection of individuals in dense crowds is one of the challenging research topics in computer vision because it is at once a problem of segmentation, recognition, and tracking. Most tracking algorithms proposed on the common problem of tracking, without specifically addressing the challenges of a crowded scene. We have reviewed the tracking approaches which are particularly designed for crowded situations. Readers interested in a detailed review of the state of the art in tracking are referred to a survey by [185]. Herewith, tracking has been roughly categorized based on low and high levels information.

2.3.1 Low-level information based tracking

The *Markov chain Monte Carlo* (MCMC) methods, are a class of algorithms for sampling from probability distributions based on constructing a Markov chain that has the desired distribution as its equilibrium distribution. The state of the chain after a large number of steps is then used as a sample from the desired distribution. The quality of the sample improves as a function of the number of steps. Authors in [100] used a MCMC based particle filter to deal with interactions among targets in crowded scenario. The interactions among the targets were shaped by a Markov Random Field based motion priors which were learnt on the fly using an MCMC sampling. The results were reported on videos of interacting insects. Authors in [190] used the initial detection of people in crowds to initialize the ellipsoid based human shape models and color histograms to accomplish tracking. Analogously, in [27], features points were tracked and clustered over time. Finally, a separate trajectory for each single was generated.

There are sundry interesting and relevant body of works which put to the test to track e.g., sparse crowds of ants [100], hockey players [30], crowds of densely packed people [70, 117, 118], a dense flock of bats [57], and biological cells [111]. A multi-target tracking approach for tracking hockey players in a video was presented by [30]. The approach consisted of a modified particle filtering algorithm where they introduced a global nearest neighbor data association algorithm for assigning Ada-boost based detections to existing tracks for the proposal distribution. Mean-shift algorithm was embedded into the particle filter framework to stabilize the trajectories of the targets for robust tracking during mutual occlusions. Using their approach, they were able to track multiple targets and correctly maintain identities in the presence of background clutter, camera motions and mutual occlusion between targets. In the work of [70], groups were defined on the basis of the position and velocity of targets. They used a set of merging and splitting rules which were embedded into a Kalman filtering framework for tracking multiple groups. The target of their work at scenarios where large number of targets form natural groups that can be efficiently tracked together. The *near regular texture* (NRT) was used for tracking groups of people in [117, 118]. The NRT is defined as a geometric and photometric deformation of a regular texture. To track, the NRT was nested in a lattice based MRF model of a 3D spatiotemporal space. Then, the tracking algorithm used the topological invariant property of the dynamic NRT by combining a global lattice structure that characterizes the topological constraint among multiple textons (e.g., people) and an image observation model that handles local geometry and appearance variations. An algorithm was proposed by [57] to track a dense crowd of bats in thermal imagery. Multi target track initiation, recursive Bayesian tracking, clutter modeling, event analysis, and multiple hypotheses filtering were combined for this purpose. Impressive results were reported by tracking up to approximately eight hundred thousand bats. In the area of biological cell tracking, authors in [111] developed an algorithm for tracking thousands of cells in phase contrast time-lapse microscopy images. The tracking was performed in two stages. At the first stage a track compiler operating in a frame-by-frame manner was producing intermediate tracking results, named track segments, which were linked into cell trajectories at the second stage by a track linker overseeing the entire tracking history. A method to track in crowded scenes using selective visual attention has been proposed by [183]. In the method, the early selection process extracts a pool of attentional regions those were defined as the salient image regions which have good localization properties, and the late selection process dynamically identified a subset of discriminative attentional regions through a discriminative learning of the historical data on the fly.

There are several body of works which are based on region covariance and/or Riemannian matrices. Authors in [139, 140] proposed approaches for detection, labelling and tracking multiple targets. The targets are represented by region covariance matrices and particle filters perform the target tracking. Authors in [63] developed a visual tracking framework based on the incremental tensor subspace learning. In a different flavor, authors in [64] developed a visual tracking framework based on the novel Log-Euclidean Riemannian metric. In their framework, covariance matrices of image features in the five modes had been used to represent object appearance.

2.3.2 High-level information based tracking

There are several works which are based on high level image information for human detection and tracking in complex crowded situations. Authors in [68] used discrete choice models as motion prior to predict human motion patterns and fused this model in a visual tracker for improved performance. Discrete choice models are disaggregate behavioral models designed to forecast the behavior of individuals in choice situations. Another algorithm proposed by [6] which used high level knowledge of the scene and the behavior of the crowd into the tracking algorithm by computing a number of floor fields. Floor fields determine the probability of move from one location to another by converting the long-range forces into local ones. The experimental results of tracking of individuals in high density crowded scenes are impressive. A target tracking framework for unstructured crowded scenes can be found in [149]. Unstructured crowded scenes are defined as those scenes where the motion of a crowd appears to be random with different participants moving in different directions over time e.g., people walking on a zebra-crossing in opposite directions. To test the approach they performed experiments on a range of unstructured crowd domains, from cluttered time-lapse microscopy videos of cell populations in vitro to videos of sporting events.

Tracking methods using object contours [125, 186] and appearances [182, 82], which represent and estimate occlusion relationships between object by using the hidden variables of depth ordering of objects toward the camera, have been introduced. Authors in [166] proposed region covariance matrix

(RCM), which is a matrix of covariance of several image statistics computed inside a region defining a target. RCM-based algorithms with feature mapping functions have achieved good results in people detection, object tracking, and texture classification [144, 167, 166]. Other works using RCM can be found in [139, 140], where authors brought forward approaches for detection, labeling and tracking multiple targets. Targets are represented by region covariance matrices and particle filters carry out the target tracking. Experiments were conducted on five people scenarios. A crowded scene has a number of characteristics which makes the firsthand application of aforementioned tracking algorithms extremely challenging. Firstly, in high density crowds it is hard to discern individuals from each other, and consequently ownership of features cannot be computed reliably. Secondly, several occlusions occur due to interactions among the members of the crowd; therefore, even if reliable features are computed tracking over longer durations of time is difficult.

2.3.3 Discussion

A crowded scene has a number of characteristics which makes the direct application of many above-mentioned tracking algorithms extremely difficult. The reasons include, but are not limited to:

(i) In high density crowds it is hard to discern individuals from each other, and therefore, ownership of the features like color, spatial templates, interest points, contours, etc. cannot be computed reliably.

(ii) Severe occlusions occur due to interactions among the members of the crowd; consequently, even if reliable features are computed, tracking over longer durations of time is difficult.

We have proposed a temporal-spatial domain algorithm, falls into the category of low-level information based tracking, to track individual targets in the cases of sparse crowd, medium density crowd, and dense crowd. We have pointed how to extract the region of interest over frame in time so-called *target* by means of the MHI function and Hu's moment. This gives us a general overview of the number of individuals on the scene, which has been neglected on the existing directional literatures. We have introduced how to use the phase-correlation techniques for *targets* detection and tracking using distinct sharp peaks from the obtained peaks of target regions and the next frame's candidate regions. If two candidates or targets are similar, then their phase-correlation function gives a distinct sharp peak. Conversely, the peak of two dissimilar targets or candidates drops significantly. A great advantage behind of this hybrid technique is that it is not necessary to search the possible target region everywhere on the candidate frame except for the candidate regions. Consequently, the searching process becomes extremely rapid.

Chapter 3

Detection of Unusual Video Events

Contents

3.1 Study of Visual Attention

3.1.1 Overview

What happens if we focus our attention to a restricted part of our visual environment? We can not perceive all the components lying our visual field with equal interest. Visual attention allows a certain spatial location (salient location) and certain types of visual features (salient features). Salience (also called saliency) at a given location is determined primarily by how different the location is from its surround in color, orientation, motion, depth, etc. In video frame or image, our visual attention allows the salient parts of the video which is more distinguishable from other parts. Since visual attention allows to focus analysis and processing on some restrained parts of images and frames, it has emerged in recent years as a convincing tool to make robot and computer vision more and more operative in a wide variety of jobs.

3.1.2 Visual attention computational models

Computing visual saliency has become a good topic of recent technological interest. The detection of salient regions in the visual field, which is similar to what is frequently called interest point detection in computer vision. For example, a multi-scale algorithm for the selection of salient regions of an image was introduced and its application to matching type problems such as tracking, object recognition and image retrieval was demonstrated [94, 93]. However, visual salience provides a relatively inexpensive and rapid mechanism to select a few likely candidates and annihilate explicit clutter [84, 135]. Several computational models of visual attention have been suggested. Computational models have been developed which use known properties of the visual system to bring forth a saliency map or landscape of visual salience across an image [84, 106]. In these models, the visual properties present in an image cause to happen to a 2D map that emphatically marks regions which are different from their surround on image dimensions e.g., color, intensity, contrast, and edge orientation [84, 106, 142, 164], contour junctions, termination of edges, stereo disparity, and shading [106], and dynamic factors such as motion [106]. The maps are generated for each image dimension over multiple spatial scales and are then combined to create a single saliency map. Regions which are uniform along some image dimension are scrutinized uninformative, whereas what differ from neighboring regions across spatial scales are considered to be potentially informative and worthy of disorder. Many saliency algorithms have been proposed in the computer and biological vision literatures. The classification of visual attention has long been believed to be driven by the interaction of two complementary components *bottom-up* and *top-down*. Saliency algorithms based on *bottom-up* (e.g., [84]) are fast and image-driven mechanism, while *top-down* algorithms (e.g., [136, 69]) are slower and goal-driven mechanism. The bottom-up component computes the visual salience of scene locations in different feature maps extracted at multiple spatial scales. The top-down component uses accumulated statistical knowledge of the visual features of the desired search target and background clutter [136]. Both models are important in visual surveillance, for example, it is important to detect goal-relevant (top-down) targets like suspects, and also to notice unexpected visual events like person falling (bottom-up).

3.1.3 Itti-Koch computational model

We have discussed about the Itti-Koch [84] bottom-up model for computing visual attention. The Fig. 3.1 shows an overview of the model of Itti-Koch [84]. Visual features are computed using linear filtering at eight spatial scales, followed by center-surround differences, which compute local spatial contrast in each feature dimension for a total of 42 maps. An iterative lateral inhibition scheme instantiates competition for salience within each feature map. After competition, feature maps are combined into a single conspicuity map for each feature type. The three conspicuity maps then are summed into the unique topographic saliency map. The WTA (winner-take-all) detects the most salient location and directs attention towards it. An inhibition-of-return mechanism transiently suppresses this location in the saliency map, such that attention is autonomously directed to the next most salient image location.

Input is provided in the form of digitized images, from a variety of sources including a consumer-electronics NTSC video camera. However, they have considered the following four assumptions. Firstly, visual input is represented in the form of iconic (appearance-based) topographic feature maps. Two crucial steps in the construction of these representations consist of center-surround computations

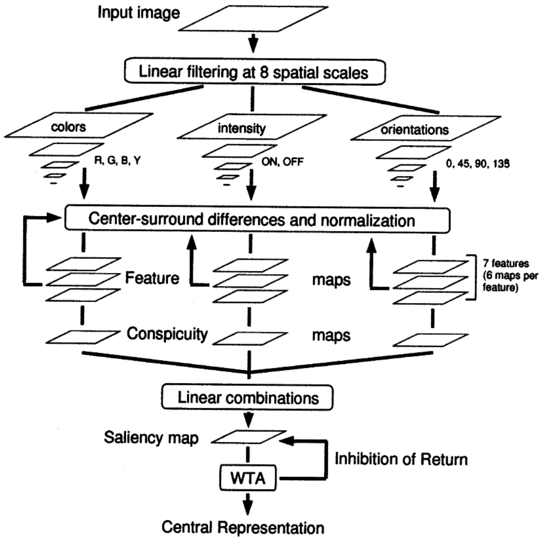


Figure 3.1: Schematic diagram for the saliency computational model used by Itti-Koch [84].

in every feature at different spatial scales, and within-feature spatial competition for activity. Secondly, information from these feature maps is combined into a single map that represents the local *saliency* of any one location with respect to its neighborhood. Thirdly, the maximum of this saliency map is the most salient location at a given time, and it determines the next location of the attentional searchlight. Finally, the saliency map is endowed with internal dynamics allowing the perceptive system to scan the visual input such that its different parts are visited by the focus of attention in the order of decreasing saliency. Input is allowed for the form of static color images, normally digitized at 640×480 resolution. Low-level vision features (color channels tuned to red, green, blue and yellow hues, orientation and brightness) are extracted from the original color image at several spatial scales, using linear filtering. The different spatial scales are created using Gaussian pyramids, which consist of gradually low-pass filtering and sub-sampling the input image. Pyramids have a depth of nine scales, providing horizontal and vertical image reduction factors ranging from 1:1 (level 0; the original input image) to 1:256 (level 8) in consecutive powers of two [85, 84]. Each feature is computed in a center-surround structure akin to visual receptive fields. Center-surround operations are implemented in the model as differences between a fine and a coarse scale for a given feature. The center of the receptive field corresponds to a pixel at level $c \in \{2,3,4\}$ in the pyramid, and the surround to the corresponding pixel at level $s = c + \delta$, with $\delta \in \{3,4\}$. Six feature maps are computed for each type of feature at scales 2–5,

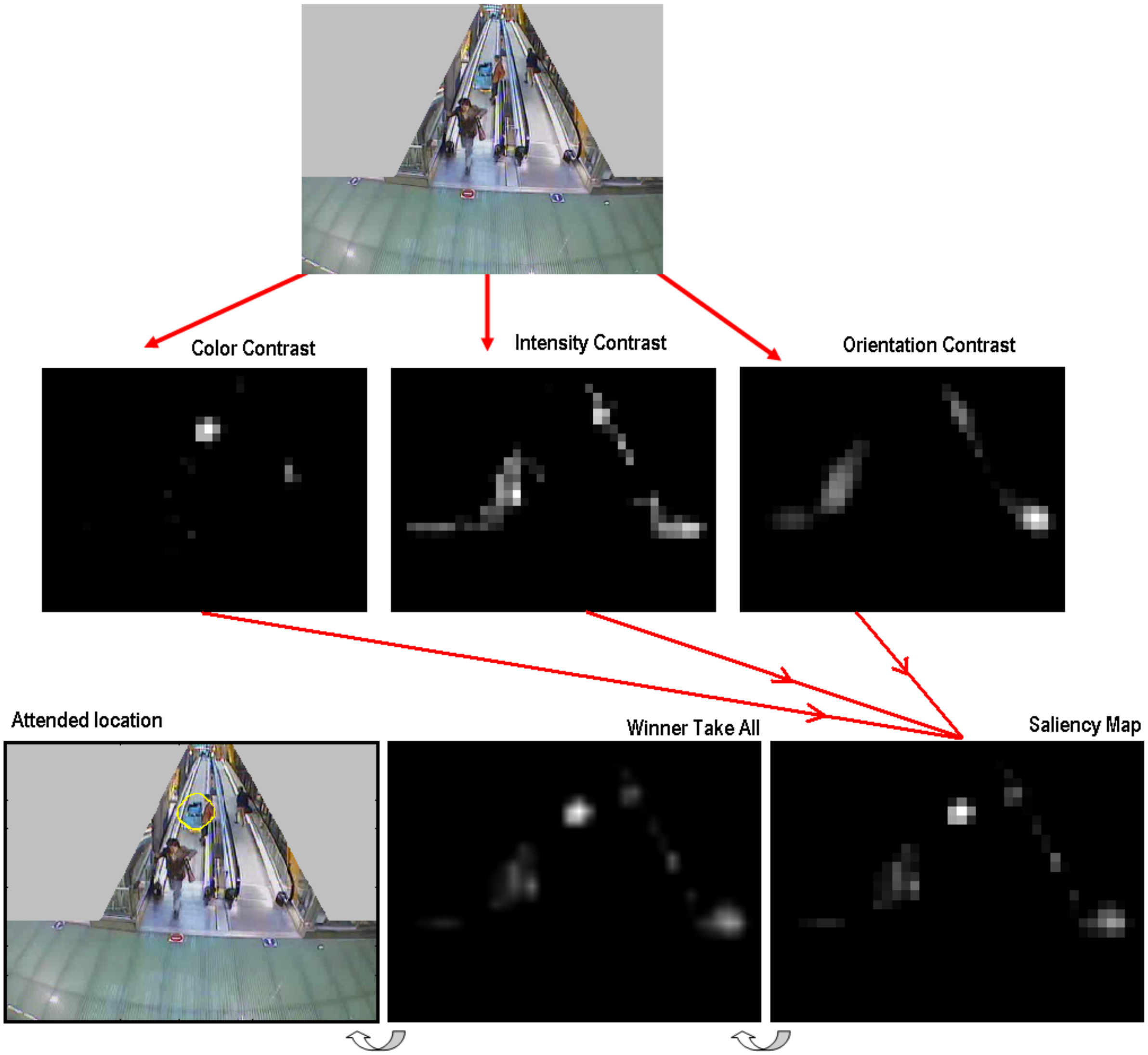


Figure 3.2: An example of the model with an input 640 × 480 pixels color image from a video of the *Escalator dataset* [132] and inside of the yellow marked region belongs to the output attended location.

2–6, 3–6, 3–7, 4–7, and 4–8. Seven types of features are computed in this manner from the low-level pyramids: one feature type encodes for on/off image intensity contrast, two encode for red/green and blue/yellow double-opponent channels, and four encode for local orientation contrast. The six feature maps for the intensity feature type encode for the modulus of image luminance contrast, i.e., the absolute value of the difference between intensity at the center (e.g., one of the three c scales) and intensity in the surround (e.g., one of the six $s = c + \delta$ scales). To isolate chromatic information, each of the red, green, and blue channels in the input image are first normalized by the intensity channel; a quantity is then computed by center-surround differences across scales. Each of the six red/green feature maps is created by first computing (red–green) at the center, then subtracting (green–red) from the surround, and finally resulting the absolute value. Six blue/yellow feature maps are similarly created. Local orientation is obtained at all scales through the creation of oriented Gabor pyramids from the intensity image. Four orientations are used (0°, 45°, 90°, and 135°) and orientation feature maps are obtained from absolute center-surround differences between these channels. These maps encode, as a group, how different the average local orientation is between the center and surround scales. After normalization, the feature maps for intensity, color, and orientation are summed across scales into three separate conspicuity maps: one for intensity, one for color, and one for orientation. The Fig. 3.2 shows an example of the model with an input 640×480 pixels color image from a escalator video and inside of the yellow marked region belongs to the output attended location. Feature maps are extracted from the input image at several spatial scales, and are combined into three separate conspicuity maps (intensity, color and orientation; see the Fig. 3.1) at scale 4 (30×40 pixels). The three conspicuity maps that encode for saliency within these three domains are combined and fed into the single saliency map (also 30×40 pixels). The winner-take-all successively selects, in order of decreasing saliency, the attended locations. Once a location has been attended to for some short interval, it is transiently suppressed in the saliency map by the inhibition of return mechanism which helps to find the next attended location. The motivation for the creation of three separate channels and their individual normalization is the hypothesis that similar features compete strongly for salience, while different modalities contribute independently to the saliency map. Conspicuity maps are linearly summed into the unique saliency map, which resides at scale 4 (reduction factor 1:16 compared to the original image). At any given time, the maximum of the saliency map corresponds to the most salient stimulus to which the focus of attention should be directed next, to allow for more detailed inspection. To find the most salient location, the maximum of the saliency map is determined by means of a winner-take-all algorithm.

3.1.4 Discussion

We tested on many varieties of real world images with Itti-Koch [84] bottom-up model[1]. All images were in color, contained some amounts of noise, strong local variations in illumination, shadows and reflections, large numbers of objects often partially occluded, etc. The results report that the system scans the image in an order which makes functional sense in few behavioral situations.

The Fig. 3.2 shows an example of the working of the model with a 640 × 480 pixels color image having few people. The image depicts a situation on the escalator just before hitting the trolley to a standing man. Parallel feature extraction yields the three conspicuity maps for color contrasts, intensity contrasts, and orientation contrasts. These are combined to form input 6 to the saliency map. The most human attended location is the pile of blue bags on the trolley which has been detected successfully. On the other hand, image at the Fig. 3.3 illustrates the situation after falling the person on the escalator exit. The first attended location has been detected as the dropped one of the colored bags, which were laid on the trolley, near to the falling person. This most salient location was appeared very strongly in color contrasts with a simulated time of 96.9 ms. After the inhibition-of-return feedback inhibits this location in the saliency map, the next most salient locations are successively selected. For example, second, third, fourth, and fifth attended locations were appeared very strongly in intensity (182.1 ms), color (182.2 ms), intensity (254.6 ms), and intensity (320.7 ms) contrasts, respectively. Noticeably, the region of the fallen person was marked as fourth attended location. As a result, the method cannot detect the sudden person fall. Other image sequences also report the same undetected results. That means, the method hardly detects abnormalities from the crowded scenarios. Most realizable reason is that solely spatial information color (six hues within the color dimension), intensity (four intensities within the luminance dimension), and orientation (four orientations within the orientation dimension) are not enough to detect abnormalities.

Along with the temporal information, the spatial information can provide much better detection result. In general, spatio-temporal information takes into account motion as an informative feature to detect and segment interesting objects or targets by means of optical flow computation, block matching or other motion detection methods. We (in [SMD08a]) have also investigated motion saliency, which helps to detect moving objects whose motion is discontinuous to its background e.g., a vehicle moving in the wrong direction while others are in the right direction would be detected as salient; a static person/object among other moving persons/objects would also be salient (e.g., Fig. 3.4). When analyzing

[1] A good implemented can be found in http://ilab.usc.edu/toolkit/

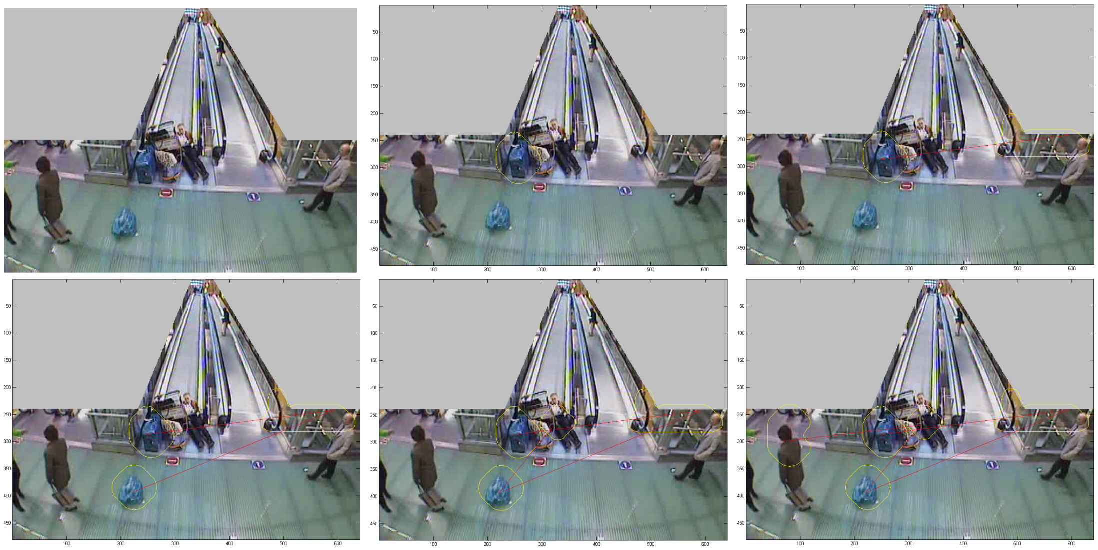


Figure 3.3: Cannot detect person falling event. Top-leftmost is the original image. In a decreasing order of attention, the attended locations have been exhibited by yellow colored contours for the winners centered at (247,287), (593,242), (220,394), (321,270), and (69,299) with simulated time 96.9ms, 182.1ms, 182.2ms, 254.6ms, and 320.7ms, respectively. Centers are connected by red lines.

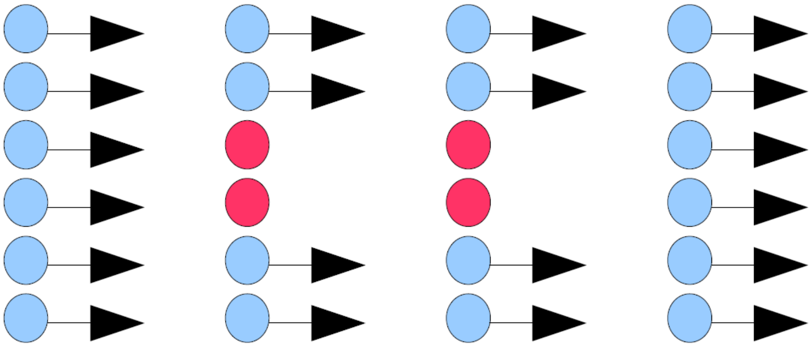


Figure 3.4: Example of four static objects among several moving objects. By definition, the four objects are salient because they are detected as regions with motion discontinuities.

video, visual saliency has to deal with moving objects. Motion is a very salient cue for bottom-up attention and should be part of a model of saliency based attention [172]. Motion saliency models e.g., [158] detect regions of interest corresponding to moving objects whose motion is salient to its background. Authors in [158], implicitly estimating global and local motion, generated motion saliency by using the rank deficiency of gray scale gradient tensors within the local region neighborhoods of the image. However, with our investigation, so far, we can conclude that saliency based models can solely be suitable for sparse crowd scenes like the Fig. 3.2 to detect abnormalities. As a result, we have to go further for a good approach to detect anomalies in various densities crowded scenes like the Fig. 1.1.

In the next sections, we have proposed different spatiotemporal information based methods for crowd behavior analysis and detection of abnormal activities.

3.2 Covariance Matrix Approach

3.2.1 Overview

Our approach (in [SID08a]) presented in this section detects abnormal events principally from unidirectional flow of crowd (e.g., escalators). The video frames are labeled *normal* or *abnormal* based on the distance measure between covariance matrices of the distributions of the optical flow vectors computed on consecutive frames. These flow vectors are the result of tracking a set of features points discovered by the Harris corner detector applied on each frame considering a region of interest. This region is produced by background subtraction to form a two dimensional histogram of motion called motion heat map. The approach is tested against a single camera data-set placed in the escalator exits in an airport.

A simple flow diagram of the proposed framework has been shown in Fig.3.5. The approach is characterized by optical flow patterns of human behaviors followed by some statistical treatments, to detect abnormal events mainly in onward crowd flow (e.g., escalators). An event is defined to be an observable action or change of state in a video stream that would be important for security management. We have started by calculating a motion heat map during a period of time to extract the main regions of motion activity. The use of heat map image improves the quality of the results and reduces processing time which is an important factor for real-time applications. Points of interest are extracted in the hot regions of the scene. Optical flow is computed on these points of interest, delimited by the hot areas of the scene. The optical flow information from video presents the crowd multi-modal behaviors as optical flow patterns variate in time. There is sufficient perturbation in the optical flow pattern in the crowd in case of abnormal and/or emergencies situations (e.g., Fig. 3.28). We have constructed *covariance*

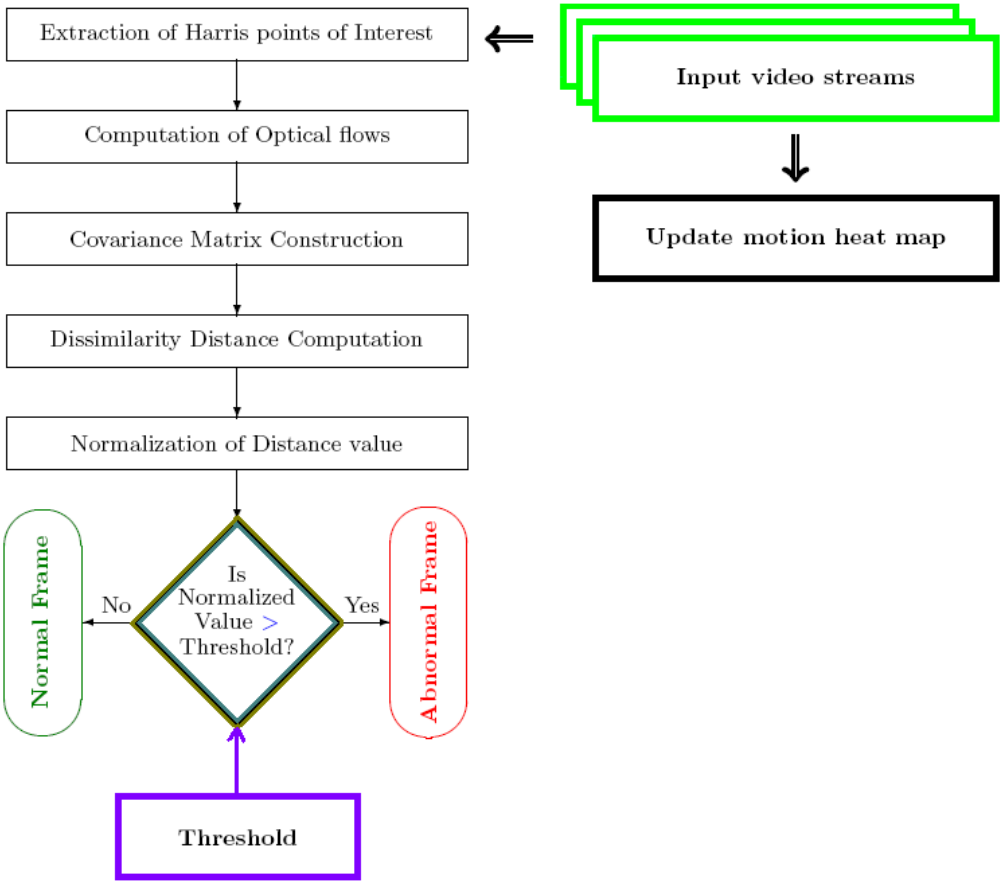

Extraction of Harris points of Interest
Input video streams
Computation of Optical flows
Update motion heat map
Covariance Matrix Construction
Dissimilarity Distance Computation
Normalization of Distance value
Normal Frame
No
Is Normalized Value > Threshold?
Yes
Abnormal Frame
Threshold

Figure 3.5: Block diagram of the proposed Covariance Matrix approach

matrix (CM) using the extracted spatiotemporal knowledge of optical flow. A CM is merely collection of several variance-covariances in the form of a square matrix. We have computed the dissimilarity as a distance measure between two consecutive CMs. We have studied the normalized distance measure to differentiate normal or abnormal frame based on a defined value (label) called threshold.

3.2.2 Low-level features Extraction

3.2.2.1 Motion heat map

A heat map is a graphical representation of data where the values taken by a variable in a 2-dimensional ($2D$) map are represented as colors. Motion heat map is a $2D$ histogram expressing briefly the most important regions of motion activity. This histogram is built from the accumulation of binary blobs of moving objects, which were extracted following background subtraction method [112]. The obtained map is used as a mask to define the $region\ of\ interest$ for the next step of the method. Fig.3.6 makes noticeable an occurrence of the obtained heat map from an escalator camera view. Better red region (region of interest) expects longer video duration. The use of heat map ameliorates the quality of the results and reduces processing time which is an important factor for real-time applications. The results are more significant when the video duration is long. In practice, even for the same place, the properties of abnormal events may vary depending on the context (day-night, indoor-outdoor, occasion, vacation, etc.). We build a motion heat map for each set of conditions. It is not necessary to consider in detail the whole scene, and fastidiously the scene where there are few motion intensities or no motions. Thus, the approach directs the attention on the processing of specific regions where the density of motions is high. The threshold related to the density elevation is a contextual information.

3.2.2.2 Points of interest extraction

Moravec's corner detector [133] is a relatively simple algorithm that was used by Moravec and others, but is now commonly considered out-of-date. It is not rotationally invariant (a property prevalent even in more modern corner detectors) as the response is not invariant with respect to direction (anisotropic), is considered to have a noise response, and is susceptible to reporting false corners along edges and at isolated pixels so is sensitive to noise. Nevertheless, it is computationally efficient which was critical for Moravec as he was interested in a real-time application and had minimal computational power at his disposal. The other way around the Harris corner detector [76] is computationally demanding, but directly addresses many of the limitations of the Moravec corner detector. In our approach, we consider Harris corner as a point of interest. The Harris corner detector is a famous point of interest detector due to its strong invariance to rotation, scale, illumination variation, and image noise [154]. It is based on the local auto-correlation function of a signal, where the local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions. Assume a point (x,y) and a shift (Δx,Δy), then the local auto-correlation function is defined as:

$$c(x,y) = \Sigma_w [I(x_i,y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2 \tag{3.1}$$

where $I(.,.)$ denotes the image function and (x_i,y_i) are the points in the smooth circular window w centered on (x,y). The shifted image is approximated by a Taylor expansion truncated to the first order

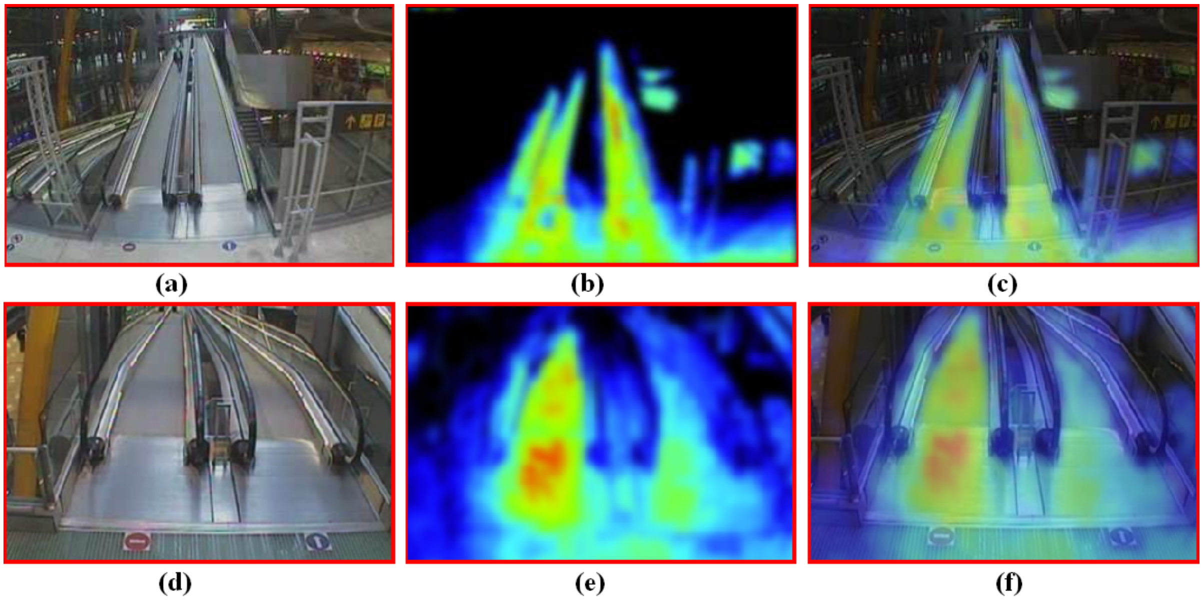


Figure 3.6: Images at (a) and (d) belong to camera view. The generated *motion heat maps* are depicted at (b) and (e). Images at (c) and (f) are masked view where red regions recommend *region of interests*.

terms as:

$$I(X^i_\delta, Y^i_\delta) \approx I(x_i, y_i) + [I_x(x_i, y_i)I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{3.2}$$

where $X^i_\delta = x_i + \Delta x$, $Y^i_\delta = y_i + \Delta y$; and $I_x(.,.)$ & $I_y(.,.)$ denote the partial derivatives in x & y, respectively. Substituting the right hand site of Eq. 3.2 into Eq. 3.1 yields:

$$c(x,y) = \Sigma_w(\ [I(x_i, y_i)I_y(x_i, y_i)] \ \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix})^2 \tag{3.3}$$

$$= [\Delta x \Delta y] M(x,y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{3.4}$$

where

$$M(x,y) = \begin{pmatrix} \Sigma_w (I_x(x_i, y_i))^2 & \Sigma_w I_x(x_i, y_i) I_y(x_i, y_i) \\ \Sigma_w I_x(x_i, y_i) I_y(x_i, y_i) & \Sigma_w (I_y(x_i, y_i))^2 \end{pmatrix}. \tag{3.5}$$

The 2×2 symmetric matrix $M(x,y)$ captures the intensity structure of the local neighborhood. Let λ_1 and λ_2 are the eigenvalues of matrix $M(x,y)$. The eigenvalues form a rotationally invariant description. There are three cases to be considered [76]:

- **No point of interest is found**: If both λ_1 & λ_2 are small, so that the local auto-correlation function is flat, i.e., little change in $c(x,y)$ in any direction, then the windowed image region is of approximately constant intensity.

- **An edge is found**: If one eigenvalue is high and the other is low, so the local auto-correlation function is rigid shaped, then only shifts along the ridge (i.e., along the edge) cause little change in $c(x,y)$ and significant change in the orthogonal direction.

- **A point of interest is found**: If both λ_1 & λ_2 are high, so the local auto-correlation function is sharply peaked, then shifts in any direction result in a significant increase in $c(x,y)$.

The left image on Fig.3.7 lets on an example of Harris corner detector. We deem that in video surveillance scenes, camera positions and lighting conditions admit to get a large number of corner features that can be easily captured and tracked.

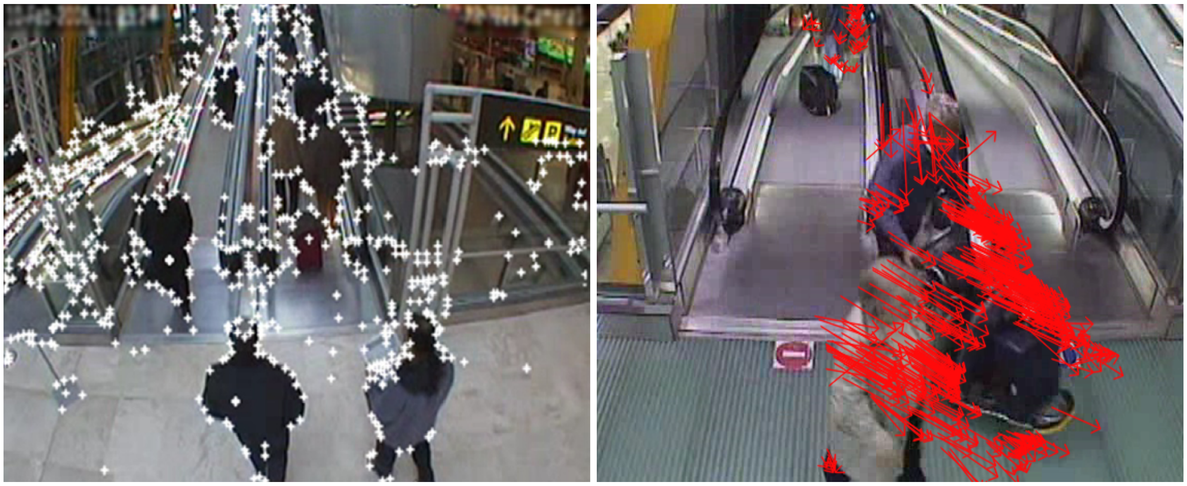


Figure 3.7: White points and red arrows pertain to Harris corner and optical flow vectors, respectively.

3.2.2.3 Estimation of optical flow

Good feature selection plays a critical role for detection and classification issues. Color, edges, optical flow, texture, gradient, and filter responses are some common example of features. Optical flow is the velocity field which warps one image into another (normally very similar) image. Optical flow algorithms provide estimation of the motion fields and are based on the idea that for most points in the image, neighboring points have approximately the same brightness. The goal of optical flow technique is to compute an approximation to the 2D motion field, a projection of the 3D velocities of surface points onto the imaging surface, from spatiotemporal patterns of images intensity [79, 169]. Normally, optical flow information is not the same as the motion field. The motion field is represented by the field of vectors that show the displacement of points in the optical field relative to the observer, whereas optical flow shows a velocity field of pixels in the image. Optical flow may be used to perform a wide variety of tasks such as motion detection, object segmentation, time-to-collision and focus of expansion calculations, motion compensated encoding and stereo disparity measurement.

There are number of ways to compute optical flow. The Lucas-Kanade method calculates the motion between two image frames which are taken at times t and $t+dt$ at every pixel position, where dt is the time deviation. This method is treated as differential since it is based on local Taylor series approximations of the image signal, i.e., it uses partial derivatives with respect to the spatial and temporal coordinates. The advantage of this method is the comparative robustness in presence of noise. To calculate the optical flow between successive video frames the well known combination of feature selection as introduced by Shi and Tomasi [160] and the algorithm of Lucas and Kanade for feature tracking [123] is used. Feature selection finds image blocks which are believed to allow the exact estimation of optical flow translation vector. The Shi-Tomasi algorithm makes use of the smallest eigenvalues of an image block as criterion to ensure the selection of features which can be tracked reliably by the Lucas-Kanade tracking algorithm. This algorithm matches the selected image blocks with blocks in the next frame using an efficient gradient descent technique. A pyramidal implementation of this algorithm is used to deal with larger feature displacements by avoiding local minima in a coarse to fine approach [26]. This combination has proven to allow fast and reliable computation of optical flow information [12]. The parameters which control the performance of the pyramidal Lucas-Kanade algorithm are [104]: (i) the block size, (ii) the number of resolution levels, and (iii) some termination criteria. The block size and the number of resolution levels are determined by the image size and the expected block displacement between two frames of an image sequence. The termination criteria are the maximum number of iterations for the gradient descent approach and an accuracy requirement for block matching to allow early termination.

In our optical flow estimation step, we have used the pyramidal implementation of optical flow algorithm. Once we define the points of interest (features), we track those features over the next frames using the above combination feature tracker of Kanade-Lucas-Shi-Tomasi. An example of optical flow vectors produced by the feature tracker shown in the right image on Fig. 3.7. After matching features between frames, we can consider that the result is a set of vectors $\mathbf{V_k(j)}$ of n elements over time:

$$\mathbf{V_k(j)} = \begin{bmatrix} \mathbf{x_1} & \mathbf{y_1} & \mathbf{v_1} & \alpha_1 \\ \mathbf{x_2} & \mathbf{y_2} & \mathbf{v_2} & \alpha_2 \\ \mathbf{x_3} & \mathbf{y_3} & \mathbf{v_3} & \alpha_3 \\ . & . & . & . \\ . & . & . & . \\ \mathbf{x_i} & \mathbf{y_i} & \mathbf{v_i} & \alpha_i \\ . & . & . & . \\ . & . & . & . \\ \mathbf{x_n} & \mathbf{y_n} & \mathbf{v_n} & \alpha_n \end{bmatrix} \tag{3.6}$$

where $k = 1,2,3,\ldots n$, $i \in k$, $j \in \{x, y, v, \alpha\}$, and

- $\mathbf{x_i} \Rightarrow$ *x-coordinate* of any feature element i,
- $\mathbf{y_i} \Rightarrow$ *y-coordinate* of the i,
- $\mathbf{v_i} \Rightarrow$ velocity v of the i,
- $\alpha_i \Rightarrow$ moving direction α of the i.

Images in Fig.3.7 (b) and (c) give evidence of the set of vectors obtained by optical flow feature tracking in two different situations. The image in Fig. 3.7 (b) divulges an orderly vector flow. The image in Fig. 3.7 (c) substantiates a littered vector flow due to the breakdown situation. This step also allows removal of static and noise features. Static features are the features that moves less than two pixels. Noise features are the isolated features that have a big angle and distance difference with their near neighbors due to tracking calculation errors.

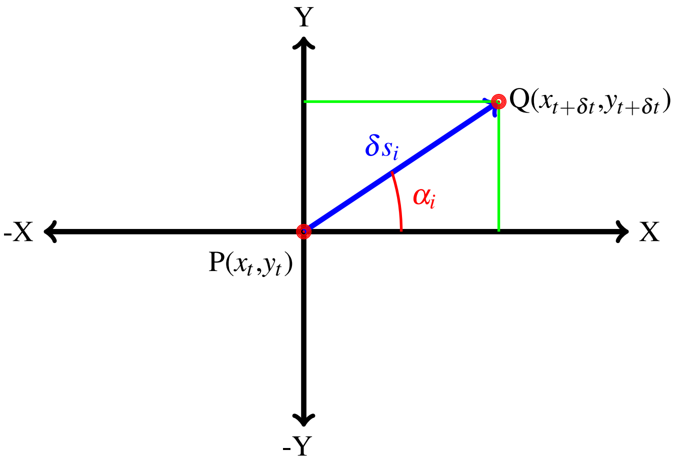


Figure 3.8: Moving direction α_i of a feature i.

3.2.2.4 Estimation of velocity & direction of an interest point

Fig.3.8 illustrates any feature i in the frame f with its coordinate $P(x_t, y_t)$ and its matched in the frame $f+1$ with coordinate Q($x_{t+\delta t}$,$y_{t+\delta t}$) and elapsed time δt. We can easily calculate the displacement δs_i (change in position or place is called a displacement) of the feature i using Euclidean metric as:

$$\delta s_i = \sqrt{x_i^2 + y_i^2} \tag{3.7}$$

where $x_i = Q_{x_{t+\delta t}} - P_{x_t}$ and $y_i = Q_{y_{t+\delta t}} - P_{y_t}$. The rate of change of position is called velocity (the rate of motion is called speed which is equal to the magnitude of velocity) v can be determined by the first derivative of δs_i with respect to time δt:

$$v_i = \frac{\delta s_i}{\delta t} = \sqrt{(\frac{x_i}{\delta t})^2 + (\frac{y_i}{\delta t})^2}. \tag{3.8}$$

The direction of motion (α_m) of the feature i can be calculated using the following trigonometric function:

$$\alpha_m = atan(\frac{y_i}{x_i}). \tag{3.9}$$

Notwithstanding, there are several potential problems if we have a desire to calculate motion direction using Eq. 3.9, for instances:

- Eq. 3.9 does not show expected performance for a complete range of angles from 0° to 360°. Only angles between -90° and $+90^\circ$ will be returned, other angles will be (say 180°) out-of-phase. For example, let us consider two defined points $(x_1 = 3, y_1 = 3)$ and $(x_2 = -3, y_2 = -3)$. Using the Eq. 3.9, the point $(x_2 = -3, y_2 = -3)$ will produce the same angle as the point $(x_1 = 3, y_1 = 3)$ will do, but from Fig.3.8 we could consider that these are not in same quadrant.
- Points on the vertical axis have $x_i = 0$, hence, if we wish to calculate y_i/x_i we will get ∞ which will generate an exception when calculated on the computer.

In order to keep away from these problems, we apply the $atan2(y_i, x_i)$ function which takes both x_i and y_i as arguments. Henceforth, the accurate direction of motion α_i, where $\alpha_i = atan2(y_i, x_i)$, of the feature i can be calculated as:

$$\alpha_i = \begin{cases} \phi.\mathbf{sign(y_i)} & if\ x_i > 0, y_i \neq 0 \\ \mathbf{0} & if\ \ x_i > 0, y_i = 0 \\ \frac{\pi}{2}.\mathbf{sign(y_i)} & if\ x_i = 0, y_i \neq 0 \\ \mathbf{undefined} & if\ x_i = 0, y_i = 0 \\ (\pi - \phi).\mathbf{sign(y_i)} & if\ x_i < 0, y_i \neq 0 \\ \pi & if\ \ x_i < 0, y_i = 0 \end{cases}$$

where ϕ is the angle in $[0, \pi/2]$ such that $\phi = atan(|\frac{y_i}{x_i}|)$. The sign function $sign(y_i)$ can be defined as:

$$sign(y_i) = \begin{cases} -1 & if\ \ y_i < 0 \\ 0 & if\ \ y_i = 0 \\ 1 & if\ \ y_i > 0. \end{cases}$$

As a consequence, the function $atan2(y,x)$ gracefully deals with infinite slope, and places the angle in the correct quadrant. For instance, $atan2(0,3) = 0$, $atan2(0,-3) = \pi$, $atan2(3,3) = \pi/4$, $atan2(-3,-3) = -3\pi/4$, etc.

3.2.3 Covariance Matrices Construction

Covariance is a statistical measure of correlation of the continual changes from one point or condition to another of two different quantities. In statistics and probability theory, a covariance matrix or dispersion matrix is a matrix of covariances between elements of a random vector. It is the bona fide generalization to higher dimensions of the concept of the variance of a scalar-valued random variable. The diagonal entries of the covariance matrix represent the variance of each feature and the non-diagonal entries represent the covariances. Due to symmetry covariance matrix has only $(m^2+m)/2$ different values, where m is number of either rows or columns.

We construct a 4×4 CM for representing a video frame using the data obtained in Eq. 3.6 where x_i & y_i as spatial information, and v_i & α_i as temporal information. Assume C_f be a CM for a frame, we define C_f as:

$$C_f = \begin{pmatrix} \mathbf{x} & \mathbf{xy} & \mathbf{xv} & \mathbf{x}\alpha \\ \mathbf{yx} & \mathbf{y} & \mathbf{yv} & \mathbf{y}\alpha \\ \mathbf{vx} & \mathbf{vy} & \mathbf{v} & \mathbf{v}\alpha \\ \alpha\mathbf{x} & \alpha\mathbf{y} & \alpha\mathbf{v} & \alpha \end{pmatrix} \tag{3.10}$$

where diagonal elements $\mathbf{x}$, $\mathbf{y}$, $\mathbf{v}$, and α are variances, and non-diagonal elements are covariances. We compute the (p,q)-th element of the C_f in the following statistical formula:

$$C_f(p,q) = \frac{1}{n-1}\left[\sum_{k=1}^{n} V_k(p)V_k(q) - \frac{1}{n}\sum_{k=1}^{n} V_k(p)\sum_{k=1}^{n} V_k(q)\right] \tag{3.11}$$

where $\{p,q\} \in \{x, y, v, \alpha\}$.

3.2.4 Covariance Matrices Dissimilarity Computation

Measuring the dissimilarity between images and parts of images is of central importance for low-level computer vision [150]. Investigating covariance matrices is an elementary task in mensuration design [65]. Mensuration is the branch of geometry that deals with the measurement of length, area, or volume. Based on the eigenvalues of the covariance matrix, in 1972, Grafarend [73] proposed a development of a satisfactory measure for comparing two covariance matrices. If the information of a Gaussian variable σ^2 (variance or standard deviation squared) increases with $log_e\sigma^2$, the author guessed the squared sum $d^2 = \Sigma_i log_e^2\lambda_i$ of the logarithms of the eigenvalues λ_i to be a better measure, as deviations in both directions would be handled the same amount if measured in percentage. It would be worth wanting that the similarity between two covariance matrices manifests the deviation in variance in both directions according to the ratio of the variances. As a consequence, deviations in variance in both by a factor ζ could be evaluated equally as a deviation by a factor $1/\zeta$, in case of $\zeta = 1$ the factor indicates no difference. Authors in [65], proposed a better metric for distance measure between symmetric positive definite $m\times m$ matrices. Esteem as C_{f_i} and $C_{f_{i+1}}$ are two consecutive 4×4 CMs, then the *distance* *measure* $d(C_{f_i}, C_{f_{i+1}})$ proposed in [65] to measure the dissimilarity of two covariance matrices can be defined by dint of:

$$d(C_{f_i}, C_{f_{i+1}}) = \sqrt{\sum_{k=1}^{4} log_e^2 \lambda_k(C_{f_i}, C_{f_{i+1}})} \tag{3.12}$$

where $\lambda_i(C_{f_i}, C_{f_{i+1}})_{i=1...4}$ are four generalized eigenvalues of C_{f_i} and $C_{f_{i+1}}$, computed from $\lambda_i C_{f_i} \mathbf{x_i} - C_{f_{i+1}} \mathbf{x_i} = 0$ and $\mathbf{x_i} \neq 0$ are generalized eigenvectors. The logarithm guarantees, that deviations are measured as factors, whereas the squaring guarantees factors ζ and $1/\zeta$ being evaluated equally. Summing squares is done in close resemblance with the Euclidean metric. The distance measure $d(C_{f_i}, C_{f_{i+1}})$ satisfies following metric axioms for positive definite symmetric matrices C_{f_i} and $C_{f_{i+1}}$ [65]:

(i) *Positivity*: $d(C_{f_i}, C_{f_{i+1}}) \geq 0$ and $d(C_{f_i}, C_{f_{i+1}}) = 0$ only if $C_{f_i} = C_{f_{i+1}}$;

(ii) *Symmetry*: $d(C_{f_i}, C_{f_{i+1}}) = d(C_{f_{i+1}}, C_{f_i})$;

(iii) *Triangle inequality*: $d(C_{f_i}, C_{f_{i+1}}) + d(C_{f_i}, C_{f_{i+2}}) \geq d(C_{f_{i+1}}, C_{f_{i+2}})$.

3.2.5 Normalization of Dissimilarity Distances

Now, we wish to transfer each dissimilarity distance measure into a normalized distance value ranges between 0 and 1. Assume that $d(C_{f_i}, C_{f_{i+1}})$ be any dissimilarity distance measure between any two consecutive frames f_i and f_{i+1}. We could use the simple equation like Eq. 3.13 for moralization, but the normalized values fall in a congested range (scaling problem) which will arise problem specially in threshold selection.

$$Normalized\ value = (1 - \frac{1}{log\, d(C_{f_i}, C_{f_{i+1}})}) \tag{3.13}$$

To solve the scaling problem, we use cumulative distribution function (cdf) which has strict lower and upper bounds between 0 and 1. Assuming $\Phi(d(C_{f_i}, C_{f_{i+1}}))$ denotes the cdf of $d(C_{f_i}, C_{f_{i+1}})$. Then $\Phi(d(C_{f_i}, C_{f_{i+1}}))$ can be expressed in terms of a special function called the $error\ function$ (erf) or $Gauss\ error\ function$, as:

$$\Phi_{\mu,\sigma}(d(C_{f_i}, C_{f_{i+1}})) = \frac{1}{2}\left[1 + erf\{\frac{d(C_{f_i}, C_{f_{i+1}}) - \mu}{\sigma\sqrt{2}}\}\right] \tag{3.14}$$

where $\sigma > 0$ is the standard deviation and the real parameter μ is the expected value. The erf can be defined as a $Maclaurin$ series:

$$erf(d(C_{f_i}, C_{f_{i+1}})) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n \{d(C_{f_i}, C_{f_{i+1}})\}^{2n+1}}{n!(2n+1)} \tag{3.15}$$

$$= \frac{2}{\sqrt{\pi}}\{d(C_{f_i},C_{f_{i+1}}) - \frac{d(C_{f_i},C_{f_{i+1}})^3}{3} + \frac{d(C_{f_i},C_{f_{i+1}})^5}{10} - \frac{d(C_{f_i},C_{f_{i+1}})^7}{42} + \frac{d(C_{f_i},C_{f_{i+1}})^9}{216} - \ldots\}. \quad (3.16)$$

Since $d(C_{f_i},C_{f_{i+1}})$ is skewed to the right (positive-definite) and variances also some what large, we can use Log-normal distribution. Log-normal distributions are usually characterized in terms of the log-transformed variable, using as parameters the expected value, or mean, of its distribution, and the standard deviation. The structure of log-normal distribution of the Eq. 3.14 yields:

$$N(d(C_{f_i},C_{f_{i+1}})) = \frac{1}{2}\left[1 + erf\{\frac{log(d(C_{f_i},C_{f_{i+1}})) - \mu}{\sigma\sqrt{2}}\}\right] \quad (3.17)$$

where $N(d(C_{f_i},C_{f_{i+1}}))$ is the normalized value of $d(C_{f_i},C_{f_{i+1}})$. With the help of Eq. 4.48 and 3.16, and knowing the values of μ and σ (say $\mu = 0$, $\sigma = 15$) we can emphatically estimate the value of $N(d(C_{f_i},C_{f_{i+1}}))$ between 0 and 1.

3.2.6 Deciding Normal or Abnormal Events

We considered that the characteristic of the state of a collapse situation as a signal of sudden change with a high peak height of duration. If there exists such signal then there is an abnormal event. The decision for normal or abnormal events is to be taken by comparing the calculated and normalized measure with a specific threshold. We compare each calculated value of $N(d(C_{f_i},C_{f_{i+1}}))$ with a predefined normalized threshold T_N, i.e., abnormal frame can be detected if $N(d(C_{f_i},C_{f_{i+1}})) > T_N$, otherwise normal frame. The theoretical aspect of computing T_N is that we consider the maximum number of $N(d(C_{f_i},C_{f_{i+1}}))$ in large videos those contain exclusively normal events:

$$T_N = \max_{k=1\ldots F}\{N(d(C_{f_i},C_{f_{i+1}}))\}_k \quad (3.18)$$

where F is the number of frames of the video database. The T_N depends on the controlled environment, namely the distance of the camera to the scene, the orientation of the camera, the type and the position of the camera, lighting system, density of the crowd, period (working day, weekend, occasion, vacation, etc.), etc. The more is the distance of the camera to the scene, the less is the quantity of optical flows and blobs. In case of escalator, T_N also depends on the escalator type and position. Taking into account of these facts, we consider that we have at least one threshold by a controlled environment (video stream). If we have M controlled environments (video streams), which are the case in sites such as airport, shopping mall, bank, play ground, subway, concert, cinema hall, school, hospital, parking place, town center, political event, etc., then we select at least M thresholds. If the environment changes, then the threshold should be regenerated.

3.2.7 Experimental Results and Discussion

To conduct experiments, we used a set of real videos provided by cameras installed in an airport to monitor the situation of escalator exits. The videos were used to provide informative data for the security team who may need to take prompt actions in the event of a critical situation such as collapsing. The data sets are videos from a video surveillance system on escalator exits, taken in spanning days and seasons, provided by a visual surveillance company. Initially, the method has been tested with 13 different length video streams. Each video stream consists of normal and abnormal events. The normal situations correspond to crowd flows without collapsing in the escalator exits. Abnormal events correspond to videos that contain collapsing events mainly in escalator exists. Generally, in the videos we have two escalators corresponding to two-way-traffic of opposite directions. The original video frame size is 640×480 pixels. We extract nearly 1500 features per frame for detection and tracking.

For example, the left image in Fig.3.9 describes a scenario of a collapsing event in an escalator exit point. Some stuffs from a heavily loaded trolley have dropped just the exit point of the moving escalator which causes an abnormal situation on the exit point. The situation was successfully detected. The blue colored curve is the output of the proposed algorithm. Different video frames in normal and aberrant situation have been differentiated by a threshold label $T_N = 0.82$ (horizontal red line). The detection result has been compared with ground truth. Ground truth is the process of manually marking what an algorithm is expected to output. This simple algorithm does work in most of cases while in some cases it shows its shortcomings. Normally, it is enough efficacious for monitoring the crowd behaviour where the movement of crowd is in a linear direction like escalator.

However, the main noticeable shortcomings of the approach are listed below:

- It does not detect abnormal events in the crowd scenes which are a bit far from the camera e.g., the red marked activities of the right image in Fig.3.9. The most probable reason is that if the video sequences include abnormal events appear at far distance from the camera, the quantity of optical flow vectors is not sufficient to draw out abnormal frames.

- It does not handle overlapping situations. Of course, handling occlusion is a laborious work in optical flow as occluded pixels violate a major assumption of optical flow that each pixel goes somewhere. In theory, the pixels at the occlusion area should not be assigned any flow vector

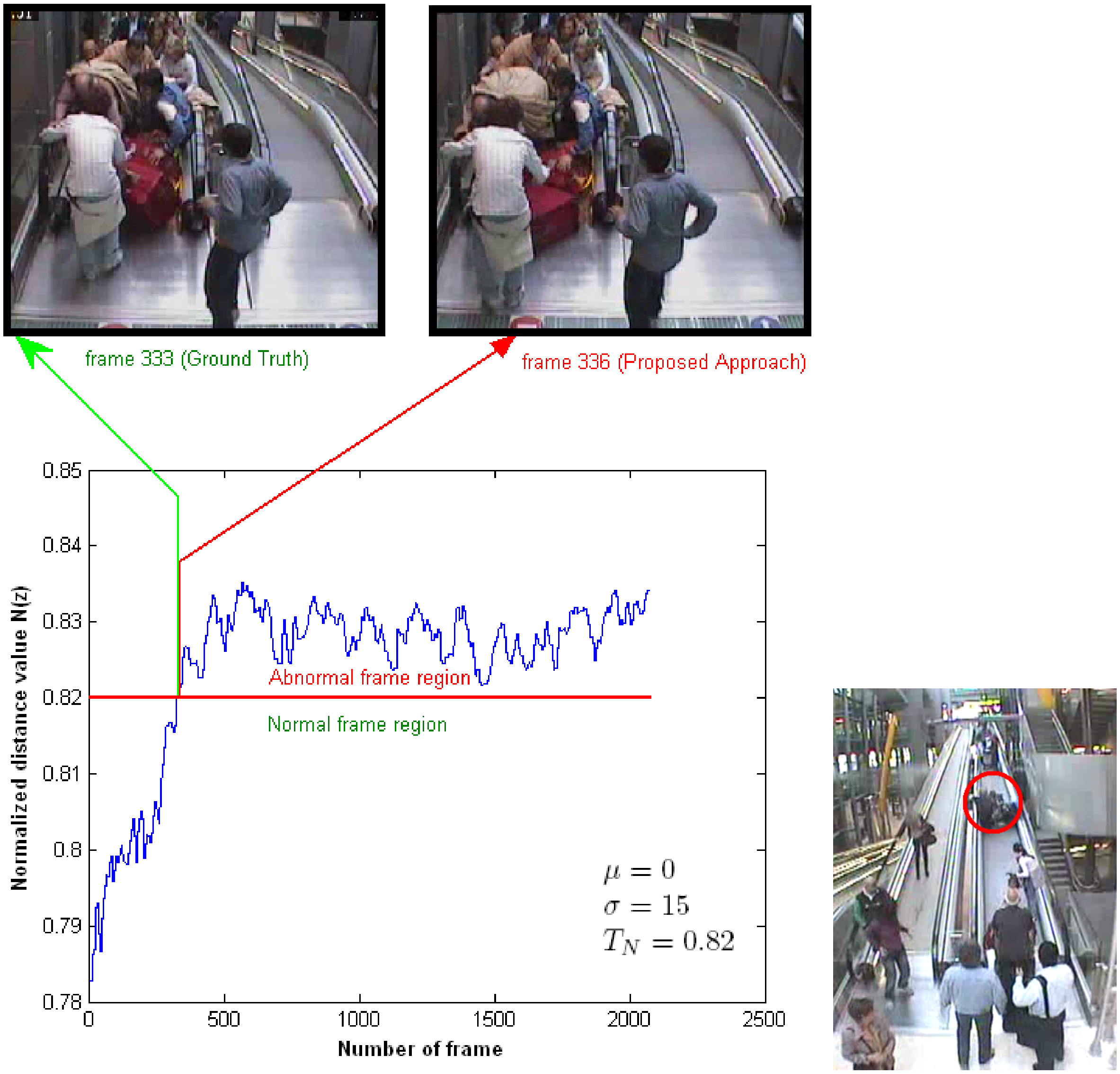


Figure 3.9: A heavily laden trolley drops some items off at the exit point of a escalator which creates an aberrant situation on the exit point. The event has been detected by the algorithm. Yet, the activities at the red circle in the right image cannot be detected by the Covariance matrix approach.

since there is no correspondence available in the other frame. Since the proposed approach is based on optical flow techniques, no flow vector can be got from occlusion areas. Consequently, the problem of occlusion is still unsolved.

- At implementation stage, it may suffer from different initial indexing problem of two video frames. The filtering process in the optical flow estimation can produce different indices of video frames. For example, if the index is to count from 0 and the filtering process allows that the features which move less than two pixels, then first video may produce index 0 which satisfies filtering condition and second video may consider index from 1 (or 2, or 3, or 4, etc.) which does not satisfies the condition. Suppose that the covariance approach is implemented on the basis of 0 indexing frame. Execution of the first frame of second video produces severe error as it has no 0 indexing frame and hence it has to face with **NaN** (*Not a Number*). During the covariance matrix construction (by Eq. 3.10) of the first frame $\mathbf{C_{f_1}}$ of this video, the variance of a **NaN** will be again a **NaN** as well as the covariance of two **NaN** elements will be considered as zero:

$$\mathbf{C_{f_1}} = \begin{pmatrix} \mathbf{NaN} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{NaN} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{NaN} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{NaN} \end{pmatrix}.$$

Assume that the next frame executes the following covariance matrix $\mathbf{C_{f_2}}$:

$$\mathbf{C_{f_2}} = \begin{pmatrix} \mathbf{0.176} & \mathbf{0.345} & \mathbf{0.457} & \mathbf{0.341} \\ \mathbf{0.345} & \mathbf{0.088} & \mathbf{0.702} & \mathbf{0.686} \\ \mathbf{0.457} & \mathbf{0.702} & \mathbf{0.001} & \mathbf{0.116} \\ \mathbf{0.341} & \mathbf{0.686} & \mathbf{0.116} & \mathbf{0.274} \end{pmatrix}.$$

The problem of transforming a regular matrix into a singular matrix is referred to as the eigenvalue problem. The generalized eigenvalue problem is to determine the nontrivial solutions of the equation of $\mathbf{C}_{\mathbf{f}_1}\mathbf{x} = \lambda \mathbf{C}_{\mathbf{f}_2}\mathbf{x}$ or more precisely:

$$\begin{pmatrix} \mathbf{NaN} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{NaN} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{NaN} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{NaN} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0.176} & \mathbf{0.345} & \mathbf{0.457} & \mathbf{0.341} \\ \mathbf{0.345} & \mathbf{0.088} & \mathbf{0.702} & \mathbf{0.686} \\ \mathbf{0.457} & \mathbf{0.702} & \mathbf{0.001} & \mathbf{0.116} \\ \mathbf{0.341} & \mathbf{0.686} & \mathbf{0.116} & \mathbf{0.274} \end{pmatrix} \mathbf{x} = \lambda \mathbf{x} \qquad (3.19)$$

where $\mathbf{x}$ is a length four column vector, and λ is a scalar. The values of λ which satisfy the above equation are the generalized eigenvalues and the corresponding values of $\mathbf{x}$ are the generalized right eigenvectors. But in the Eq. 3.19 any matrix must not contain any **NaN**. Consequently, there exists severe error. Possible solutions for this problem would include to convert *Not a Number* to *zero*, change the filtering process during optical flow estimation, etc.

3.3 Normalized Continuous Rank Increase Measure (NCRIM) Approach

3.3.1 Overview

Our approach (in [SD09a]) exposes in this section also detects exceptional motion frames from real videos irrespective of both static and dynamic backgrounds. The approach is based on the use of the *spatiotemporal region of interest (ST-RoI) features* obtained from ST-RoI, which is estimated using *motion history image* (MHI). Within ST-RoI, exceptional motion makes the motion vectors (e.g., directions) change a lot as compared to normal motion. The *normalized continuous rank-increase measure* (NCRIM) calculated from the ST-RoI features has been used as the judgement index for determining normal or exceptional motion frame. To demonstrate the interest of the proposed approach, the results based on the detection of exceptional motion frames in real videos (escalator dataset) obtained from a single camera placed on the escalator exit in an airport have been presented.

Objects detection and tracking have been used for large-scale surveillance camera systems. Exceptional motions are rare, difficult to describe, and hard to predict. Exceptional motion frames detection from the surveillance camera is becoming a very important research topic for next-generation security system that can detect emergencies and provide useful and informative surveillance. An approach for exceptional motion detection, that uses the probability distributions estimated by SVM, was introduced by [54]. Authors in [191] put forward a method using the covariance matrix of video stream features. The *cubic higher-order local auto-correlation* (CHLAC) features are used for event detection [105].

The CHLAC features are extracted by applying set CHLAC patterns to binary images which are obtained by using temporal deviation. So, the CHLAC features have information on the appearance of motion. Authors in [134] proposed a method combining the CHLAC features with a linear subspace method, which achieved a robust exceptional motion detection without using human detection and tracking. However, these conventional methods face a common problem when a background is dynamic or objects which are not targets exist in the images, it is very difficult to detect the exceptional motion. For instance, the method that uses CHLAC features and the linear subspace has difficulty for detecting motion, because any exceptional motion that moves in the same direction as the dynamic background will be buried in the feature space according to the properties of the CHLAC with the additivity of the features. To detect exceptional motion frames in real videos with both static and dynamic backgrounds, we focus on the following algorithmic steps: (i) estimation of *spatiotemporal region of interest* (ST-RoI) using *motion history image* (MHI), (ii) color segmentation of MHI, (iii) calculation of ST-RoI features, (iv) irregularity measure using *normalized continuous rank-increase measure* (NCRIM) (v) decision of normal/exceptional motion frame by comparing between the calculated NCRIM and a predefined threshold R_δ. The ST-RoI features contain information on the ST-RoI. The ST-RoI features are very similar of the space-time patch (ST-patch) features proposed by [159]. In general, exceptional motion, such as a person falling on the escalator, makes the motion vectors (directions) change a lot as compared to normal motion. So, we can use NCRIM which is calculated from the ST-RoI features as the judgement index for determining normal or exceptional motion. The NCRIM is considered as the measure of irregular motion vectors within ST-RoI. Consequently, we make exceptional motion frame detection possible by thresholding the processed value of NCRIM. Practical applications of our approach include the detection of real-time exceptional motion frames, which could lead to potentially dangerous situations in long escalators or narrow passages. Another possible application is to monitor the exceptional activity (e.g., opposite flow of vehicles in the same way) on the high ways, where vehicles primarily flow in unidirection.

3.3.2 Estimation of the Spatiotemporal Region of Interest (ST-RoI)

ST-RoI can be defined by considering the main motion region, where foreground subjects would primarily move unidirectionally. The ST-RoI can be built desirably from the accumulation of real-time computer vision representation of object movements, so called Motion History Image (MHI) presented by [46]. The MHI is a compact template representation of movement originally based on the layering of successive image motions. To generate MHI for the movement sequence, we layer successive silhouette images rather than image differences.

In an MHI H_τ, pixel intensity is a function of the temporal history of position or motion at that point. Based on time-stamping, a simple replacement and duration operator is used [22]:

$$H_\tau(x,y,t) = \begin{cases} \tau \ ifD(x,y,t) = 1 \\ max(0, H_\tau(x,y,t-1) - \delta) \ otherwise \end{cases} \tag{3.20}$$

The use of time-stamp allows for a more consistent port of the system between platforms where speeds may differ. System time is consistent during processing where frame rate is not. Thus time is explicitly encoded in the motion template. The above update function is called each time a new image is received and the corresponding silhouette image is formed [45]. The result of the function is a scalar-valued image where more recently moving pixels are brighter (see Fig.3.10). Explicitly, the Eq. 3.20 indicates that the MHI pixels where motion occurs are set to the current time-stamp τ, while the pixels where motion happened far ago are cleared. Now, if we wish to build a ST-RoI, we need to store the information of pixels where motion happened far ago so that the accumulation of object silhouettes in the motion template can yield useful motion information along the contours of the silhouette. Fig.3.11 makes noticeable an occurrence of the obtained ST-RoI for an escalator case. The results are very significant and desirable when the video duration is very long. The use of ST-RoI ameliorates the quality of the results and reduces processing time significantly which is an important factor for real-time applications. To reduce both static and dynamic noises in a great amount we segment the MHI inside of the ST-RoI considering RGB color channels. The resulting color segments (silhouetted contours) are suitable for allocating points of interest inside ST-RoI. We consider Harris corner detector as a point of interest. This detector is a famous corner detector due to its strong invariance to rotation, scale, illumination variation, and image noise [154]. It is based on the local auto-correlation function of a signal, where the local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions. However, since the approach considers color silhouetted region, it minimizes the number of unnecessary corners caused by e.g., a person wears/carries grid-dress-like cloth/stuffs within ST-RoI, there will be too many corners detected from the his/her region.

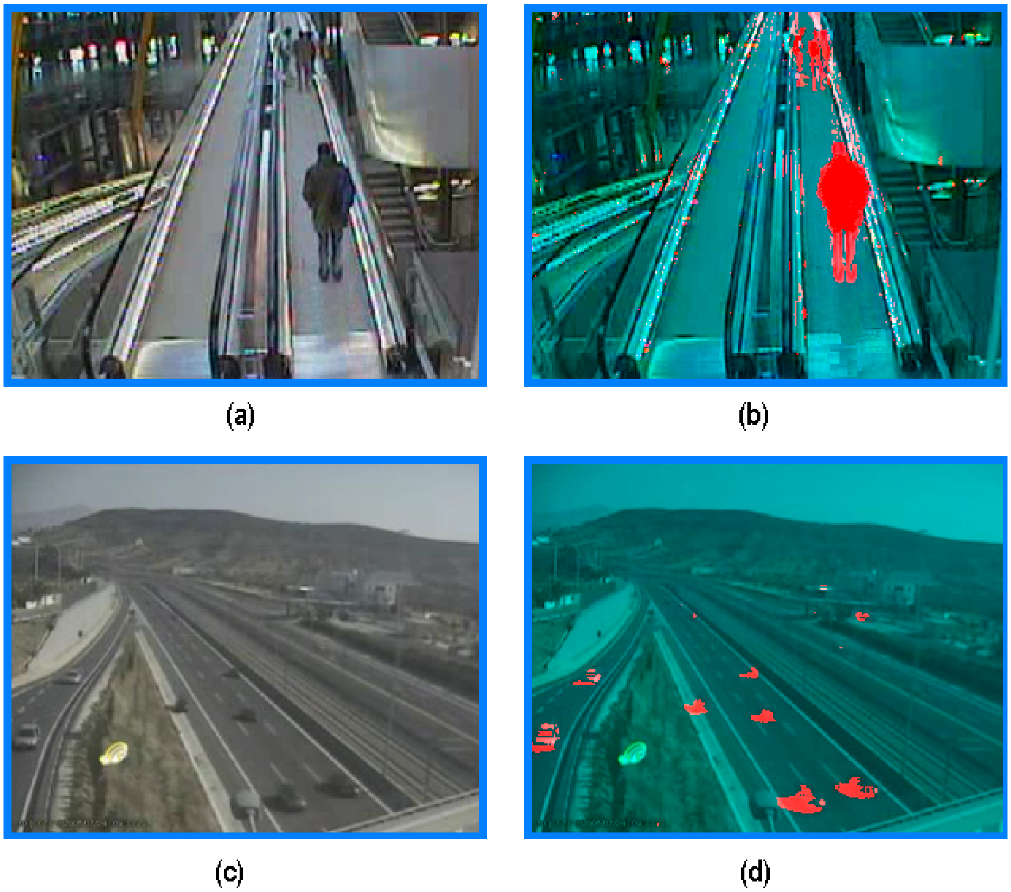


Figure 3.10: (a) & (c) depict camera views with dynamic & static backgrounds respectively; (b) & (d) represent *motion mask* where only current silhouette motion has been colored as red.

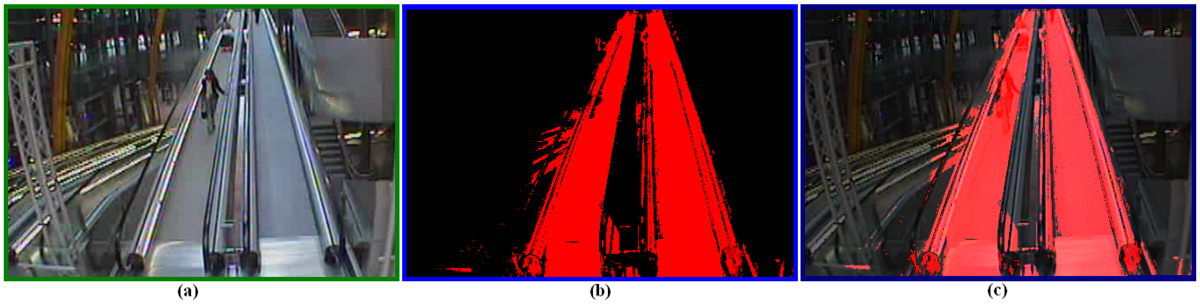


Figure 3.11: (a) camera view, (b) the red colored region represents *Spatiotemporal Region of Interest* (ST-RoI) or *Motion Map* (MM), (c) masked view.

3.3.3 Calculation of ST-RoI features

Once we define n (say 2000) corners on the ST-RoI, we track those over the next ST-RoIs using the feature tracker of Kanade-Lucas-Shi-Tomasi [123, 160]. After matching features between frames, the result is a matrix $\mathbf{G_{n\times3}}$ formulated by:

$$\mathbf{G_{n\times3}} = \begin{bmatrix} M_{x_1} & M_{y_1} & M_{\alpha_1} \\ M_{x_2} & M_{y_2} & M_{\alpha_2} \\ M_{x_3} & M_{y_3} & M_{\alpha_3} \\ . & . & . \\ . & . & . \\ M_{x_i} & M_{y_i} & M_{\alpha_i} \\ . & . & . \\ . & . & . \\ M_{x_n} & M_{y_n} & M_{\alpha_n} \end{bmatrix}_{n\times3} \tag{3.21}$$

where

- $M_{x_i} \mapsto x$ coordinate of any feature element i (where $i \in n$),
- $M_{y_i} \mapsto y$ coordinate of the i,
- $M_{\alpha_i} \mapsto$ moving direction α of the i.

Let constant optical flow within the ST-RoI, then the optical flow within the ST-RoI can be estimated by solving the following 3-dimensional scatter matrix F as:

$$[\mathbf{F}]_{3\times3} \times \begin{bmatrix} x \\ y \\ \alpha \end{bmatrix}_{3\times1} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{3\times1} \tag{3.22}$$

where $\mathbf{F} = \mathbf{G^T G}$, which can be obtained by multiplying both sides of Eq. 3.21 by $\mathbf{G^T}$ (the transpose of $\mathbf{G}$), more explicitly:

$$\mathbf{F} = \begin{bmatrix} \sum_{i=1}^{n} M_{x_i}^2 & \sum_{i=1}^{n} M_{x_i} M_{y_i} & \sum_{i=1}^{n} M_{x_i} M_{\alpha_i} \\ \sum_{i=1}^{n} M_{y_i} M_{x_i} & \sum_{i=1}^{n} M_{y_i}^2 & \sum_{i=1}^{n} M_{y_i} M_{\alpha_i} \\ \sum_{i=1}^{n} M_{\alpha_i} M_{x_i} & \sum_{i=1}^{n} M_{\alpha_i} M_{y_i} & \sum_{i=1}^{n} M_{\alpha_i}^2 \end{bmatrix} \tag{3.23}$$

where the 3×3 matrix $\mathbf{F}$ belongs to the *spatiotemporal region of interest (ST-RoI) features*. Thus $\mathbf{F}$ contains information on both the *appearance* and *motion direction* simultaneously.

3.3.4 Irregularity measure using Normalized Continuous Rank Increase Measure

In general, Eq. 3.22 is a set of linear equations. Therefore, if the optical flows are constant within the ST-RoI, there will be a non-zero solution of Eq. 3.22. As a result, the 3×3 coefficient matrix $\mathbf{F}$ should be rank deficient, i.e., $rank(\mathbf{F}) \leq 2$. Explicitly, if $\mathbf{F}$ is not rank deficient, i.e., its smallest eigenvalue λ_{min} is not equal to zero ($\lambda_{min}(\mathbf{F}) \neq 0$), then ST-RoI can not be motion consistent. The matrix $\mathbf{F}$ contains information about the appearance and motion of the ST-RoI. Consequently, the rank of the coefficient matrix $\mathbf{F}$ can be used to analyze the brightness distribution and motion types within the ST-RoI. In case of $rank(\mathbf{F}) = 3$, there will be multiple motions within the ST-RoI. A distributed spatial brightness structure moves at a constant motion when $rank(\mathbf{F}) = 2$. If we examine all possible ranks of the 3D structural tensor $\mathbf{F}$, which contains *only uniform motion*, then to come to know the spatial properties of the ST-RoI we could consider the upper left minor $\mathbf{M_{xy}}$ of the tensor $\mathbf{F}$ as:

$$\mathbf{M_{xy}} = \begin{bmatrix} \sum_{i=1}^{n} M_{x_i}^2 & \sum_{i=1}^{n} M_{x_i} M_{y_i} \\ \sum_{i=1}^{n} M_{y_i} M_{x_i} & \sum_{i=1}^{n} M_{y_i}^2 \end{bmatrix}. \tag{3.24}$$

The symmetric matrix $\mathbf{M_{xy}}$ captures the intensity structure of the ST-RoI. The common properties of this symmetric matrix are: (i) real eigenvalues, (ii) real eigenvectors, and (iii) orthogonal eigenvectors. Let us assume that ψ_1 and ψ_2 are the eigenvalues of matrix $\mathbf{M_{xy}}$. The eigenvalues form a rotationally invariant description. When there is *only uniform motion* within the ST-RoI, the added temporal component at the third row and column does not introduce any increase in rank. This condition satisfies $rank(\mathbf{F}) = rank(\mathbf{M_{xy}})$. However, condition does not satisfy when the motion is not along a single straight line. In such cases, the added temporal component introduces an increase in the rank, $rank(\mathbf{F}) = rank(\mathbf{M_{xy}}) + 1$. The difference in rank can not be more than 1, because only one column/row is added in the transition from $\mathbf{M_{xy}}$ to $\mathbf{F}$. Then measuring the rank-increase δ_r between $\mathbf{F}$ and $\mathbf{M_{xy}}$ reveals whether the ST-RoI contains a single or multiple motions:

$$\delta_r = rank(\mathbf{F}) - rank(\mathbf{M_{xy}}) = \begin{cases} 0 & if\ single\ motion \\ 1 & if\ multiple\ motions. \end{cases}$$

Simple way to estimate the rank-increase from $rank(\mathbf{M_{xy}})$ to F is to compute their individual ranks and then take the difference, which provides either 0 or 1. The rank of a matrix is determined by the number of nonzero eigenvalues it has. Notwithstanding, in the presence of noise, eigenvalues are never zero. Applying a threshold to the eigenvalues is mainly data dependent, and a wrong choice of a threshold would lead to wrong rank values. If two motions are very similar but not identical - are they consistent or not? Therefore, a normalized and continuous measure is needed to quantify the matrix deficiency. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and $\psi_1 \geq \psi_2$ be the eigenvalues of $\mathbf{F}$ and $\mathbf{M_{xy}}$, respectively. From the interlacing property of eigenvalues in symmetric matrices [71], it follows that $\lambda_1 \geq \psi_1 \geq \lambda_2 \geq \psi_2 \geq \lambda_3$ which yields the following interest bearing observations:

$$\lambda_1 \geq \frac{\lambda_1 \times \lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} = \frac{det(\mathbf{F})}{det(\mathbf{M_{xy}})} \geq \lambda_3 \tag{3.25}$$

$$1 \geq \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} \geq \frac{\lambda_3}{\lambda_1} \geq 0 \tag{3.26}$$

$$1 \geq \delta_r \geq \frac{\lambda_3}{\lambda_1} \geq 0 \tag{3.27}$$

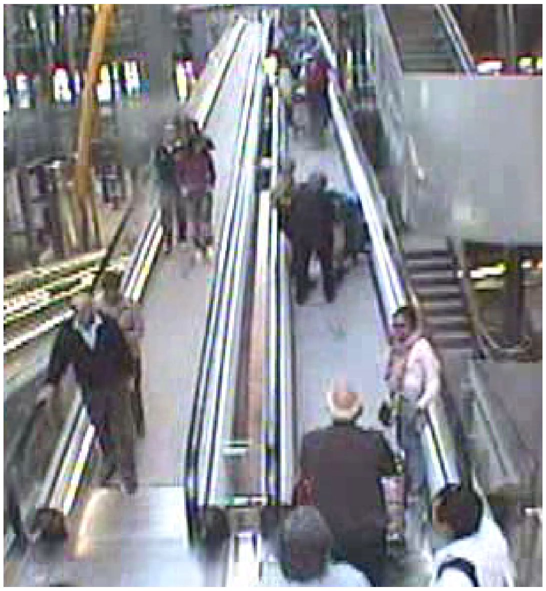
where the *Normalized Continuous Rank Increase Measur*e (NCRIM) δ_r follows $0 \leq \delta_r \leq 1$ as well as

$$\delta_r = \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2}. \tag{3.28}$$

The case of $\delta_r = 0$ is an ideal case of no rank increase, and when $\delta_r = 1$ there is a clear rank-increase. Nevertheless, the δ_r allows to handle noisy data and provides varying degrees of rank-increases for varying degrees of motion-consistencies. It is easy to show that the term $\frac{det(\mathbf{F})}{det(\mathbf{M_{xy}})}$ is the *pure temporal* eigenvalue that was derived in [119] using a Galilean diagonalization of the matrix F. This diagonalization compensates for the local constant velocity and the pure temporal eigenvalue encodes information about the non-linearity of the local motion [119]. Fig.3.12 depicts an instance of the *normalized continuous rank increase measure* δ_r calculation during normal and abnormal behavior. The estimated values suit to distinguish between normal and abnormal events.

3.3.5 Decision of normal or abnormal motion frames

A predefined threshold R_δ value, calculated from large videos that contain exclusively normal motions, can differentiate each frame with respect to its assigned δ_r value whether its motion is normal or ex-



Normal Behavior : camera view frame

$$\mathbf{F} = \begin{bmatrix} 75498 & 5278671 & -495495 \\ 5278671 & 664134 & -617319 \\ -495495 & -617319 & 24392 \end{bmatrix}$$

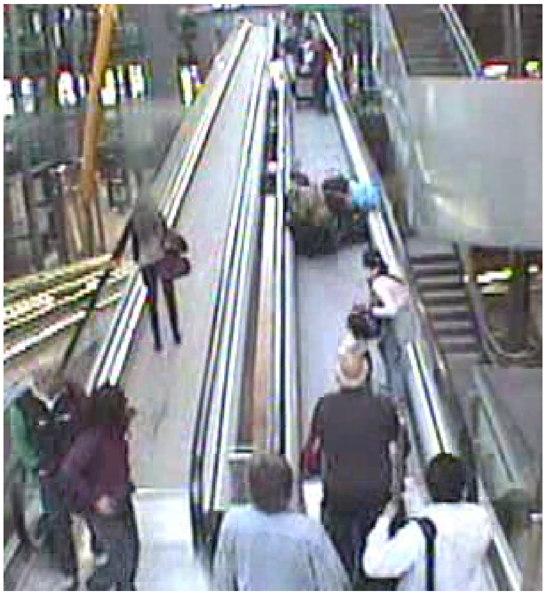
$$\lambda_1 = -4917900,\ \lambda_2 = -83203,\ \lambda_3 = 5765100$$

$$\mathbf{M_{xy}} = \begin{bmatrix} 75498 & 5278671 \\ 5278671 & 664134 \end{bmatrix}$$

$$\psi_1 = -4917100,\ \psi_2 = 5656700$$

$$\delta_{\mathrm{r}} = \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} = \frac{-83203 \times 5765100}{-4917100 \times 5656700} = 0.0172$$

Calculation of *NCRIM*



Abnormal Behavior : camera view frame

$$\mathbf{F} = \begin{bmatrix} 987447 & 4275963 & -816385 \\ 4275963 & 919281 & -920319 \\ -816385 & -920319 & 52979 \end{bmatrix}$$

$$\lambda_1 = -3324700,\ \lambda_2 = -221560,\ \lambda_3 = 5505900$$

$$\mathbf{M_{xy}} = \begin{bmatrix} 987447 & 4275963 \\ 4275963 & 919281 \end{bmatrix}$$

$$\psi_1 = -3322700,\ \psi_2 = 5229500$$

$$\delta_{\mathrm{r}} = \frac{\lambda_2 \times \lambda_3}{\psi_1 \times \psi_2} = \frac{-221560 \times 5505900}{-3322700 \times 5229500} = 0.0702$$

Calculation of *NCRIM*

Figure 3.12: Normal and abnormal behaviors of crowd. Calculation of the *normalized continuous rank increase measure* δ_r in both cases. Abnormal situation caused when the heavily loaded trolley suddenly became unbalanced and hit two accompanied age-old persons. Consequently, they were forced down on the opposite direction of the moving escalator along with their belongings.

ceptional. Any frame having value of δ_r which is greater than the R_δ will be considered as exceptional motion frame. The R_δ depends on the controlled environment namely the distance of the camera to the scene, the orientation of the camera, the type and the position of the camera, lighting system, density of the crowd, etc. If we have $\mathscr{N}$ video streams, then we select at least $\mathscr{N}$ thresholds. If the environment changes, then R_δ should be regenerated as:

$$R_\delta = \max_{i=1\ldots f}\{\delta_r\}_i + \min_{i=1\ldots f}\left[\frac{1}{(2\pi)^2}\sum_{k=0}^{\infty}\frac{(-1)^k(\delta_r)^{2k+1}}{k!(2k+1)}\right]_i$$

where f is the number of frames of the video database.

3.3.6 Experimental Results and Discussion

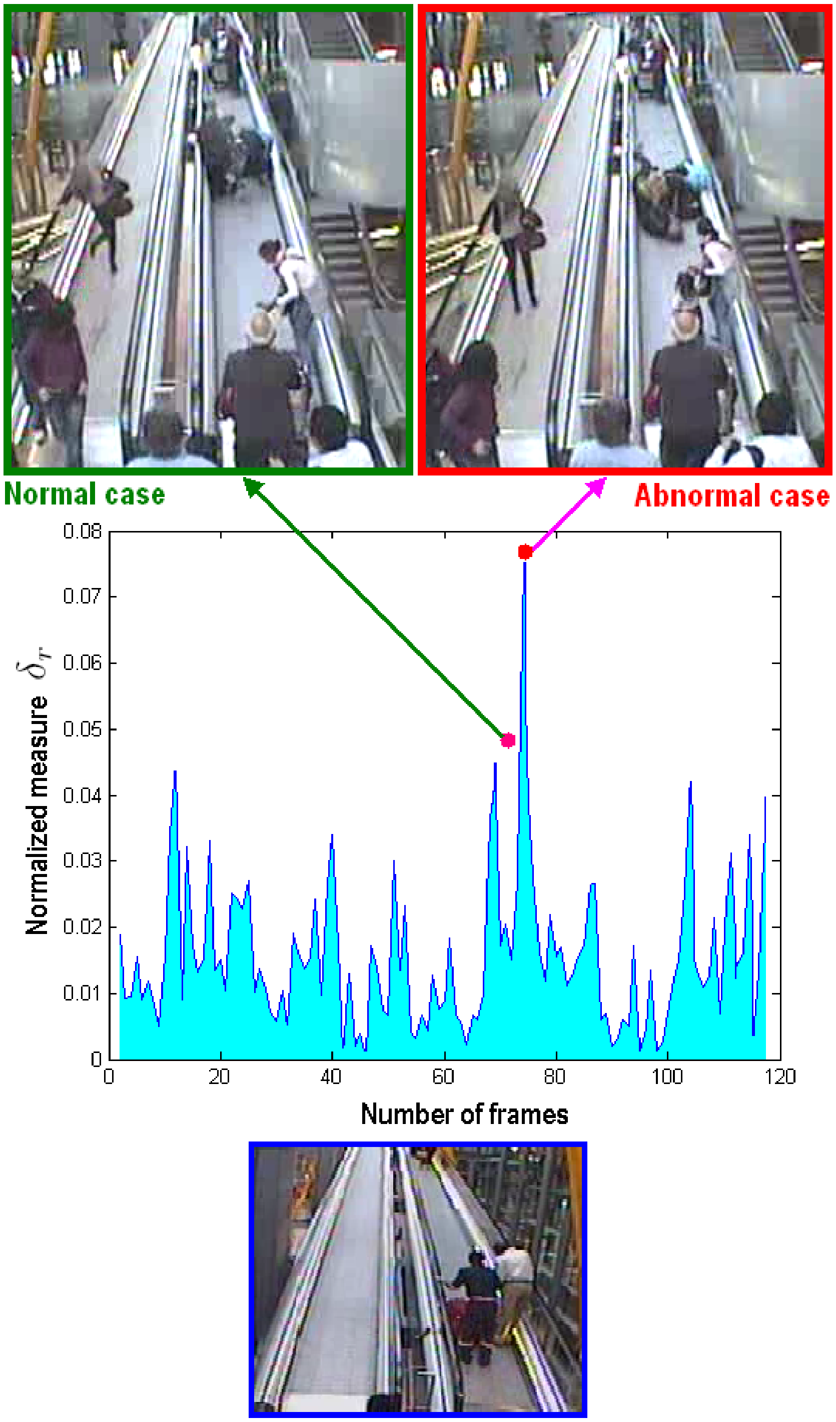
To conduct the experiment, we used mainly, escalator data-set, unidirectional motion videos of frame size 640×480 pixels, where both normal and exceptional movements exist. All images were in color, contained some amounts of noise, strong local variations in illumination, shadows and reflections, large numbers of objects often partially occluded, etc.

Fig.3.13 describes a scenario of exceptional motions which happens nearly in the middle of the escalator. A heavily laden trolley suddenly became unbalanced and hit two accompanied persons. Finally, they were forced down on the opposite direction of the moving escalator along with their items. The proposed method can detect successfully the accidental circumstances by detecting aberrant motion frames. The detection results have been compared with ground truth, which is the process of manually marking what an algorithm is expected to output. However, the algorithm has seldom effect on the video stream like Fig.3.13 (a). This is due to the fact that the video sequences include abnormal events occur with occlusion. Hence, the quantity of extracted optical flow vectors is not sufficient to draw out abnormal frames. Occlusion handling is a very difficult part of optical flow as occluded pixels violate its major assumption.

3.4 Mahalanobis Metric Approach

3.4.1 Overview

In this section, we have presented another approach ([SD09c]), which detects abnormal events in surveillance video systems (e.g., escalators, narrow passages, etc.), based on optical flow analysis of



(a) Overlapped Exceptional Frame

Figure 3.13: The peaked curve depicts exceptional motion frames (e.g., red marked frame), nevertheless the NCRIM approach cannot detect events e.g., (a).

crowd behavior followed by *Mahalanobis* and χ^2 metrics. The *Mahalanobis distance* is a *metric*, i.e., it satisfies metric conditions: (i) non-negativity, (ii) identity of indiscernible, (iii) symmetry or commutativity, and (iv) triangle inequality. Mahalanobis metric uses an appropriate correlation matrix to take account of differences in variable variances and correlations between variables. The video frames are flagged as *normal* or *eccentric* (abnormal) established on the statistical classification of the distribution of Mahalanobis distances of the normalized spatiotemporal information of optical flow vectors. Those optical flow vectors are computed from the small blocks of the specific region of successive frames namely *region of interest image* (RII), which is discovered by *region of interest image map* (RIIM) (e.g., Fig.3.14). The RIIM is obtained from specific treatment of foreground segmentation of moving subjects. Like *motion heat map* (e.g., Fig.3.6) or *motion map* (e.g., Fig.3.11), the use of RIIM improves the quality of the results and reduces processing time.

The approach primarily has been tested against a single camera data-set, *Escalator dataset* [132], collected by installing the camera on the escalator egresses in an airport as well as the data-set of Minnesota University so-called *UMN dataset* [137].

3.4.2 RIIM and Feature Extraction

3.4.2.1 Region of Interest Image Map (RIIM)

The RIIM can be defined automatically by building a color histogram [see Fig.3.14 (a) & (b) for escalator case], which is built from the accumulation of binary blobs of moving subjects, which were extracted following foreground segmentation method [102]. The adaptive background subtraction algorithm proposed by [102] is able to model a background from a long training sequence with limited memory, works well on moving backgrounds, illumination changes, and compressed videos having irregular intensity distributions. The RIIM will be brought into existence mainly off-line. On-line is possible but it makes the system complicated. Off-line is better as the generated RIIM will be more significant and accurate when the video duration will be very long. RIIM improves the quality of the results and reduces processing time which is an imperative factor for real-time applications.

3.4.2.2 Spatiotemporal Information (ST-Info) Extraction

The region of interest image (RII), ascertained by RIIM, is separated into small blocks. Once we define *n* (say 1500) points of interest in the RII, we track those points over the small blocks of two successive region of interest images using the combination feature tracker of Kanade-Lucas-Shi-Tomasi [123, 160]

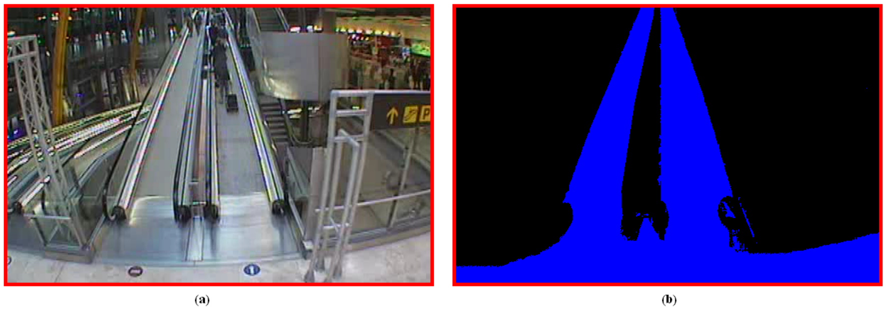


Figure 3.14: (a) Camera view. (b) Generated *Region of Interest Image Map (RIIM)* and blue region on the RIIM recommends *Region of Interest Image* (RII).

easily. But one encountered problem is that people near the camera are supposed to generate large optical flow vectors and people far from the camera cannot generate such flow vectors even when they would make very quick motion (e.g., running or falling). In order to get an acceptable distribution of optical flow pattern over the RII, people near or far from the camera should be fairly treated. To solve this problem, we take into account vertical coordinate of each block. Consequently, a weighing coefficient λ is calculated according to the vertical coordinate of the block. A block far away from the camera has small vertical coordinate, as a result its λ should be large. Equally, block with large vertical coordinate get smaller λ. The value of λ heavily depends on the context of application and implementation. However, for our escalator videos data-set typically λ limits $0.6 \leq \lambda \leq 1$. Adjacent to camera (starting of the RII) region the value of $\lambda = 0.6$ is appropriate, whereas λ bears the maximum value 1 at the end of part of the RII. We also take down the static and noise features. Static features are the features which moves less than two pixels. Noise features are the isolated features which have a big angle and distance difference with their near neighbors due to tracking calculation errors. Finally, for each frame irrespective of normal or eccentric events, we obtain an acceptable and workable spatiotemporal information, i.e., a $n \times 5$ matrix which is a function of time, explicitly speaking a set of vectors $\mathbf{M}(\mathbf{j})(\mathbf{k})$ of n elements as defined:

$$\mathbf{M}(\mathbf{j})(\mathbf{k}) = \begin{bmatrix} x(1)(1) & x(2)(1) & x(3)(1) & x(4)(1) & x(5)(1) \\ . & . & . & . & . \\ x(1)(i) & x(2)(i) & x(3)(i) & x(4)(i) & x(5)(i) \\ . & . & . & . & . \\ x(1)(n) & x(2)(n) & x(3)(n) & x(4)(n) & x(5)(n) \end{bmatrix} \tag{3.29}$$

where $j = 1, 2, \ldots 5$, $k = 1, 2, 3, \ldots n$, i be any feature element in k, and

- $x(1)(i) \mapsto x$ coordinate of the i,
- $x(2)(i) \mapsto y$ coordinate of the i,
- $x(3)(i) \mapsto x$ velocity with multiply by λ_i of the i,
- $x(4)(i) \mapsto y$ velocity with multiply by λ_i of the i,
- $x(5)(i) \mapsto$ moving direction of the i.

3.4.3 Statistical treatments of the spatiotemporal information

3.4.3.1 Normalization of Raw Data

A normalized value is a value that has been processed in a way that makes it possible to be efficiently compared against other values. For each column of $\mathbf{M}(\mathbf{j})(\mathbf{k})$, we calculate the *average* $\overline{x_j}$ and *standard deviation* σ_j. Subtracting the average $\overline{x_j}$ from each value in the columns of $x(j)(k)$, and then dividing by the standard deviation σ_j for that column in $x(j)(k)$ generated a new matrix $z(j)(k)$ as:

$$\overline{x_j} = \frac{1}{n} \sum_{k=1}^{n} x(j)(k) \tag{3.30}$$

$$\sigma_j = \sqrt{\frac{\sum (x(j)(k) - \overline{x_j})^2}{n-1}} \tag{3.31}$$

$$z(j)(k) = \frac{x(j)(k) - \overline{x_j}}{\sigma_j}. \tag{3.32}$$

All values in $z(j)(k)$ are *dimensionless* and *normalized*, hence the new form of $\mathbf{M}(\mathbf{j})(\mathbf{k})$ yields:

$$\mathbf{Z}(\mathbf{j})(\mathbf{k}) = \begin{bmatrix} z(1)(1) & z(2)(1) & z(3)(1) & z(4)(1) & z(5)(1) \\ . & . & . & . & . \\ z(1)(i) & z(2)(i) & z(3)(i) & z(4)(i) & z(5)(i) \\ . & . & . & . & . \\ z(1)(n) & z(2)(n) & z(3)(n) & z(4)(n) & z(5)(n) \end{bmatrix}. \tag{3.33}$$

3.4.3.2 Calculation of Correlation Matrix

A covariance matrix is merely collection of several variance-covariances in the form of a square matrix. Due to the symmetry property of covariances, it is necessarily a symmetric matrix. For a real symmetric matrix all the eigenvalues and eigenvectors are real. A $n \times n$ symmetric matrix satisfies the following: (i) it has exactly n (not necessarily distinct) eigenvalues, (ii) there exists a set of n eigenvectors, one for each eigenvalue, that are mutually orthogonal. Clearly, a covariance (symmetric) matrix has n eigenvalues and there exist n linearly independent eigenvectors (because of orthogonality) even if the eigenvalues are not distinct. However, one problem with covariance is that it is sensitive to the scales. We would like a measure of the strength of the link between two components of covariance that does not depend on the units used to measure these quantities. To obtain a more direct indication of how two components co-vary, we scale covariance to obtain correlation. Correlation is dimensionless while covariation is in units obtained by multiplying the units of each variable. Using $\mathbf{Z}(\mathbf{j})(\mathbf{k})$, scaling is performed by means of the following equations:

$$r_{pq} = \frac{S_{pq}}{S_p S_q} \tag{3.34}$$

$$S_{pq} = \frac{1}{n-1} \sum_{k=1}^{n} [z_p(k) z_q(k)] \tag{3.35}$$

$$S_l = \sqrt{\frac{1}{n-1} \sum_{k=1}^{n} [z_l(k)^2]} \tag{3.36}$$

where $\{p,q\} \in j$ and $l \in \{p,q\}$.

3.4.3.3 Calculation of Mahalanobis Distance $D_m(i)$

Distance metric is a key issue in many computer vision algorithms. In statistics, Mahalanobis distance is a distance measure introduced by Mahalanobis [126]. It is based on correlations between variables by which different patterns can be identified and analyzed. It is a useful way of determining similarity of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant, i.e., not dependent on the scale of measurements. The region of constant Mahalanobis distance around the mean forms an ellipse in two dimensional space (i.e., when only 2 variables are measured), or an ellipsoid or hyperellipsoid when more variables are used. The Mahalanobis distance is the same as the Euclidean distance if the correlation matrix is the identity matrix. We calculate the Mahalanobis distance $D_m(i)$ for each row of the normalized matrix $\mathbf{Z}(\mathbf{j})(\mathbf{k})$ by multiplying the row by the *inverted correlation matrix*, then multiplying the resulting vector by the transpose of the row of the $\mathbf{Z}(\mathbf{j})(\mathbf{k})$, then dividing the obtained result by the degree of freedom, finally grasping square root of the up-to-the-minute result as:

$$\mathbf{D_m(i)} = \sqrt{\left[\frac{\mathbf{z(1)(i)\ z(2)(i)\ z(3)(i)\ z(4)(i)\ z(5)(i)}}{\mathbf{5}}\right] \begin{bmatrix} \mathbf{1} & \mathbf{r_{12}} & \mathbf{r_{13}} & \mathbf{r_{14}} & \mathbf{r_{15}} \\ \mathbf{r_{21}} & \mathbf{1} & \mathbf{r_{23}} & \mathbf{r_{24}} & \mathbf{r_{25}} \\ \mathbf{r_{31}} & \mathbf{r_{32}} & \mathbf{1} & \mathbf{r_{34}} & \mathbf{r_{35}} \\ \mathbf{r_{41}} & \mathbf{r_{42}} & \mathbf{r_{43}} & \mathbf{1} & \mathbf{r_{45}} \\ \mathbf{r_{51}} & \mathbf{r_{52}} & \mathbf{r_{53}} & \mathbf{r_{54}} & \mathbf{1} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{z(1)(i)} \\ \mathbf{z(2)(i)} \\ \mathbf{z(3)(i)} \\ \mathbf{z(4)(i)} \\ \mathbf{z(5)(i)} \end{bmatrix}} \tag{3.37}$$

where the number of columns contained in $\mathbf{Z}(\mathbf{j})(\mathbf{k})$ is referred to as the degree of freedom which is 5 in this case. The diagonal **1**s indicate that a random variable co-varies perfectly with itself (auto-correlation), and the off diagonal terms contain the correlation between two components. Like covariance matrices, correlation matrices must be positive definite or positive semi-definite. Exactly same as covariance matrix, there is a handy simple little formula $\frac{\tau\times(\tau-1)}{2}$ that tells how many pairs (e.g., correlations) there are for τ number of variables.

Geometrically, samples with an equal $D_m(i)$ lie on an ellipsoid (Mahalanobis Space). The $D_m(i)$ is small for samples lying on or close to the principal axis of the ellipsoid. Samples further away from the principal axis have a much higher $D_m(i)$. The larger the $D_m(i)$ for a sample is, the more likely the sample is an outlier. An outlier (extreme sample) is a sample that is very different from the average sample in the data set. An outlier may be an ordinary sample, but of which at least one attribute has been severely corrupted by a mistake or error (e.g., tracking calculation errors). An outlier may also be a bona fide sample, that simply turns out to be exceptional. Since Mahalanobis distance satisfies the conditions (symmetry, positivity, triangle inequality) of metric, it is a metric. The use of the Mahalanobis metric removes several limitations of the Euclidean metric, e.g.,

- It automatically accounts for the scaling of the coordinate axes;
- It makes improvements for correlation between the different features;
- It can provide curved as well as linear decision boundaries.

Nonetheless, there is a disbursement to be paid for those advantages. The computation of the correlation matrix can give rise to problems. When the investigated data are measured over a large number of variables, they can keep under control much redundant or correlated information. This is so-called *multicollinearity* in the data which leads to a singular correlation matrix that cannot be inverted. Another precinct for the calculation of the correlation matrix is that the number of samples in the data set has to be larger than the number of variables. Yet, in the proposed approach, both problems have been minimized by dint of 5 variables and tracking about 1500 samples (points of interest) in each frame, respectively.

3.4.4 Analysis of Mahalanobis Distances

Mahalanobis distance is a *metric* (a rule for calculating the distance between two points) which is better adapted than the usual Euclidian metric to settings involving non-spherically symmetric distributions.

3.4.4.1 Classification of Mahalanobis Distances

Mahalanobis squared distances are calculated in units of standard deviation from the group mean. Therefore, the calculated circumscribing ellipse formed around the samples actually defines the one standard deviation of that group. This allows the designing of a statistical probability to that measurement. In theory, Mahalanobis squared distance is distributed as a χ^2 distribution with degree of freedom equal to the number of independent variables in the analysis. The χ^2 distribution is very important because many test statistics are approximately distributed as χ^2. However, the χ^2 distribution has only one parameter called the degree of freedom. The shape of a χ^2 distribution curve is skewed for very small degrees of freedom and it changes drastically as the degrees of freedom increase. Eventually, for large degrees of freedom, the χ^2 distribution curve looks like a *normal distribution* curve. Like all other continuous distribution curves, the total area under a χ^2 distribution curve is 1.0. The *three sigma rule*, or *68-95-99.7 rule*, or *empirical rule*, states that for a normal distribution, about 68%, 95%, 99.7% of the values lie within 1, 2, and 3 standard deviation of the mean, respectively. Clearly, almost all values lie within 3 standard deviations of the mean. Consequently, samples that have a squared Mahalanobis distance larger than 3 have a probability less than 0.01. These samples can be classified as members of *non-member group*. Samples those have squared Mahalanobis distances less than 3 are then classified as members of *member group*. The determination of the threshold depends on the application and the type of samples. In the proposed approach, we settle that each $D_m(i)$ goes either *member group* or *non-member group*. Sample with a higher $D_m(i)$ than $\sqrt{3}$ is treated as *non-member group*, otherwise *member group*. *Member group* contains absolutely the samples of a normal event, whereas *non-member group* contains essentially samples of eccentric events (including outliers). Fig.3.15 depicts, while Mahalanobis metric produces elliptical cluster where samples are well correlated, Euclidean metric produces circular subsets. The *non-member group* consists of samples s_1, s_2, s_3, s_4, s_5, s_6, s_7, and the outlier s_8, while the *member group* groups the rest samples. Presuming in any *non-member group*, having M samples including outliers (where also assuming that in general $M \gg outliers$ satisfies), we sum up their Mahalanobis distances, S_d, with the help of:

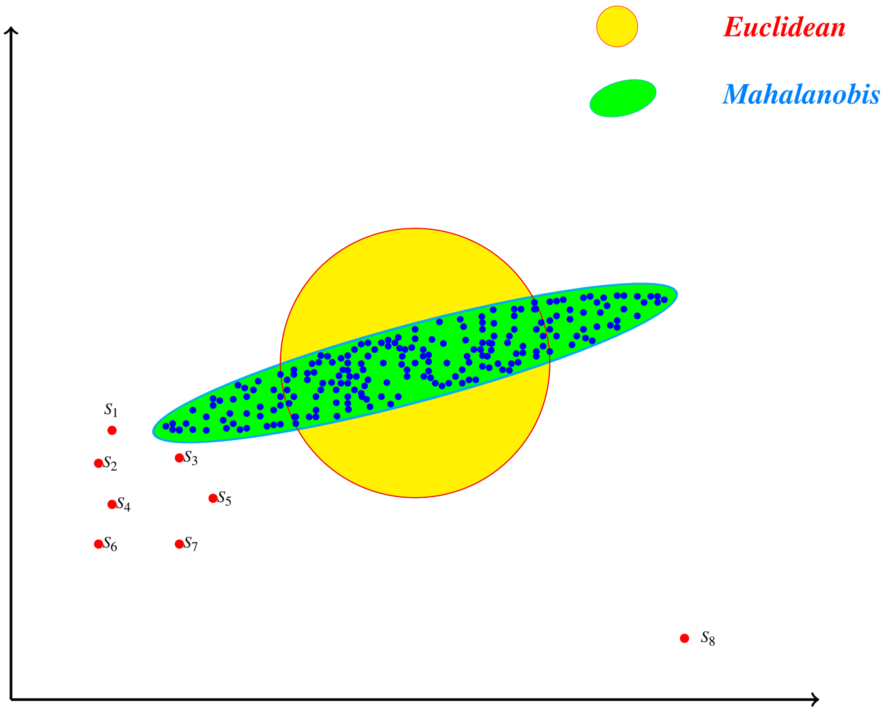


Figure 3.15: Mahalanobis metric with respect to Euclidean metric.

$$S_d = \sum_{i=1}^{M} D_m(i). \tag{3.38}$$

3.4.4.2 Normalization of S_d

Now, we transfer each S_d into a normalized distance (probability) value ranges between 0 and 1. The normalization may be done by using the simple formula like $1/log(S_d)$, but the normalized values fall into a congested range (scaling problem) which will arise problem specially in threshold selection. To solve the scaling problem, we take the advantage of cumulative distribution function (cdf), which has strict lower and upper bounds between 0 and 1, we can easily pick up the normalized distance of each S_d. Since all values of S_d are skewed to the right (positive-definite) and their variances are also large, we can use *Log-normal* distribution. Skewed distributions are particularly common when mean values are low, variances large, and values cannot be negative. Log-normal distributions are usually characterized in terms of the log-transformed variable, using as parameters the expected value, or mean (*location* parameter μ), of its distribution, and the standard deviation (*scale* parameter σ). The σ is entitled as *scale* as its value determines the *scale* or statistical dispersion of the probability distribution. If N_d be the normalized value of S_d, then N_d can be gently estimated by means of:

$$N_d = \frac{1}{2}\left[1 + erf\{\frac{log(S_d) - \mu}{\sigma\sqrt{2}}\}\right], erf(r) = \frac{2}{\sqrt{\pi}}\left[r - \frac{r^3}{3} + \frac{r^5}{10} - \frac{r^7}{42} + \ldots\right] \qquad (3.39)$$

where erf is a $Gauss\ error\ function$ and $r = \frac{log(S_d) - \mu}{\sigma\sqrt{2}}$. Using Eq. 3.39, and placing congenial values of μ & σ (say $\mu = 0$, $\sigma = 5$) we can explicitly estimate the value of N_d between 0 and 1. Now, it is important to define an appropriate threshold T_d to make a distinction between normal and abnormal frames. We make a similitude measure between N_d and T_d to reach an explicit conclusion for each frame, i.e., a frame is said to be *eccentric if* $N_d > T_d$, otherwise *normal*.

3.4.4.3 Estimation of T_d

The hypothetical outlook of the estimation of threshold T_d is that we estimate it from long videos which contain none but normal motions as:

$$T_d = \sqrt{\left[arg \max_{i=1\ldots f} [N_d]_i\right]^2 + \left[arg \min_{i=1\ldots f}\left[\frac{2}{\pi^2}\sum_{n=0}^{\infty}\frac{(-1)^n (N_d)^{2n+1}}{n!(2n+1)}\right]_i\right]^2} \qquad (3.40)$$

where f be the total number of frames. The T_d depends on the controlled environment namely the distance of the camera to the scene, the orientation of the camera, the type and the position of the camera, lighting system, density of the crowd in working, vacation, day, night, weekend, etc. In case of escalator it depends also escalator type and position. Deeming these facts, we have minimum one threshold by a video stream. If we have N video streams, then we choose at least N thresholds. If the video stream changes, then T_d should be regenerated.

3.4.5 Experimental Results and Discussion

To conduct experiments, we have used *Escalator dataset* [132] and the dataset of Minnesota University (UMN dataset) [137]. The publicly available dataset of normal and abnormal crowd videos from University of Minnesota [137] comprises the videos of 11 different scenarios of an escape event in 3 different indoor and outdoor scenes.

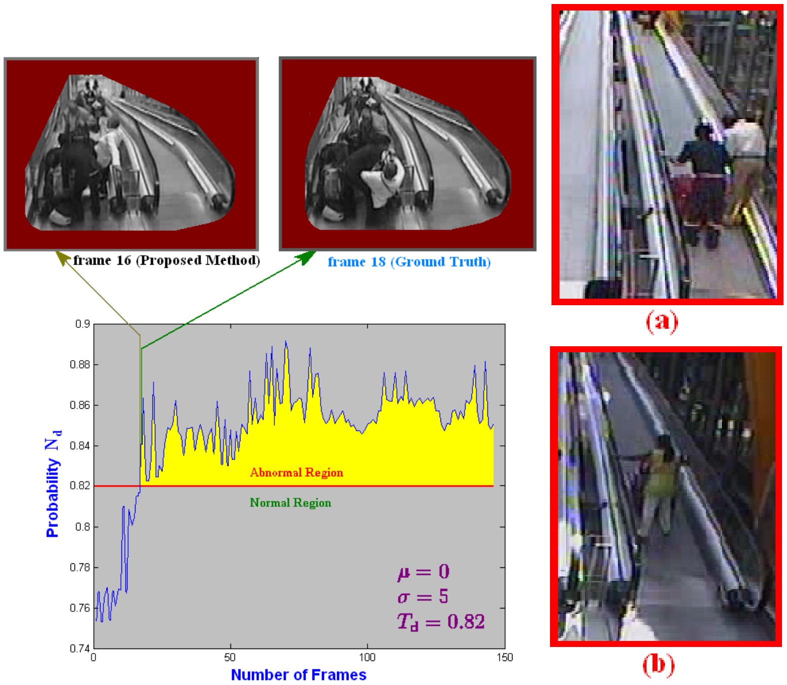


Figure 3.16: Curves are the outputs of the algorithm, which detect eccentric events on escalator exits. But the state of affairs of eccentric events e.g., in images **(a)** & **(b)** cannot be detected due to occlusion.

An example of detection results as shown in the left image of Fig.3.16 which describes a scenario of a collapsing event in an escalator exit point. Some stuffs from a heavily loaded trolley have dropped just the egress point of the moving escalator which has caused one kind of emergency situation on the egress point. The situation has been detected by the proposed algorithm. Nevertheless, the algorithm does not work where video frames bear the situations like Fig.3.16 (a) and (b). This is due to the fact that the video sequences which include abnormal events have occurred with occlusion. Consequently, the quantity of extracted optical flow vectors is not sufficient to draw out abnormal frames. Beyond

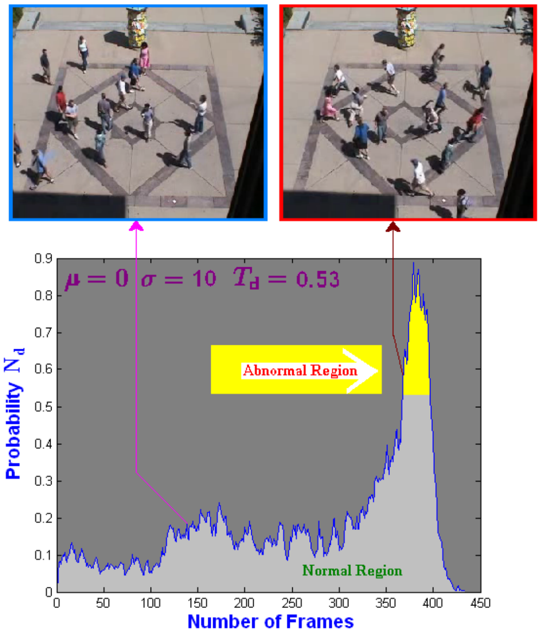


Figure 3.17: Anomaly detection results from a video in UMN [137] by Mahalanobis metric approach.

the escalator unidirectional flow of mob videos, the method has been tested on the videos existing both normal and eccentric events, attributed 320 × 240 pixels, where the movements of people are random directions. Fig.3.17 depicts such a scenario. Initially, the movement of people was random, suddenly they tent to leave their places with very quick motion.

3.5 Bhattacharyya Metric Approach

3.5.1 Overview

An important problem in computer vision is measuring the dissimilarity between distributions of features, e.g., distributions of color and texture features [150]. The focus of this observation is mainly on the Bhattacharyya measure and its derivatives. The χ^2 statistic is used to provide a measure of similarity between two distributions or histograms [110]. The Bhattacharyya measure approximates the χ^2 statistic. The Mahalanobis distance is a particular case of the Bhattacharyya measure. By transforming all variances to be constant the Bhattacharyya measure avoids the singularity problem of the χ^2

statistic when comparing empty histogram bins [2]. The author of [95] compared the Bhattacharyya distance and the Kullback-Leibler divergence, and observed that Bhattacharyya yields better results in some respects while in other respects they are equivalent. A number of measures (e.g., Bhattacharyya, Euclidean, Kullback-Leibler, Fisher) have been studied for image discrimination and it was concluded that the Bhattacharyya distance is the most effective discriminator [18]. Dissimilarity measures, based on empirical estimates of the distribution of feature, have been developed for classification [138], image retrieval [145, 151], unsupervised segmentation [77], edge detection [153], object tracking [38], etc. Introductory benchmark studies have confirmed that distribution–based dissimilarity measures exhibit excellent performance in image retrieval [145], in unsupervised texture segmentation [77], and in conjunction with a k–nearest-neighbor classifier, in color-based or texture-based object recognition [163, 138].

Our proposed approach ([SD10a, SDa]) estimates sudden changes and abnormal motion variations of a set of interest points detected by Harris detector and tracked by optical flow technique and classified by K-means. The overhaul of normalized Bhattacharyya distance measure over time provides the knowledge of the state of abnormal activity. Emphatically, we have noticed that distances between clusters of tracked corners on movers are a reasonable way to characterize abnormal behavior as the distances vary significantly in case of abnormalities. To demonstrate the interest of the approach, we have conducted the experiments on both *Escalator dataset* [132] and *UMN dataset* [137].

3.5.2 Region of interest estimation

Both indoor and outdoor video surveillance would expect *region of interest* (RoI) for making video processing faster. Depending on applications and type of videos, RoI would extend from few parts of a video frame to the whole frame. There are some indoor and outdoor applications (e.g., to keep under surveillance the linear passages, escalator egresses, high-way, etc.) where video processing region can be fixed by using a mask instead of analyzing the whole video frame. We can use either *motion heat map* (e.g., Fig. 3.6) or *spatiotemporal region of interest* (e.g., Fig. 3.11) or *region of interest image map* (e.g., Fig. 3.14) for such applications. Such type map ameliorates the quality of the results and makes the processing time faster as it is not necessary to take into account the whole frame and fastidiously where there are few motion intensities or no motions.

3.5.3 Points of interest estimation

The Harris corner detector [76] is a famous point of interest detector due to its strong invariance to rotation, scale, illumination variation, and image noise [154]. It is based on the local auto-correlation function of a signal, where the local auto-correlation function measures the local changes of the signal with patches shifted by a small amount in different directions. A discrete predecessor of the Harris detector was depicted by Moravec [133], where the discreteness refers to the shifting of the patches. We consider Harris corner as a point of interest. But there is a potential problem for camera positions and lighting conditions which allow to get an extremely large number of corner features that can be not easily be captured and tracked. For example, Fig. 3.18 (a) is the original video frame with moving subjects, if we apply Harris detector algorithm directly then the output contains lot of unwanted corners as shown in Fig. 3.18 (b). To avoid this situation, we prefer to use a background and foreground estimation method before applying Harris corner detector. An estimated foreground can be derived after background estimation. Ideally, residual pixels obtained after applying background subtraction should represent foreground subjects.

Foreground estimation is relatively easy in an indoor environment (see Fig. 3.18 (e) and (f)), because the illumination conditions do not change significantly. An outdoor environment, on the other hand, is much more complicated, as varying weather and sunlight (e.g., shadow of each subject in Fig. 3.18 (a)) affect the correct detection of foreground. Some authors have adopted the adaptive Gaussian approach to model the behaviour of a pixel [161, 75, 193]. However, the background region of a video sequence often contains several moving objects. Therefore, rather than explicitly estimating the values of all pixels as one distribution, we would prefer to estimate the value of a pixel as a mixture of Gaussians [161, 75, 193]. The foreground pixels obtained after applying background subtraction are shown in Fig. 3.18 (c), in which noise, caused by shadows, which are the result of extremely strong lighting condition (e.g., sunlight). On the other hand, Fig. 3.18 (f) is almost light invariant.

3.5.4 Points of interest tracking

Once we define the points of interest, e.g., Fig. 3.18 (d), we track those points over the next frames using optical flow techniques. For this, we use the pyramidal implementation of Kanade-Lucas-Tomasi tracker [123, 160, 26]. Upon matching points of interest between frames, the result is a set of vectors over time:

$$\Psi = \{\Psi_1 \, \dots \, \Psi_N | \Psi_i = (x_i, y_i, \delta_i, \alpha_i)\}$$

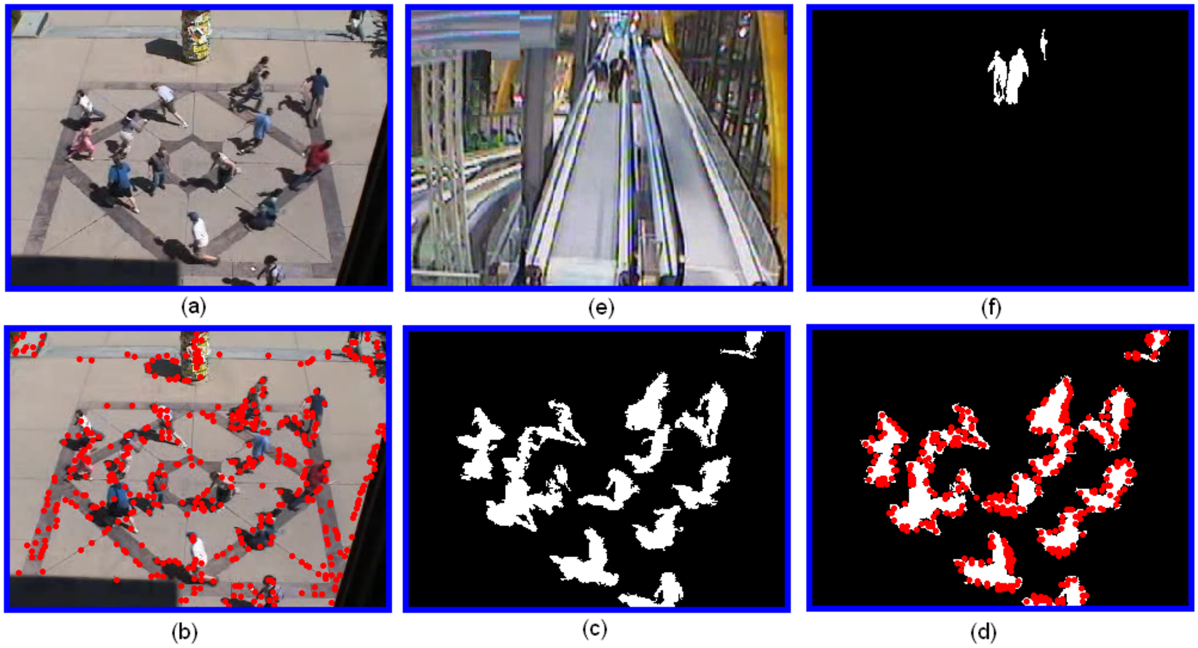


Figure 3.18: The (a) & (e) are the original frames and the results of their foreground estimation have been depicted in (c) & (f) successively; (b) & (d) point to the Harris corner for (a) & (c), respectively.

where

- $x_i \mapsto x$ coordinate of a point of interest i,
- $y_i \mapsto y$ coordinate of the i,
- $\delta_i \mapsto$ displacement of the i from one frame to the next,
- $\alpha_i \mapsto$ direction of motion of the i.

If any feature i in the frame f with its coordinate $U(x_i, y_i)$ and its matched in the frame $f+1$ with coordinate $V(x_i, y_i)$, it is easy to calculate the change of position (displacement) δ_i of the feature i using Euclidean metric as:

$$\delta_i = \sqrt{(U_{x_i} - V_{x_i})^2 + (U_{y_i} - V_{y_i})^2}. \tag{3.41}$$

Simple trigonometric function *atan* comes into notice few potential problems (as described in 3.2.2.4), e. g., infinite slope, false quadrant. On the other hand, trigonometric function *atan*2 gracefully handles infinite slope and places the angle in the correct quadrant [e.g., $atan2(1,1) = \pi/4$, $atan2(-1,-1) = -3\pi/4$, etc.]. Thus, the accurate moving direction α_i of the feature i can be calculated as:

$$\alpha_i = atan2(U_{y_i} - V_{y_i}, U_{x_i} - V_{x_i}). \tag{3.42}$$

Furthermore, we remove static and noisy features. Points of interest having $\delta_i \approxeq 0$ are considered as static features. Noise features are the isolated features which have a big angle and distance difference with their near neighbors due to tracking calculation errors. The resulting points of interest are suitable for clustering.

3.5.5 Classification of points of interest

After static error suppression points of interest, we apply K-means method to get clusters. The geometric clustering method, k-means, is a simple and fast method for partitioning data points into clusters, based on the work done by [120] (so-called Voronoi iteration). It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data. On clustering we represent each class, which contains points of interest of an unknown distribution, as a polygon, as shown in Fig. 3.19. To obtain a quantitative measure of how separable are two classes, a distance measure is required. We calculate the Bhattacharyya distances of all the classes between two consecutive frames over time.

3.5.6 Calculation of Bhattacharyya distance between classes

Original interpretation of the Bhattacharyya measure has few problems, hence we consider the Bhattacharyya bound which commonly uses in pattern recognition. The Bhattacharyya distance has been used as a class separability measure for feature selection and is known provide the upper and lower bounds of the Bayes error.

3.5.6.1 Original Derivation of Bhattacharyya Measure

The original interpretation of the Bhattacharyya measure was geometric [19]. He considered two multinomial populations each consisting of n classes with respective probabilities $P(1), P(2), P(3), \ldots, P(n)$ and $Q(1)$, $Q(2)$, $Q(3)$, $\ldots$, $Q(n)$. Since $P(i)$ and $Q(i)$ present probability distributions, where $i \in n$, it is easy to write $\sum_i^n P(i) = \sum_i^n Q(i) = 1$. He observed that $\sqrt{P(1)}$, $\sqrt{P(2)}$, $\ldots$, $\sqrt{P(n)}$ and $\sqrt{Q(1)}$,

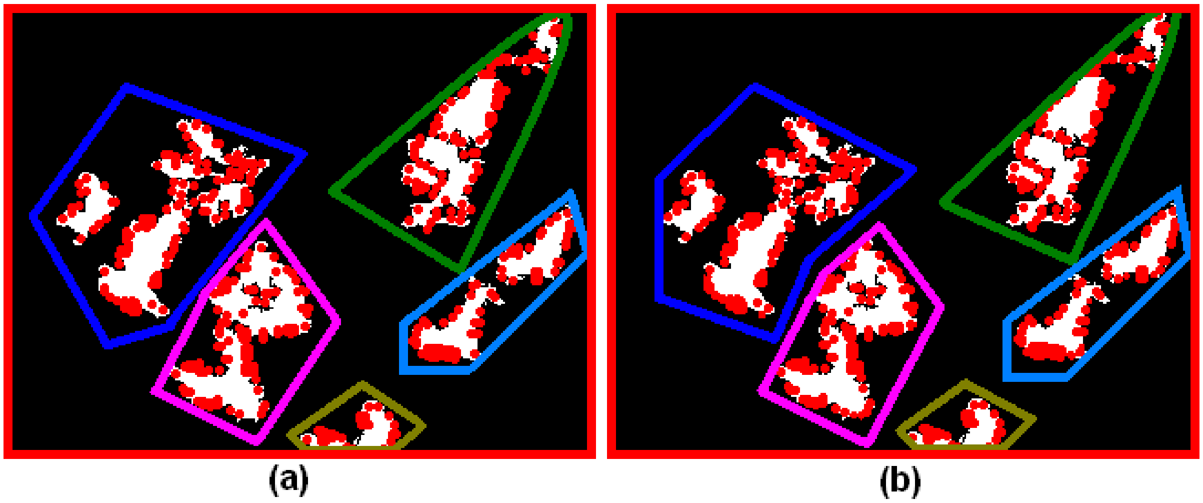


Figure 3.19: Polygons on the two consecutive frames (a) & (b) are the classification of interest points executed by K-means.

$\sqrt{Q(2)}, \ldots, \sqrt{Q(n)}$ could be considered as the direction *cosines* of two vectors in n-dimensional space referred to a system of orthogonal co-ordinate axes. He used the square of the angle between the two position vectors as a measure of divergence between the two populations. Assume that θ be the angle between the vectors, then we have:

$$cos\theta = \sum_{i}^{n} \sqrt{P(i)Q(i)} \tag{3.43}$$

$$\theta = cos^{-1} \sum_{i}^{n} \sqrt{P(i)Q(i)}. \tag{3.44}$$

If the two distributions are identical (e.g., $P(i) = Q(i)$), then we have:

$$\theta = cos^{-1} \sum_{i}^{n} \sqrt{P(i)P(i)} = cos^{-1} \sum_{i}^{n} P(i) = cos^{-1} 1 = cos^{-1} cos 0^\circ = 0^\circ. \tag{3.45}$$

Consequently, we see the intuitive motivation behind the definition as the vectors are collinear. However, a potentially undesirable property of the Bhattacharyya measure or coefficient is that it does not impose a metric structure since it violates at least one of the distance metric axioms [67]. The authors in [38] proposed a derivative of the Bhattacharyya measure in the form of $\sqrt{1 - cos\theta}$ which does indeed represent a metric distance between distributions as this distance obeys all of the metric axioms. Yet, we are interested in the going-over of Bhattacharyya bounds which usually use in pattern recognition.

3.5.6.2 Classification Error

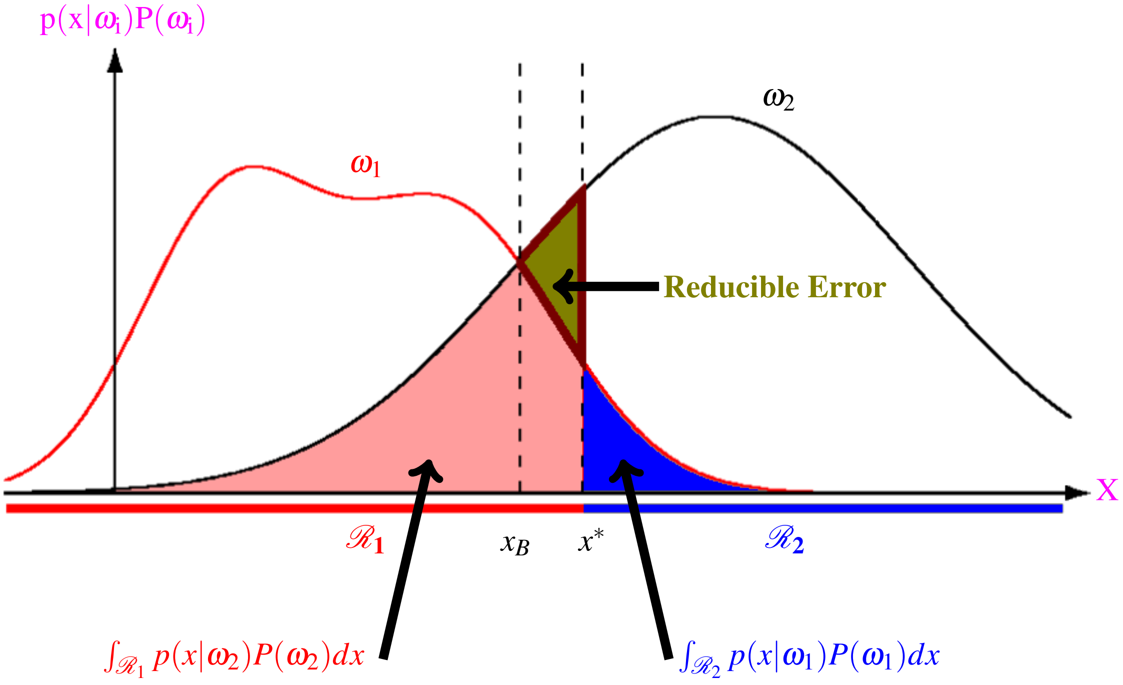


Figure 3.20: Components of the probability of error for equal priors and non-optimal decision point x^*. If the decision boundary is instead at the point of equal posterior probabilities, x_B, then the reducible error is eliminated.

The classification error is the ultimate measure of the performance of a classifier. Bayesian decision theory is a fundamental statistical approach to the problem of pattern classification. A pattern is represented by a set of m attributes or features, viewed as a m-dimensional feature vector $x \in \mathbb{R}^m$. Let us assume a pattern recognition problem, in which the class label ω is a random variable taking values in the set of class labels $\Omega = \{\omega_1, \omega_2, \ldots, \omega_n\}$. The *priori probabilities*, $P(\omega_i)$ where $i \in n$, constitute the probability mass function of the variable ω such that $\sum_{i=1}^{n} P(\omega_i) = 1$. Let consider that the objects from class ω_i are distributed in $x \in \mathbb{R}^m$ according to the *class-conditional probability density function* $p(x|\omega_i)$

such that $p(x|\omega_i) \geq 0$ for $\forall x \in \mathbb{R}^m$ and $\int_{\mathbb{R}^m} p(x|\omega_i)dx = 1$ where $i \in n$. Given the prior probabilities and the class-conditional probability density functions, we can calculate the *posterior probability* $P(\omega_i|x)$ that the true class label of the measured x is ω_i using the *Bayes rule* as:

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)} \tag{3.46}$$

where the evidence is $p(x) = \sum_{i=1}^{n} p(x|\omega_i)P(\omega_i)$. The Eq. 3.46 provides the probability mass function of the class label variable ω for the observed x. The decision for that particular x should be made with respect to the posterior probability. If for some x we have $p(x|\omega_i) = p(x|\omega_{i+1})$ where $\{i, i+1\} \in n$, then the decision hinges entirely on the prior probabilities. On the other hand, if $P(\omega_i) = P(\omega_{i+1})$, then the decision is based entirely on the likelihoods $p(x|\omega_i)$. In general, both of these factors are important in making a decision, and the Bayes decision rule combines them to achive the minimum probability of error. If we have an observation x for which $P(\omega_i|x)$ is greater than $P(\omega_{i+1})$, we would be inclined to choose ω_i. Similarly, if $P(\omega_{i+1}|x)$ is greater than $P(\omega_i|x)$, we would be inclined to choose ω_{i+1}. Thus, we can minimize the probability of error. The average probability of error is given by [50]:

$$P(error) = \int_{-\infty}^{\infty} P(error|x)p(x)dx \tag{3.47}$$

where

$$P(error|x) = min[P(\omega_i|x), P(\omega_{i+1}|x)]. \tag{3.48}$$

If for every x we insure that $P(error|x)$ is as small as possible, then the integral must be as small as possible. For instance, consider the *reducible error* in Fig. 3.20 where a dichotomizer (classifier) has divided the space into two regions $\mathscr{R}_1$ and $\mathscr{R}_2$ in a possibly non-optimal way. There are two ways in which classification error can occur: (i) an observation **x** falls in $\mathscr{R}_2$, (ii) **x** falls in $\mathscr{R}_1$. Since these events are mutually exclusive and exhaustive, the probability of error is [50]:

$$P(error) = P(x \in \mathscr{R}_2, \omega_1) + P(x \in \mathscr{R}_1, \omega_2) = \int_{\mathscr{R}_2} p(x|\omega_1)P(\omega_1)dx + \int_{\mathscr{R}_1} p(x|\omega_2)P(\omega_2)dx. \tag{3.49}$$

Since the decision point x^* were chosen arbitrarily, the probability of error $P(error)$ is not as small as it might be. The triangular area marked *reducible error* can be eliminated if the decision boundary is moved to x_B. This is the Bayes optimal decision boundary and gives the lowest probability of error.

In general, the calculation of the error probability is a difficult task. Even when observation vectors have a normal distribution, we must resort to numerical techniques. However, a closed-form expression for the error probability is the most desirable solution for a number of reasons [67]. Not only is the computational effort greatly reduced, since we need only to evaluate a formula, but more importantly, the use of the closed-form solution provides insight into the mechanisms causing the errors. When we cannot obtain a closed-form expression for the error probability, we may take some other approach. We may seek either an approximate expression for the error probability, or an upper bound on the error probability.

3.5.6.3 The Chernoff and Bhattacharyya Bounds & Distances

In order to derive a bound for the error of Eq. 3.47, the following inequality [67, 50] is extremely helpful:

$$min[a,b] \leq a^{\xi} b^{1-\xi} \; for \; a,b \geq 0 \; and \; 0 \leq \xi \leq 1. \tag{3.50}$$

Using Eq. 3.48 & 3.46, we have:

$$P(error|x) = min\left[\frac{p(x|\omega_i)P(\omega_i)}{p(x)}, \frac{p(x|\omega_{i+1})P(\omega_{i+1})}{p(x)}\right]. \tag{3.51}$$

On applying the inequality of the Eq. 3.50, we obtain:

$$P(error|x) \leq \left(\frac{p(x|\omega_i)P(\omega_i)}{p(x)}\right)^{\xi} \left(\frac{p(x|\omega_{i+1})P(\omega_{i+1})}{p(x)}\right)^{1-\xi}. \tag{3.52}$$

Using Eq. 3.52 and Eq. 3.47, the new equation yields:

$$P(error) \leq P^{\xi}(\omega_i)P^{1-\xi}(\omega_{i+1}) \int p^{\xi}(x|\omega_i)p^{1-\xi}(x|\omega_{i+1})dx \tag{3.53}$$

which is called the *Chernoff bound* where $0 \leq \xi \leq 1$. In general, the *Chernoff upper bound* of error is expressed as:

$$P(error) = P^{\xi}(\omega_i)P^{1-\xi}(\omega_{i+1}) \int p^{\xi}(x|\omega_i)p^{1-\xi}(x|\omega_{i+1})dx. \tag{3.54}$$

The optimal ξ can be found by minimizing $P(error)$. The integral of Eq. 3.53 is over all feature space, we do not need to impose integration limits corresponding to decision boundaries. If two density functions are normal as $\mathcal{N}(\mu_i, \Sigma_i)$ and $\mathcal{N}(\mu_{i+1}, \Sigma_{i+1})$, where μ_i & μ_{i+1} and Σ_i & Σ_{i+1} are the *mean* vectors and *covariance* matrices of classes i & $i+1$ respectively, then the integral in Eq. 3.53 can be evaluated analytically (a close-form expression), yielding [67, 50]:

$$\int p^{\xi}(x|\omega_i)p^{1-\xi}(x|\omega_{i+1})dx = e^{-k(\xi)} \tag{3.55}$$

where

$$k(\xi) = \frac{\xi(1-\xi)}{2}[\mu_{i+1}-\mu_i]^T[\xi\Sigma_i+(1-\xi)\Sigma_{i+1}]^{-1}(\mu_{i+1}-\mu_i)+\frac{1}{2}log_e\frac{|\xi\Sigma_i+(1-\xi)\Sigma_{i+1}|}{|\Sigma_i|^{\xi}|\Sigma_{i+1}|^{1-\xi}}. \tag{3.56}$$

This expression of $k(\xi)$ is called the *Chernoff disrance*. The optimum ξ is the one which gives the maximum value for $k(\xi)$. The Chernoff bound, on $P(error)$ is found by analytically or numerically finding the value of ξ that minimizes $e^{-k(\xi)}$, and substituting the results in Eq. 3.53. The key benefit here is that this optimization is in the one-dimensional ξ space, despite the fact that the distributions themselves might be in a space arbitrarily high dimension [50]. The Chernoff error bound is loose for extreme values (i.e., $\xi \to 1$ and $\xi \to 0$) and tighter for intermediate ones. While the precise value of the optimal ξ depends on the parameters of the distributions and the prior probabilities, a computationally simpler, but slightly less tight bound can be derived by merely asing the results for $\xi = \frac{1}{2}$. Substituting $\xi = \frac{1}{2}$ in Eq. 3.55 & 3.53 and rename $k(\frac{1}{2})$ as β (i.e., $\beta = k(\frac{1}{2})$), we have the form:

$$P(error) \leq \sqrt{P(\omega_i)P(\omega_{i+1})}\int p(x|\omega_i)p(x|\omega_{i+1})dx = \sqrt{P(\omega_i)P(\omega_{i+1})}\,e^{-\beta} \tag{3.57}$$

which is the so-called *Bhattacharyya bound* on the error. Bhattacharyya error bound is always looser than Chernoff error bound. In general, the *Bhattacharyya upper bound* on the error has the form as:

$$P(error) = \sqrt{P(\omega_i)P(\omega_{i+1})}\int p(x|\omega_i)p(x|\omega_{i+1})dx = \sqrt{P(\omega_i)P(\omega_{i+1})}\,e^{-\beta}. \tag{3.58}$$

Substituting $\xi = \frac{1}{2}$ in Eq. 3.56, we have for the normal distributions case:

$$\beta = \mathbf{\frac{1}{8}}[\mu_i - \mu_{i+1}]^{\mathbf{T}}\left[\frac{\Sigma_i+\Sigma_{i+1}}{\mathbf{2}}\right]^{\mathbf{-1}}[\mu_i-\mu_{i+1}]+\mathbf{\frac{1}{2}\log}_e\frac{|\frac{\Sigma_i+\Sigma_{i+1}}{\mathbf{2}}|}{\sqrt{|\Sigma_i||\Sigma_{i+1}|}} \tag{3.59}$$

where the term β is called the ***Bhattacharyya distance***, which will be used as an important measure of the separability of two distributions.

3.5.6.4 Effective Distance G_β Calculation

The first term of Eq. 3.59 gives the class separability due to the difference between class means, while the second term gives the class separability due to the difference between class covariance matrices. To compute the (*p*,*q*)-th element of the Σ_i or Σ_j, where $j = i+1$, we use the following equation:

$$\Sigma_r(p,q) = \frac{1}{s-1}\left[\sum_{r=1}^{s}\Psi_r(p)\Psi_r(q) - \frac{1}{s}\sum_{r=1}^{s}\Psi_r(p)\sum_{r=1}^{s}\Psi_r(q)\right] \tag{3.60}$$

where m and n are the number of points of interest in the classes of i and j respectively, $r \in \{i,j\}$, $s \in \{m,n\}$, $\{p,q\} \in \{x_r,\, y_r,\, \delta_r,\, \alpha_r\}$. Now, we calculate the difference of class means by means of:

$$\mu_i - \mu_j = \begin{bmatrix} \frac{1}{m}\sum_{i=1}^{m} x_i - \frac{1}{n}\sum_{j=1}^{n} x_j \\ \frac{1}{m}\sum_{i=1}^{m} y_i - \frac{1}{n}\sum_{j=1}^{n} y_j \\ \frac{1}{m}\sum_{i=1}^{m} \delta_i - \frac{1}{n}\sum_{j=1}^{n} \delta_j \\ \frac{1}{m}\sum_{i=1}^{m} \alpha_i - \frac{1}{n}\sum_{j=1}^{n} \alpha_j \end{bmatrix}. \tag{3.61}$$

The Mahalanobis distance is a particular case of the Bhattacharyya, when the variances of the two classes are equal, this would eliminate the second term (Eq. 3.59) of the distance. This term depends solely of the variances of the distribution. If the variances are equal this term will be zero, and it will grow as the variances are different. The first term, on the other hand will be zero if the means are equal and is inversely proportional to the variances. Besides the mathematical formulation, it may be interesting to consider some of its properties. Fig. 3.21 shows a one-dimensional example: as a comparison of (a) and (c) we can come across that, while the Euclidean distance is the same in this two cases, β is larger in (c) than that of (a). This is because the distance between the means is scaled by the variances and expresses the degree of overlapping of the two distributions. The similar view can be viewed by considering (a) and (b): in this case β is approximately the same, while the distance between the means is different. Finally, (d) shows how the variances of the two variables may be differing in general. Upon calculating all Bhattacharyya distances among classes, we calculate the geometric means of the Bhattacharyya distances among classes and come together those means to calculate the final geometric mean (or log-average) to represent a single effective distance G_β between two consecutive frames using Algorithm 1. The advantage of using the geometric mean is that it reduces the effect of very high and low (perhaps even exponentially changing data) values in a number set. Theoretically, clustering may be very sensitive and distances between clusters may change significantly from frame to

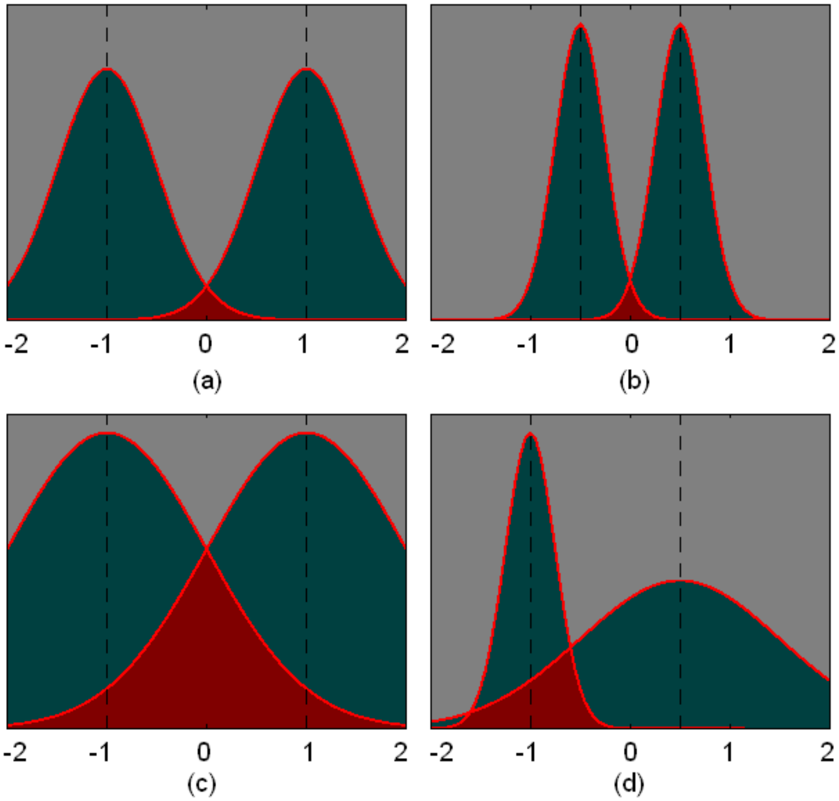


Figure 3.21: Bhattacharyya distance surrounds completely for one-dimensional example of twosomes of Gaussian distributions: (a) and (c) present twosomes with the nondescript mean Euclidean distance nevertheless different Bhattacharyya distances, (a) and (b) have in like manner Bhattacharyya distance but different mean Euclidean distances; (d) depicts differing distributions and distances.

frame. The advantage of calculating single effective distance G_β between consecutive frames is that it minimizes such effect in a great amount.

In the crowd scene in case of abnormal and/or emergencies situations physically there exists sufficient agitation and hence the positions, displacements, and directions of points of interest in the clustering are noticeably different between frames. In such situation, clustering configurations like 3.21 (a) or (b) or (d) tend to (c) between two consecutive frames. Explicitly, the value of β and hence is the G_β will be higher. Similarly, as compare to abnormal case, the clustering configurations of normal cases are almost similar between two consecutive frames. Thus the value of β and so is the G_β will be smaller, i.e., clustering configurations like 3.21 (a) or (b) or (d) remain almost the same. Intuitively speaking, the distances between clusters of tracked corners on movers are a reasonable way to characterize abnormal behavior as the distances vary significantly in case of abnormalities.

Algorithme 1 : Effective Distance $\mathbf{G}_\beta$ Calculation

```
▷ F: total number of classes in any frame f
▷ S: total number of classes in the frame $f+1$
▷ $c_m$: class counter in frame f
▷ $c_n$: class counter in frame $f+1$
$c_m \leftarrow 1$ ; $c_n \leftarrow 1$
while $c_m \leq F$ do
    while $c_n \leq S$ do
        using Eq. 3.59 calculate $\beta$ between classes of $c_m$ & $c_n$, and store as $\beta_{c_n}$ $c_n \leftarrow c_n + 1$
    end
    calculate the geometric mean of $\beta_{c_n}$ by means of:
        $\Omega_{c_m} = \left[\prod_{i=1}^{c_n} \beta_i\right]^{\frac{1}{c_n}} = exp\left[\frac{1}{c_n}\sum_{i=1}^{c_n} log_e \beta_i\right] \quad (3.62)$
    and store the calculated $\Omega_{c_m}$
    $F \leftarrow F - 1$
    $c_n = 1$ ; $c_m \leftarrow c_m + 1$
end
calculate geometric mean of $\Omega_{c_m}$ by dint of:
    $G_\beta = \left[\prod_{j=1}^{c_m} \Omega_j\right]^{\frac{1}{c_m}} = exp\left[\frac{1}{c_m}\sum_{j=1}^{c_m} log_e \Omega_j\right] \quad (3.63)$
```

3.5.7 Normalization

Now, we wish to transfer each G_β into a normalized distance value between 0 and 1. For normalization purpose, we could use the simple formula like $1/(1 + log_e G_\beta)$, but the normalized values fall in a congested range (scaling problem) which will arise problem specially in threshold selection. To solve the scaling problem, we would like to use a versatile distribution which has significant effect on its shape and scale parameters. In this respect we use cumulative distribution function (cdf) of Weibull distribution [174] which has strict lower and upper bounds between 0 and 1. Due to accurate model quality and performance characteristics of Weibull distribution and its flexibility that makes it ideal for analysis on a dataset with unknown distribution. It is worth mentioning that Weibull distribution can mimic the behavior of other statistical distributions such as the normal and the exponential. If Φ_β denotes the normalized distance value of G_β, then Φ_β can be formulated as:

$$\Phi_\beta = 1 - e^{-(G_\beta/\lambda)^\nu} \tag{3.64}$$

where $\nu > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution. Using Eq. 3.64, and knowing the values of ν, λ, and G_β we can explicitly estimate the value of Φ_β between 0 and 1.

3.5.8 Threshold estimation

A predefined threshold Γ_β value can differentiate each frame with respect to its assigned distance value whether its motion is normal or abnormal. There are several methods which may apply to estimate Γ_p. One of the simplest approaches of computing Γ_p is that we consider the maximum number of distances in large videos that contain exclusively normal motions:

$$\Gamma_\beta = arg \max_{k=1...t} \left[\ \Phi_\beta\ \right]_k + arg \min_{k=1...t} [G_{error}]_k \tag{3.65}$$

$$G_{error} = \frac{1}{\sqrt{\pi}} \sum_{m=0}^{\infty} \left(\frac{\Phi_\beta}{2m+1} \prod_{k=1}^{m} \frac{-{\Phi_\beta}^2}{k} \right) \tag{3.66}$$

where t is the number of frames of the video database and the Gauss error function G_{error} is exactly 0.5 at ∞. On affixing the order of the Eq. 3.66 series, G_{error} depends on the Φ_β.

Any frame having value of Φ_β which is greater than the Γ_β will be considered as abnormal motion frame. The Γ_β depends on the controlled environment, namely the distance of the camera to the scene, the orientation of the camera, the type and the position of the camera, density of the crowd, varying illumination, light reflection, over head light, shadowing, day-night, week days, winter-spring, indoor-outdoor, occasion, vacation, etc. The more is the distance of the camera to the scene, the less is the quantity of optical flows and blobs. Taking into account all of these facts, we assume that we have at least one threshold by a video stream. If we have *M* video streams, which are the case in sites such as play grounds, sporting events, town centers, parking places, political events, airports, subways, stations, banks, concerts, cinema halls, schools, shopping malls, hospitals, hotels, etc., then we select at least *M* thresholds. If the environment changes, then the threshold should be regenerated.

3.5.9 Experimental Results and Discussion

To conduct experiments, we have used *Escalator dataset* [132] and the dataset of Minnesota University so-called *UMN dataset* [137].

Fig. 3.22 describes an example of an abnormal situation on the escalator exit point in a video stream. Two persons were standing on the moving escalator, suddenly a trolley rushed out toward them. One person escaped by running while other did not. The non-escapee was rundown by the run-away trolley, and subsequently fell down at the exit point of the moving escalator. The situation was detected by the proposed algorithm.

The qualitative results of the abnormal behavior detection for a sample videos of UMN dataset have been presented in Fig. 3.23. The video of the given sample abnormal motion includes a sudden situation when a group of people start running and the assigned distance Φ_β will be higher than any other before assigned distances. The Gaussian like curve represents the abnormal motion when the group of people is trying to leave the place with very quick motion. For explicitness, two arbitrary video frames and their corresponding positions on the output curves have been indicated by arrows. Fig. 3.23 demonstrates that the proposed framework accomplishes something to a greater degree to distinguish aberrant sequences. So far, we can conclude that distances between clusters of tracked corners on movers are an acceptable way to characterize abnormal behavior as the distances vary significantly in case of abnormalities.

However, the approach does not work on occlusion cases due to the shortcomings of optical flow technique. As inconvenient, the approach used the distance to measure differences between clusters or classes and compute a single value to determine the difference between activity in two frames. It would presumably have difficulties when there are multiple co-occurring activities and one changes.

As a future work, one would in principle try to apply the same idea to individual clusters. It is also noticeable that the lighting condition, which causes specially shadows of moving bodies, has a severe effect on the background subtraction which has been mostly overlooked. It would be worth interesting to count this effect in many computer vision applications. Accordingly, in future work the effect could be taken into account and minimized.

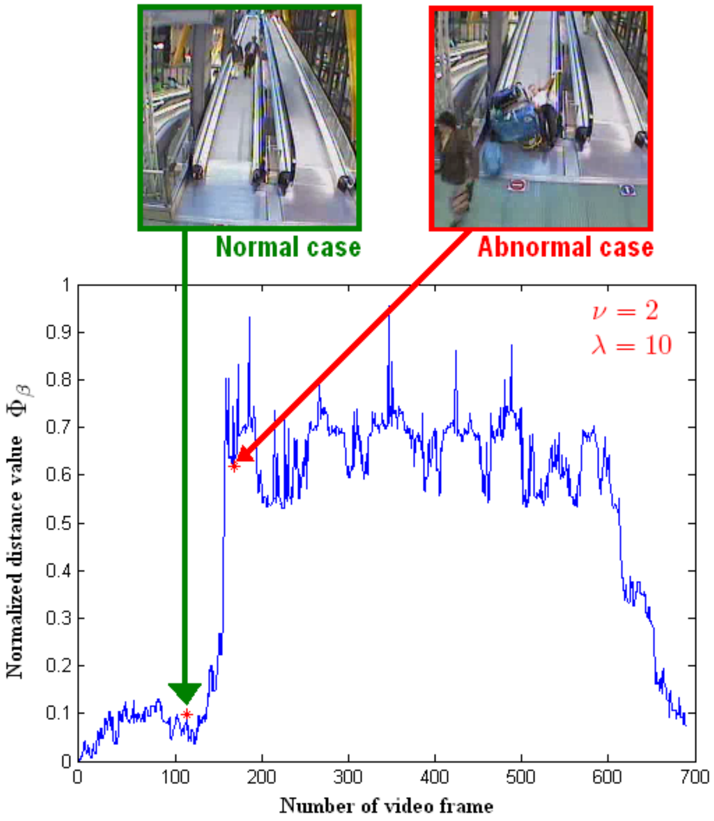


Figure 3.22: Person falling event on the escalator exit has been detected by Bhattacharyya metric approach.

3.6 Enumerated Entropy Approach

3.6.1 Overview

Likewise, the framework exposes in this section makes known abnormal motion frames from real videos. The hypothesis of our approach ([SID08b, SID10, ISD08]) is to consider the detection of abnormal events in a crowded context from video surveillance data. The framework does not consider individual person tracking and consider the study of the general motion aspect and more particularly assesses sudden shift and strange locomotion discrepancy of a set of interest points discovered by Harris point of interest detector, instead of tracking persons one by one. The detection and tracking of indi-

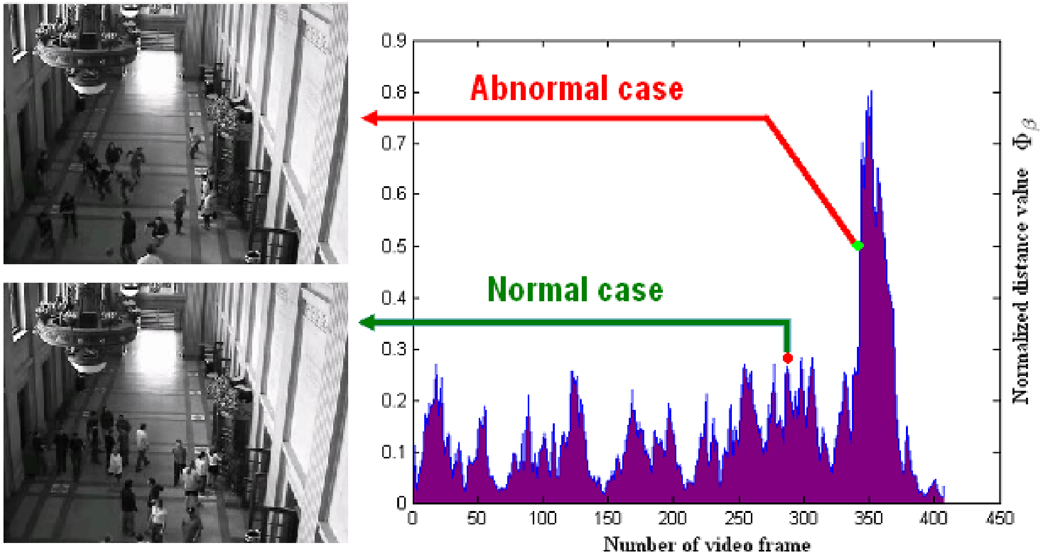


Figure 3.23: Qualitative results of the proposed Bhattacharyya metric approach for abnormality detection from a video in UMN dataset.

vidual persons are difficult in the case of crowded situations. The framework is composed of several steps:

- The motion heat map is extracted. The heat map represents the motion intensities e.g., hot area corresponds to high motion intensities, cold areas represent to low motion intensities, etc.

- Harris points of interest are extracted in the hot regions of the scene. In the simplest case, it is applied in well limited areas. We consider a binary heatmap, white (movement), and black (no movement). Points of interest are applied into white regions and blobs are extracted.

- Optical flows are computed on those points of interest, marked off the boundaries by the hot areas of the scene.

- Mid-level features, the statistical scrutiny of the optical flow information, e.g., density, coefficient of direction variation, coefficient of distance variation, and direction histogram, are computed.

- On the basis of mid-level features computed in the previous step, we define high-level features (e.g., entropy) which classify events in abnormal/normal and return different types of abnormality.

The 1-4 steps are generic and do not depend of a specific application domain. They concern the extraction of low and mid level of features. The fifth step is dependent of the application domain and requires a specific learning process. The flow diagram of our planned work has been depicted in Fig.3.24, which is articulated on a framework operating in three-level of features as noted below.

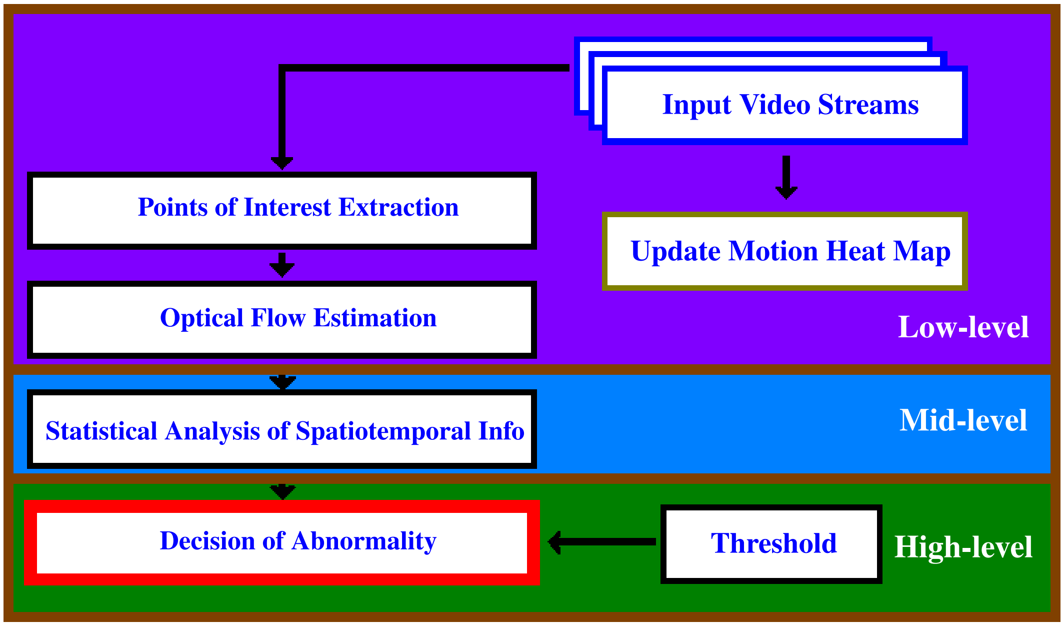


Figure 3.24: Simple block diagram of the proposed framework.

- Low-level: It bears reference to measurements those extracted directly from the signal (visual data), e.g., point of interests, region of interests (blobs), edges, ridges, optical flow, etc. We use mixture of Gaussian to detect foregrounds in case of low crowded (low density) areas, and optical flows on points of interest for high crowded (high density) areas.

- Mid-level: It concerns of features those are generated after a learning process directly from the low-level features, and helps more to enhance the upper level features (semantics), e.g., crowd density (ratio of the blobs in the scene), trajectory, velocity, direction, acceleration, energy, and so forth. The mid-level features are computed on lowlevel features (e.g., interest regions, interest points) and are stored in basic structures.

- High-level: It pertains to the features with more semantics than mid-level, and they are enough to take decision. Here we are in the possession of the normal/abnormal events.

3.6.2 Low-level Features

Motion heat map, points of interest extraction, and estimation of optical flow have been already explained in 3.2.2.1, 3.2.2.2, and 3.2.2.3, respectively. We will discuss a necessary part of optical flow.

On defining points of interest (features), we can obtain a set of vectors as:

$$V = \{V_1 \ \ldots \ V_N | V_i = (X_i, Y_i, D_i, \theta_i)\} \tag{3.67}$$

where

- $X_i \mapsto x$ coordinate of some feature i,
- $Y_i \mapsto y$ coordinate of the i,
- $D_i \mapsto$ distance between the feature i in the frame f and its matched feature in frame $f+1$,
- $\theta_i \mapsto$ direction of motion of the i.

If any feature i in the frame f with its coordinate $P(x_i, y_i)$ and its matched in the frame $f+1$ with coordinate $Q(x_i, y_i)$, it is easy to calculate the change of position (displacement) D_i of the feature i using Euclidean metric as:

$$D_i = \sqrt{(Q_{X_i} - P_{X_i})^2 + (Q_{Y_i} - P_{Y_i})^2}. \tag{3.68}$$

Also, the accurate moving direction θ_i of the feature i can be calculated as:

$$\theta_i = atan2(Q_{Y_i} - P_{Y_i}, Q_{X_i} - P_{X_i}). \tag{3.69}$$

3.6.3 Mid-level Features

We define some mid-level features those will be necessary to induce a specific abnormal event.

3.6.3.1 Motion area ratio M_R

In each video frame, the M_R estimates the ratio between the number of blocks containing motion and the total number of defined blocks. In crowded scenes the area covered by the moving blobs is important as compared to uncrowded scenes. We use this measure as a density estimator. To estimate M_R, we divide each video frame into $N \times M$ blocks, where N & M are number of columns & rows respectively. For any block (i, j), we define the moving block by means of:

$$movingblock(i,j) = \begin{cases} 1; & if\ movement\ \ exists \\ 0; & otherwise \end{cases}$$

If there are several movements exist in one block, then that block will be enumerated as one moving block. We count out the total number of moving blocks to define M_R as:

$$M_R = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} movingblock(i,j)}{N \times M}. \tag{3.70}$$

3.6.3.2 Direction variance-mean ratio θ_R

To estimate direction variance (θ_V), it is important to estimate the mean direction $\overline{\theta}$ of the optical flow vectors in each video frame. The $\overline{\theta}$ is determined by:

$$\overline{\theta} = \frac{1}{n}\sum_{i=1}^{n}\theta_i \tag{3.71}$$

where n is cardinality of the optical flow vectors in the frame and $360^\circ \geq \theta_i > 0^\circ$. Having firsthand knowledge of $\overline{\theta}$, we calculate the θ_V of those vectors as:

$$\theta_V = \frac{1}{n-1}\sum_{i=1}^{n}(\theta_i - \overline{\theta})^2 = \frac{1}{n-1}\sum_{i=1}^{n}\theta_i^2 - \frac{n}{n-1}\overline{\theta}^2. \tag{3.72}$$

The direction variance-to-mean ratio (coefficient of direction variation) is defined as the ratio of the variance to the mean:

$$\theta_R = \frac{\theta_V}{\overline{\theta}}. \tag{3.73}$$

3.6.3.3 Direction histogram θ_H

The θ_H gives directions to the direction tendencies and the number of peaks. In the histogram each column puts an address on the number of vectors in a given angle. The θ_H, which is affiliated to the frame, can be clearly characterized by the way of:

$$\theta_H = \{\theta_H(\theta_i), i = 1\ldots s\} \tag{3.74}$$

$$\theta_H(\theta_i) = \frac{\sum_{i=1}^{n} angle(i)}{s} \tag{3.75}$$

$$angle(i) = \begin{cases} 1, & if \quad angle(i) = \theta_i \\ 0, & otherwise \end{cases}$$

where $\theta_H(\theta_i)$ is the normalized frequency of optical flow vectors those have the same angle θ_i. The θ_H is a vector of size s where s is the total number of angles considering the angle range between $-\pi$ and $+\pi$.

3.6.3.4 Distance variance-mean ratio D_V

Observation shows that distance variance (D_V) increases in abnormal situations. With one or many people walking even in different directions, they tend to have the same speed, which means a small value of the motion distance variance. But in case of abnormal observable activities (e.g., collapsing situations, a sudden overwhelming fear, escape circumstances, etc.) those often give rise to a big value for the D_V. The mean of distance variance $\overline{D}$ is clearly delimited by:

$$\overline{D} = \frac{1}{n}\sum_{i=1}^{n} D_i \tag{3.76}$$

where n is the number of optical flow vectors in the frame. Having $\overline{D}$ it is easy to ascertain D_V by:

$$D_V = \frac{1}{n-1}\sum_{i=1}^{n}(D_i - \overline{D})^2 = \frac{1}{n-1}\sum_{i=1}^{n} D_i^2 - \frac{n}{n-1}\overline{D}^2. \tag{3.77}$$

The distance variance-to-mean ratio (coefficient of distance variation) is defined as the ratio of the variance to the mean:

$$D_R = \frac{D_V}{\overline{D}}, \quad where \ \overline{D} > 0. \tag{3.78}$$

3.6.4 High-level features

High-level features concern the decision of event which is either normal or abnormal. These features are denoted as entropies. We developed a function entitled *Entropy* that extracts the features at the frame f. The f is not explicated in the formula to keep the presentation simple.

3.6.4.1 Entropy estimation

The enumerated function *Entropy*, that depends on motion area ratio, coefficient of direction variation, coefficient of distance variation, and direction histogram characteristics at any frame f, is formulated as:

$$\{Entropy\}_f = P(E_f) log \frac{1}{P(E_f)} = -P(E_f) log P(E_f) \tag{3.79}$$

where $0 \leq P(E_f) \leq 1$. In case of $P(E_f) = 0$, the $P(E_f)logP(E_f)$ will be considered as 0. The E_f is defined as:

$$E_f = M_R \times \theta_R \times \theta_H \times D_R. \tag{3.80}$$

What we propose here is a way to detect collapsing event, which is an eccentric event in a crowded environment. The framework may be extended by any high-level features which are computed of mid-level features.

To calculate $P(E_f)$ we use cumulative distribution function (cdf) which has strict lower and upper bounds between 0 and 1. Deeming $\Omega_{\mu,\sigma}(E_f)$ denotes the cdf of E_f. Then $\Omega_{\mu,\sigma}(E_f)$ can be expressed in terms of a special function called the *error function* (erf) or *Gauss error function*, as:

$$\Omega_{\mu,\sigma}(E_f) = \frac{1}{2}[1 + erf\{\frac{E_f - \mu}{\sigma\sqrt{2}}\}] \tag{3.81}$$

where $\sigma > 0$ is the standard deviation and the real parameter μ is the expected value. The erf can be defined as a *Maclaurin* series:

$$erf(E_f) = \frac{2}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n \{E_f\}^{2n+1}}{n!(2n+1)} \tag{3.82}$$

$$= \frac{2}{\sqrt{\pi}}\{E_f - \frac{{E_f}^3}{3} + \frac{{E_f}^5}{10} - \frac{{E_f}^7}{42} + \frac{{E_f}^9}{216} - \ldots\}. \tag{3.83}$$

Since E_f is skewed to the right (positive-definite) and variances also large, we can use Log-normal distribution. Skewed distributions are particularly common when mean values are low, variances large, and values cannot be negative. Log-normal distributions are usually characterized in terms of the log-transformed variable, using as parameters the expected value, or mean, of its distribution, and the standard deviation. This characterization can be advantageous as log-normal distributions are symmetrical again at the log level [115]. The structure of log-normal distribution of the Eq. 3.81 yields:

$$P(E_f) = \frac{1}{2}[1+erf\{\frac{log(E_f)-\mu}{\sigma\sqrt{2}}\}]. \tag{3.84}$$

By means of Eq. 4.48 & 3.83, and having extensive information of the values of μ and σ (say $\mu = 0$, $\sigma = 10$) we can explicitly estimate the value of $P(E_f)$ between 0 and 1.

3.6.4.2 Threshold estimation

To decide the normality or abnormality of the event on the basis of the function analysis, we examine and note the similarities or differences of each calculated value of $Entropy$ with a beforehand defined entropy threshold T_E, i.e., a deviant frame can be detected *if & only if* $Entropy < T_E$, otherwise standard frame. Hypothetical outlook of reckoning T_E is that we pay attention to the minimum number of entropies in large videos which keep under control snobbishly standard events:

$$T_E = \min_{k=1...n}\{Entropy\}_k \tag{3.85}$$

where n is the number of frames of the video database. If we have N video streams, which are the case in sites such as airport, shopping mall, bank, play ground, subway, concert, cinema hall, school, hospital, parking place, town center, political event, etc., then we put forward at least N thresholds. If the video stream leaves for another, then the threshold should be regenerated.

3.6.5 Experimental Results and Discussion

To conduct experiments, we have used *Escalator dataset* [132] as well as the dataset of Minnesota University namely *UMN dataset* [137].

Fig.3.25 demonstrates a breakdown circumstances on the escalator exit points in the presence of large amount of people respectively. Fig.3.25 manifests aberrant situation where suddenly the wheels of a trolley clenched on the escalator egress and in the long run as a consequence causing perilous and inconsistent circumstances on the escalator egress. The neurotic situations were successfully detected by the proposed algorithm.

The approach has been tested on the videos where the movements of people are random directions (e.g., *UMN dataset* [137]), for instance Fig.3.26. This video consists of 657 frames with attribute 320×240 where both normal and abnormal motion exist. Abnormal motion includes a sudden situation

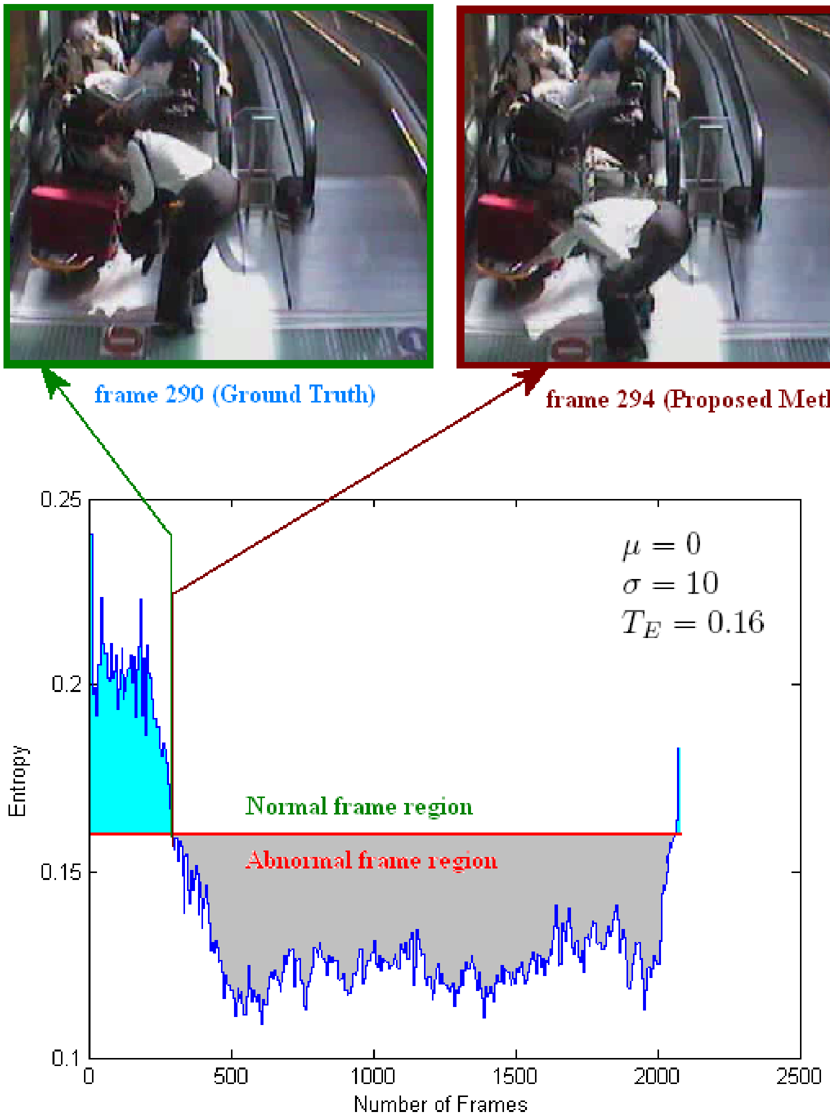


Figure 3.25: Suddenly the wheels of a trolley held firmly and tightly on the escalator exit and eventually as a result causing perilous and inconsistent circumstances on the egress. The blue colored curve indicates the output of the algorithm.

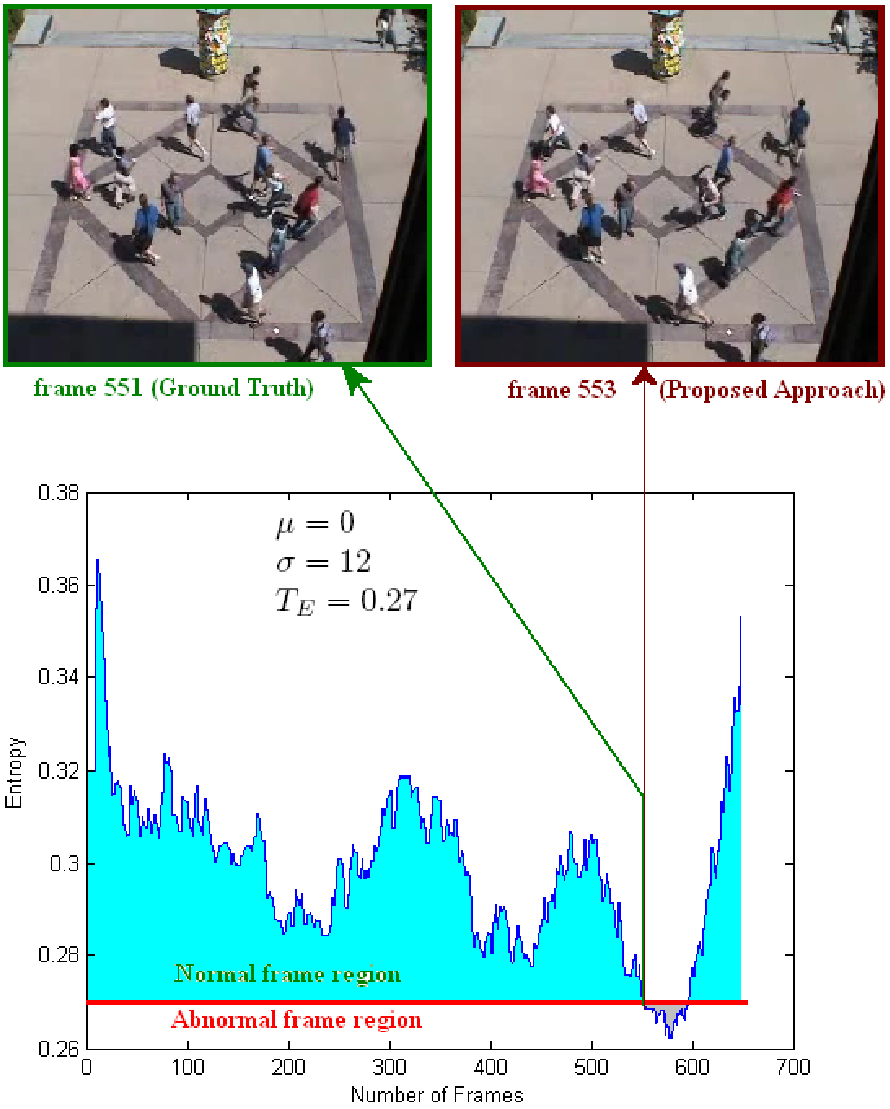


Figure 3.26: Aberrant event (canyon like part) has been detected by the algorithm when the group of people has started rushing along random directions. Blue colored curve points to algorithm's output.

when a group of people start running. From frame 1 to 550 the motion of people is normal. People tend to run from frame number 551. More precisely, the entropy of frame 551 will be lower than that of any other before encountered. Consequently, this frame can be considered as the ground truth frame. In Fig.3.26 the blue colored curve is the output of the proposed approach. The canyon like region represents the abnormal motion activities when the group of people has started to leave their places with very quick motion. For clarity, the ground truth frame 551 and the output abnormal video frame 553, and their corresponding positions on the output curve have been indicated by arrows.

In conclusion, we can explicitly come to an end that the approach is befitting for detection diverse kind of abnormal events. Notwithstanding, it would presume a few limitations as directed in the following.

- First, we have defined the function *Entropy* ourselves, and hence it does not reflect the exact definition of *Entropy* which normally used in *information theory* (so called Shannon Entropy). Explicitly, our enumerated *Entropy* is constructed from a single probability rather than from a set of probabilities summing to 1. It noteworthy that in Shannon Entropy the estimation of the maximum entropy is interesting, but the interest of our defined entropy is the estimation of the minimum entropy.

- Second, we considered that in video surveillance scenes, camera positions, and lighting conditions allow getting a large number of Harris corners that can be easily captured and tracked. Since the mid-level features are extracted based on the result of Harris corner detection, these features might be sensitive to textures. For example, if a person wears a grid-dresslike cloth, there will be too many corners detected from the region of him/her so that most motion directions (e.g., 50% or more) in this frame are the same as the movement direction of the person. Features like direction histogram would be distorted in this situation. Further investigation would take into account this presupposition. A potential solution of this problem could be figured out as: each object can be characterized by a set of corners obtained with a color Harris detector. Each corner can be distinguished by its local appearance such as a vector of local characteristics. The use of a set of corners allows tracking the object through partial occlusion as long as one or more corners remain visible. To increase robustness, it could be important to exploit potential geometric relationships between the corners. Finally, it would be worth investigating to include more high level features and using suitable classifiers to eliminate the threshold computing step and evaluating the technique with a wide range of data-set.

3.7 Shannon Entropy Approach

3.7.1 Overview

Estimation of entropy is an important problem that arises in statistical pattern recognition, adaptive vector quantization, image registration and indexing, and other areas. Non-parametric estimation of Shannon entropy has been of interest to many in non-parametric statistics, pattern recognition, model identification, image registration and other areas [3, 168, 49, 90, 74, 13, 171].

In this section, we propose another simple but effective method ([SDb]) to detect anomalies in videos using Entropies, which are measured on the statistical treatments of the spatiotemporal information of a set of points of interest within a region of interest by measuring their degrees of disorder/chaos over time. It does not depend on segmentation or individual subject tracking, instead it takes the advantages of the use of entropy such as robustness against variable number of subjects in the scenes. Normalized entropies provide the knowledge of the state of anomalousness. Experiments have been conducted on different real video datasets. Experimental results show that Entropies among video frames on movers over time are a reasonable way to characterize abnormalcy as they change noticeably in case of abnormalities. Fig.3.27 outlines the framework.

3.7.2 Low-level features Extraction

3.7.2.1 Region of Interest (RoI) Estimation

Irrespective of indoor and outdoor video surveillance, RoI makes the video processing faster. Based on applications and type of videos, RoI would extend from few parts of a video frame to the whole frame. In case of applications, e.g., to monitor escalators, linear passages, high-way, etc., video processing region can be fixed by using a mask instead of analyzing the whole video frame. We build a RIIM or MM for such applications.

3.7.2.2 Modeling of Spatiotemporal Information (STI)

To analyze the scene, we treat moving interest points as the main cue instead of tracking individual subjects. The RoI, ascertained by MM, is divided into small blocks. Once we define n points of interest in the RoI, we track those points over the small blocks of two successive region of interest images using optical flow technique. We take down the static and noise features. Static features are the features which moves less than two pixels. Noise features are the isolated features which have

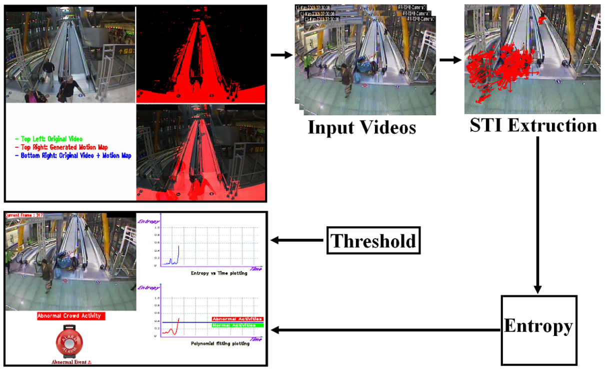


Figure 3.27: The summary of the proposed framework

a big angle and distance difference with their near neighbors due to tracking calculation errors. Yet, one broadly problem in some applications is that people near the camera are supposed to produce ample optical flow vectors and people far from the camera cannot produce such fully sufficient flow vectors even if they would make very quick motion (e.g., running or falling). That might be right in many examples but it does not generalize. For example, a fronto parallel wall has the same depth everywhere in the RoI, same for a person close to the camera. In the direction of generalization one reasonable solution would be a vertical coordinate system in the image. Moreover, authors in [180] used vertical coordinate to model their motion vector. We can count vertical coordinate system of each block where a weighing coefficient ζ is calculated according to the vertical coordinate of the block. Vertical coordinate system is an implementation stage coordinate system, it depends on several factors of the context of application and implementation e.g., area of RoI, number of defined blocks within RoI, etc. A weighing coefficient $\zeta \leq 1$ is calculated according to the vertical coordinate of the block. A block far away from the camera has small vertical coordinate, as a result its ζ should be large. Equally, block with large vertical coordinate gets smaller ζ. If we see with attention the applications which are related to fronto parallel wall, then ζ is just 1. Finally, for each video frame [e.g., Fig.3.28] irrespective of normal or abnormal events, we come into possession of a reliable and workable *spatiotemporal information* (STI), i.e., a $n \times 5$ matrix which is a function of time, broadly speaking a set of vectors $\mathbf{V}$ of n elements variate in time, formulated as:

$$\mathbf{V} = \begin{bmatrix} x_1 & y_1 & \delta_1 & \alpha_1 \\ . & . & . & . \\ x_i & y_i & \delta_i & \alpha_i \\ . & . & . & . \\ x_n & y_n & \delta_n & \alpha_n \end{bmatrix} \tag{3.86}$$

where $i \in n$, and

- $x_i \mapsto x$ coordinate of any feature element i,
- $y_i \mapsto y$ coordinate of i,
- $\delta_i \mapsto$ some weighing factor ζ_i is multiplied with the displacement of i from one frame to the next,
- $\alpha_i \mapsto$ moving direction of i.

We will use *displacement* and *vector length* interchangeably. As simple trigonometric function *atan* comes into notice few potential problems e.g., infinite slope, false quadrant, etc., the trigonometric function $atan2$ has been used to estimate the accurate moving direction α_i of the feature i. On the whole, the function $atan2$ gracefully handles infinite slope and places the angle in the correct quadrant [e.g., $atan(\frac{-1}{-1}) = \pi/4$ differs from $atan2(-1,-1) = -3\pi/4$, etc.].

3.7.3 Statistical Treatments of the STI

In this subsection, we will formulate *Entropy*, which is a measure of the disorder or randomness of video sequence, from its two crude elements namely degree of randomness of the directions (*circular variance*) and the degree of randomness of the displacements (*coefficient of displacement variation*).

3.7.3.1 Degree of Randomness of the Directions

Consider two cars on the high-way have changed directions with respect to their original directions, i.e., one from 0° to 10° and other from 0° to 340°. The *arithmetic means* of these pairs of direction changes are 5° and 170°, respectively. The direction mean 5° seems intuitively reasonable, while the average

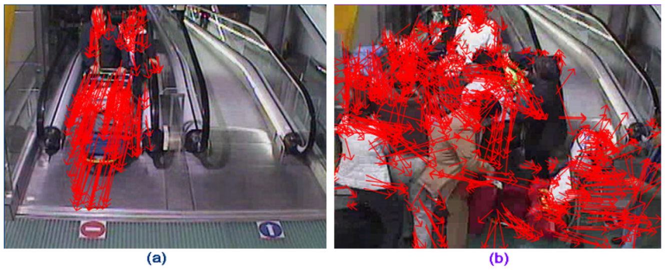


Figure 3.28: Optical flow: (a) monomorphically directed vector flows *normal case*, (b) haphazardly directed vector flows *abnormal case*. The more is the disorder/chaos presents in the video frame, the more is the *Entropy*; e.g., entropy of (b) is greater than that of (a).

of 170° is clearly in error. As the arithmetic mean is ineffective for angles, it is important to find a good method to obtain both the mean value and measure for the variance of the angles. Assume that two interest points of a frame went somewhere in the next frame with a maneuver of unit vector lengths **A** and **B** having angles α_1 and α_2, respectively. Their directional mean **R** can be found graphically as shown in Fig.3.29. But the graphical solution becomes extremely inefficient when a large number of directions to be added and also often arises the problem of precision. Yet an elementary trigonometric analysis can solve the problem with high accuracies. If $\alpha_1, \ldots, \alpha_i, \ldots, \alpha_n$, where $i \in n$, be a set of directions of n interest points taken from a single origin, then the *tangent of* **R**, symbolized as θ_R, can be defined by:

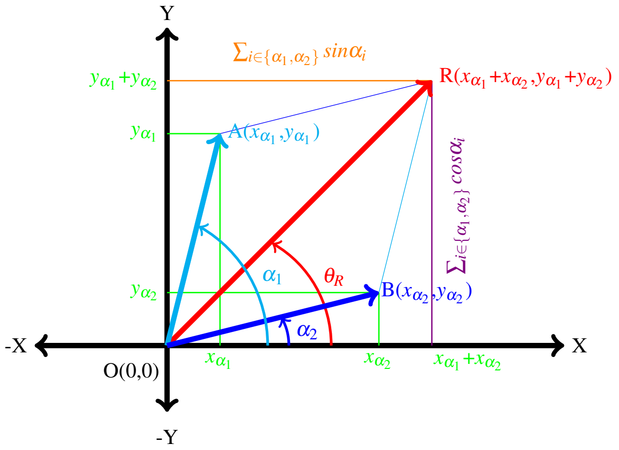

Y
$\sum_{i\in\{\alpha_1,\alpha_2\}} sin\alpha_i$
$y_{\alpha_1}+y_{\alpha_2}$
R($x_{\alpha_1}+x_{\alpha_2}$,$y_{\alpha_1}+y_{\alpha_2}$)
y_{α_1}
A(x_{α_1},y_{α_1})
$\sum_{i\in\{\alpha_1,\alpha_2\}} cos\alpha_i$
α_1
θ_R
y_{α_2}
B(x_{α_2},y_{α_2})
α_2
-X
X
O(0,0)
x_{α_1}
x_{α_2}
$x_{\alpha_1}+x_{\alpha_2}$
-Y

Figure 3.29: Elementary vectors and trigonometric analysis

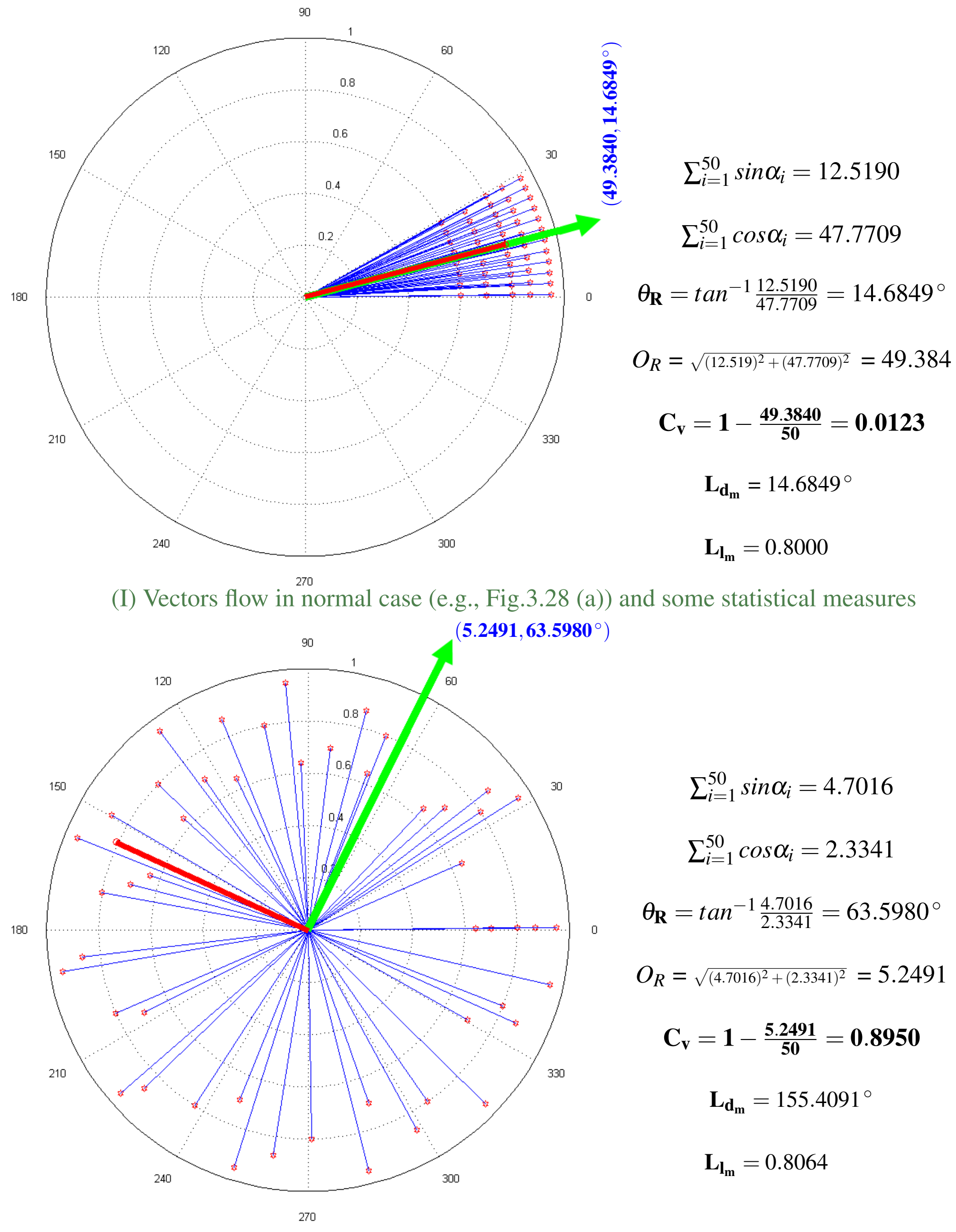
$$\theta_R = \begin{cases} tan^{-1}\frac{\sum_{i=1}^{n} sin\alpha_i}{\sum_{i=1}^{n} cos\alpha_i} & if \quad \sum_{i=1}^{n} sin\alpha_i > 0,\ \sum_{i=1}^{n} cos\alpha_i > 0 \\ tan^{-1}\frac{\sum_{i=1}^{n} sin\alpha_i}{\sum_{i=1}^{n} cos\alpha_i} + 180^\circ & if \quad \sum_{i=1}^{n} cos\alpha_i < 0 \\ tan^{-1}\frac{\sum_{i=1}^{n} sin\alpha_i}{\sum_{i=1}^{n} cos\alpha_i} + 360^\circ & if \quad \sum_{i=1}^{n} sin\alpha_i < 0,\ \sum_{i=1}^{n} cos\alpha_i > 0. \end{cases} \tag{3.87}$$

An interesting manner is that the sum of the *sines* of the angular deviations from each observation to the resultant is *zero*, mathematically this property can be shown as: $\sum_{i=1}^{n} sin(\alpha_i - \theta_R) = 0$. The variability of a sample of directional measurements is indicated by the length of **R**, which can be defined for n vectors using Pythagorean theorem as: $o_R = \sqrt{(\sum_{i=1}^{n} sin\alpha_i)^2 + (\sum_{i=1}^{n} cos\alpha_i)^2}$ which means the larger sample sizes can have longer resultant lengths than smaller samples without having less variability. A standardized measure of variability can solve this unacceptable property. To develop such a measure of variability it is necessary to account for differing sample sizes. Let α_i be a set of directional measurements with sample size n where $i \in n$, then the *degree of randomness of the directions* or *circular variance* C_v is defined as:

$$\mathbf{C_v = 1 - \frac{\sqrt{(\sum_{i=1}^{n} \sin\alpha_i)^2 + (\sum_{i=1}^{n} \cos\alpha_i)^2}}{n} = 1 - \frac{O_R}{n}}. \tag{3.88}$$

The $\mathbf{\frac{O_R}{n}}$ ranges from 0 to 1. Its extreme values have some agreeable properties. The case $\mathbf{\frac{O_R}{n} = 1}$ implies that all the data points are coincident, whereas $\mathbf{\frac{O_R}{n} = 0}$ does not imply uniform dispersion around the circle. Therefore, $\mathbf{\frac{O_R}{n}}$ is not necessarily a useful indicator of dispersion or spread of the data unless they constitute a single group. The $\mathbf{C_v}$ provides a smooth (0,1) scale. The smaller is the value of $\mathbf{C_v}$, the more is the concentration of distribution. It is worth mentioning that $\mathbf{0 \leq C_v \leq 1}$, unlike an ordinary *linear variance*; and the interpretation of $\mathbf{\frac{O_R}{n} = 0}$, the estimation of $\mathbf{C_v = 1}$ does not necessarily imply a maximally dispersed distribution.

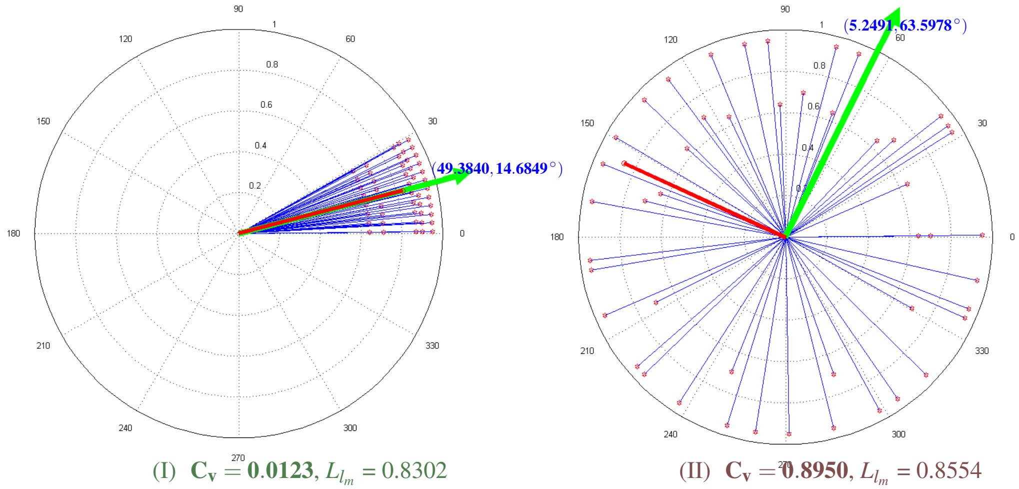
Forthwith, we wish to pay our attention on: *How differently does the circular variance behave in normal and abnormal situations?* Superposable to the observation of Fig.3.28, where directions and displacements of interest points vary randomly in abnormal case and they are almost symmetrically directed in normal case, we have simulated the two cases in simpler way. Fig. 3.30 depicts the fate of 50 interest points for two cases. The directions of interest points have been simulated in between 0° and 30° with their vector lengths between 0.5 and 1 for normal case. While in abnormal case, directions vary in between 0° and 360° with vector lengths between 0 and 1. On account of simplicity outlier has not been taken into account. Both linear and circular measures have been estimated in each circumstances. In the symmetrically directed directions case, there is almost no pressure on the choice of which preferred direction, either linear or circular, is to be used because they both perform similarly. In other words, in normal case, either linear or circular direction can be the preferred direction painlessly. In spite of that, circular measure is preferable because of its more accurateness. The circular variance $\mathbf{C_v} = 0.0123$ illustrates that the flow vectors of the interest points are well concentrated and the interest points are systematically directed. Emphatically, the movements of crowd in video for normal cases are hazard free. On the other hand, there is sufficient difference between linear and circular measures in abnormal case. Nevertheless, the linear mean goes wrong and only choice is the circular measure. The circular variance $\mathbf{C_v} = 0.8950$ exemplifies that the flow vectors of the interest points are highly scattered around. Intuitively speaking, the movements of crowd in video for abnormal cases are full of hazard. Heretofore, we can conclude in a gross manner that the circular variance varies consequentially in abnormal circumstances.



(I) Vectors flow in normal case (e.g., Fig.3.28 (a)) and some statistical measures

(II) Vectors flow in abnormal case (e.g., Fig.3.28 (b)) and some statistical measures

Figure 3.30: A simple example of how the circular variances behave in normal and abnormal cases. *Linear mean of directions* (L_{d_m}) and *circular resultant vector lengths* (O_R) are shown using heavy red line and heavy green arrow, respectively. Unlike O_R, the *linear mean of vector lengths* (L_{l_m}) is normalized. There is a significant variation in C_v between normal and abnormal situations.



(I) $\mathbf{C_v} = \mathbf{0.0123}$, $L_{l_m} = 0.8302$

(II) $\mathbf{C_v} = \mathbf{0.8950}$, $L_{l_m} = 0.8554$

Figure 3.31: Linear mean of vector length (L_{l_m}) varies with the length variation of interest point. Conversely, there is no effect on C_v, O_R, and L_{d_m}. Comprehensibly, circular variance does not change with vector lengths variation of interest points but does vary only their directions variation.

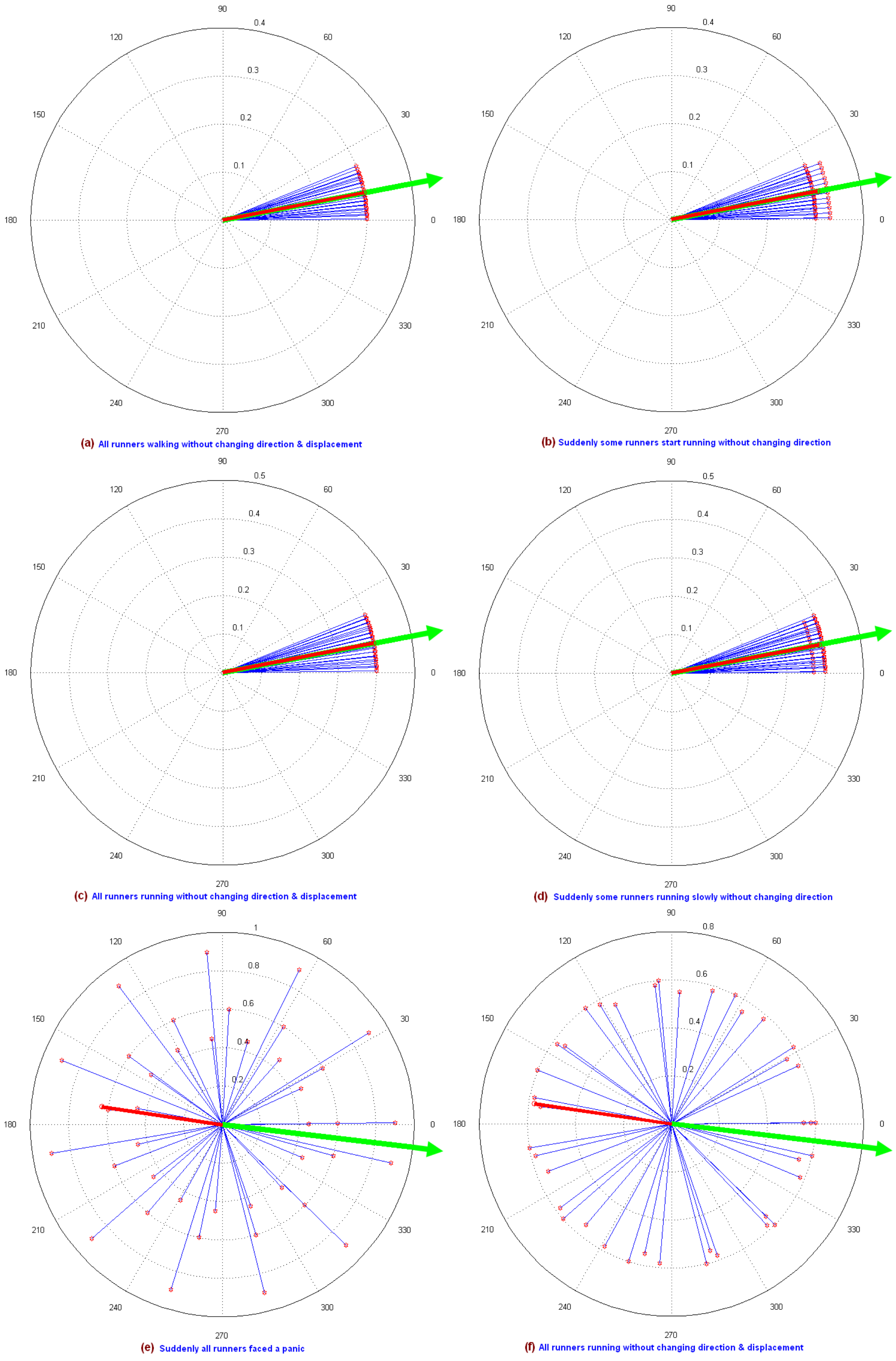
How does the circular variance behave, if some (or all) points will move slower or quicker than those of previous frame without changing their directions in the next frame? Normally, vector lengths in running case are larger than that of walking. What does happen, in real world crowd video scenes, if some persons will stop or start running suddenly without changing their direction of movements? Based on the context both situations would be abnormal. For example, some persons stopped running while Marathon running or some persons started running while others walking. Do these situations concern with the circular variance, any way? Let us take into account the vector length variation while direction remains unchanged in both cases of Fig. 3.30. Images in Fig.3.31 (I) and (II) depict circumstances where some interest points changed their vector lengths only. The estimated circular variances, circular resultant vector lengths, and linear mean of directions continue the same as estimated in Fig.3.30, solely the linear mean of vector lengths has been changed from 0.8000 to 0.8302 and from 0.8064 to 0.8554 for normal and abnormal cases, respectively. From this estimation, it is easy to show that any change of the vector length without varying their directions, the circular variance remains unaffected. Without any shadow of doubt, we can conclude that the circular variance does not bear any information when some persons stopped running while Marathon running or some persons started running while others walking, if and only if the direction of movement be the same. From this knowledge of observation, we can reach a conclusion that the circular variance is an extremely important factor for direction changing case but exclusively it is not always adequate to pick up abnormality from the real world video scenes. Henceforth, it needs its complement for detecting wide varieties of aberration.

3.7.3.2 Degree of Randomness of the Displacements

We have observed that circular variance is a necessary factor but not sufficient for detecting abnormality from the real world videos where both systematic and unsystematic movements exist. Along with circular variance, it is important to take into consideration the vector lengths or displacements of the point of interests for exemplifying the abberation detection purposes.

One common query would be: *How does the displacement behave in normal and abnormal cases*? To accord the answer in a good way, let us simulate six different instances of the occurrence of a straight avenue race (e.g., Marathon) and the number of participating runners is 40. Beginning of the run all runners were walking with some 0.30 unit displacement per frame without changing their directions as simulated in the Fig. 3.32 (a). In real world scene, this type of event usually holds up no surprisal and thus it is normal. At certain frame, suddenly some runners started running with some 0.33 unit displacement per frame without changing their directions as simulated in Fig.3.32 (b). Such type of event poses some degree of visual attention for the primates and accordingly it would be abnormal. Afterwards, all runners were running with some 0.40 unit displacement per frame without changing their directions as simulated in Fig. 3.32 (c). It is a usual event like 3.32 (a) as very systematic run or walk does not sustain interesting facts. After a while, some runners grew fatigued and at certain frame suddenly they decided to run slowly at 0.37 unit displacement per frame without changing their directions as simulated in Fig.3.32 (d). Such type of change in the crowd has connection with some interesting information for the primates and in this way it would be an abnormal event. At certain frame, all runners faced a sudden panic situation (e.g., explosion, gun shot, fire) and accordingly they were randomly scattered, i.e., they changed their directions as well as displacements as simulated in Fig.3.32 (e). This type of variation in the crowd bears very high degree of interest for the primates and to this extent it is necessarily an abnormal event. After the explosion, all the scattered runners were running without changing their directions and displacements (maybe varying displacements with respect to others but constant for each runner) over frames as simulated in Fig.3.32 (f). This event is similar to Fig.3.32 (a) and (c) and does not endorse interesting information and consequently it is normal.

From Fig.3.32 it is noticeable that if an event where both direction and displacement vary, then



(a) All runners walking without changing direction & displacement

(b) Suddenly some runners start running without changing direction

(c) All runners running without changing direction & displacement

(d) Suddenly some runners running slowly without changing direction

(e) Suddenly all runners faced a panic

(f) All runners running without changing direction & displacement

Figure 3.32: Simulation of six different instances of the occurrence of an avenue race (e.g., Marathon).

there will be high possibility to become that event an abnormal. So it is important to consider carefully both direction and displacement simultaneously. The directional measure circular variance (C_v) is both dimensionless and normalized. On the other hand, displacement is neither a dimensionless nor a normalized quantity. Henceforth, we put forward a reasonable solution of these problems in a different way by taking ratio between two statistical measures of displacements. The displacement variance to mean ratio would be a good solution. Customarily, variance to mean ratio is a measure used to quantify whether a set of observed occurrences are clustered or dispersed compared to a standard statistical model. It provides a good measure of the degree of randomness of the displacements and may be dealt with normalization. But the variance of a variable has different units from the variable, for example square centimeters when the variable is in centimeters. As a result, the displacement variance to mean ratio has unit of centimeter. Since the displacement variance to mean ratio is dimensional, the unit does not cancel, the ratio is not scale invariant. Scale invariance is a feature of rules which do not change if length scales are multiplied by a mutual factor. One possible good solution of the scale invariance for this problem would be the standard deviation (the square root of variance) which is a widely used measure of the variability or dispersion. A useful property of standard deviation is that, unlike variance, it is expressed in the same units as the data (using the *mean* as a measure of scale). The unit of the displacement standard deviation to mean ratio is canceled out as they are measured in the same scale and is thus a pure number. Evidently, the obtained ratio is now scale invariant. The displacement standard deviation to mean ratio (coefficient of displacement variation) is not only a dimensionless quantity but also can provide a good measure of the degree of randomness of the displacements. Having a complement factor of circular variance for a wide variety of aberration detections, the coefficient of displacement variation plays an important role to detect some kind of abnormal activities from real world videos.

Considering Eq. 3.86, the *mean* of displacements $\overline{\delta}$ is delimited by dint of:

$$\overline{\delta} = \frac{1}{n}\sum_{i=1}^{n}\delta_i \tag{3.89}$$

where n is the number of optical flow vectors in the frame. With this *mean* it is easy to ascertain displacement of *standard deviation* by means of:

$$\delta_{std} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\delta_i - \overline{\delta})^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\delta_i^2 - \frac{n}{n-1}\overline{\delta}^2}. \tag{3.90}$$

The displacement standard deviation to mean ratio or *degree of randomness of the displacements* is formulated as the ratio of the standard deviation to the mean with the help of:

$$D_r = \frac{\delta_{std}}{\overline{\delta}} \tag{3.91}$$

where $\overline{\delta} > 0$. Accordingly, D_r is scale invariant and normalized particularly for positive distribution such as the exponential distribution and Poisson distribution.

3.7.4 Entropy Estimation

Up until now, it is clear that *circular variance* and *coefficient of displacement variation* are necessary and sufficient factors to detect various aberrations in videos. How can the effective power of them get mixed together? One possible solution would be the usage of those statistical measures as the crude parameters of the *Entropy*. The more is the entropy, the more is the disorder/chaos in the system. For instance, to have order on the high-way means to have cars follow the order of lanes, speed limits, directions, etc. When these things get mixed in, entropy increases causing disorder/chaos on the high-way traffic system.

Thermodynamic entropy indicates a measure of how organized or disorganized a system of atoms or molecules is. It has an enabling factor of energy. Information (Shannon) entropy with no inherent or integral energy factor, thus it is solely related in form and not in function. Shannon entropy, a measure of uncertainty, is the expectation value of $-ln(p)$, where p is the probability assigned to the measured value of a random variable. Shannon entropy is a broad and general concept which finds applications in information theory and thermodynamics. Shannon entropy and information uncertainty can be used interchangeably [88]. Definition of the Shannon entropy E_S is, quite usual, and is expressed in terms of a discrete set of n probabilities p_i with $i \in n$ as:

$$E_S = p(x_1)log\frac{1}{p(x_1)} + p(x_2)log\frac{1}{p(x_2)} + \cdots + p(x_n)log\frac{1}{p(x_n)} = -\sum_{i=1}^{n} p(x_i)log\, p(x_i) \tag{3.92}$$

where $\sum_{i=1}^{n} p(x_i) = 1$. If $p(x_i) = 0$ for some i, the value of the corresponding summand $0log0$ is taken to be 0. The entropy is zero signifies there is no uncertainty and hence there is no information. Consequently, entropy always follows the nonnegativity rule ($E_S \geq 0$).

To fit for the statistical measures of C_v and D_r in the Eq. 3.92, the measures have been modeled with their respective probabilities as:

$$p(c_v) = \frac{C_v}{C_v + D_r} \tag{3.93}$$

$$p(d_r) = \frac{D_r}{C_v + D_r}. \tag{3.94}$$

Then the Shannon entropy at some frame f can be formulated as:

$$E_f = p(c_v) log \frac{1}{p(c_v)} + p(d_r) log \frac{1}{p(d_r)} \tag{3.95}$$

where $p(c_v) + p(d_r) = 1$. The more is the E_f, the more is the disorder/chaos in the video frame. Higher value of E_f means the corresponding video frame has a high possibility to become a frame of abnormal situation. The $E_f = 0$ means the video frame bears no information and we are no longer interested with that. To define and usage the Shannon entropy by Eq. 3.95 is an agreeable way. Up to this point, we would apply a threshold on the obtained E_f measure to get a decision whether the frame belongs to normal or abnormal situations. But the Shannon entropy is not normalized, i.e., Eq. 3.95 needs a little correction to have a normalized structure. For instant, consider a probability space where exists $p(x_1) = 0.51$, $p(x_2) = 0.26$, and $p(x_3) = 0.23$; then Eq. 3.92 estimates $E_S = 1.0317$, which is not normalized. For the sake of normalization, we may use the function $1/(1 + ln E_f)$ but it does not offers a friendly change option of E_f measure for the user, i.e., scaling problem. To solve the scaling problem, we would like to take up a versatile distribution which has significant effect on its shape and scale parameters. In this respect, we take advantage of *cumulative distribution function* (cdf) of *Weibull distribution* [174] which has strict lower and upper bounds between 0 and 1. Due to accurate model quality and performance characteristics of Weibull distribution and its flexibility that makes it ideal for analysis on a dataset with unknown distribution. It is worth mentioning that Weibull distribution can mimic the behavior of other statistical distributions such as the normal and the exponential. Now, we can formulate the *normalized entropy* of some frame f as:

$$[Entropy]_f = 1 - e^{-(E_f/\lambda)^{\nu}} \tag{3.96}$$

where $\nu > 0$ is the shape parameter and $\lambda > 0$ is the scale parameter of the distribution. Using Eq. 3.96, and knowing the values of ν, λ, and E_f we can desirably estimate the normalized entropy of a frame f, $[Entropy]_f$ between 0 and 1. As a result the Weibull distribution not only provides a fair normalized measure for E_f between its strict lower and upper bounds but also offers a friendly change option of that measure for the user by its shape (ν) and scale (λ) parameters. With the help of 3.96, we can fully and clearly estimate *Entropy* of the simulated situations as simulated on Fig. 3.30, 3.31, and 3.32. Table 3.1 demonstrates the estimated results where user friendly parameters have been selected as $\nu = 2$ and

$\lambda = 0.5$.

Table 3.1: *Entropy estimation of the simulated situations as simulated on Fig.3.30, 3.31, and 3.32.*

Different Cases	C_v	D_r	$p(c_v)$	$p(d_r)$	E_f	$[Entropy]_f$	Remarks
Fig. 3.30 (I)	0.0123	0.1525	0.0748	0.9252	0.2658	0.2462	Normal
Fig. 3.30 (II)	0.8950	0.1386	0.8659	0.1341	0.3940	0.4626	Abnormal
Fig. 3.31 (I)	0.0123	0.1399	0.0809	0.9191	0.2810	0.2709	Normal
Fig. 3.31 (II)	0.8950	0.1570	0.8508	0.1492	0.4214	0.5085	Abnormal
Fig. 3.32 (a)	0.0063	3.7992×10^{-6}	0.9994	6.0069×10^{-4}	0.0051	1.0225×10^{-4}	Normal
Fig. 3.32 (b)	0.0063	0.0451	0.1230	0.8770	0.3729	0.4266	Abnormal
Fig. 3.32 (c)	0.0063	2.8494×10^{-6}	0.9995	4.5058×10^{-4}	0.0039	6.1533×10^{-5}	Normal
Fig. 3.32 (d)	0.0063	0.0356	0.1508	0.8492	0.4240	0.5128	Abnormal
Fig. 3.32 (e)	0.9735	0.2959	0.7669	0.2331	0.5430	0.6926	Abnormal
Fig. 3.32 (f)	0.9735	0.0351	0.9652	0.0348	0.1510	0.0871	Normal

We can apply a threshold on the obtained *Entropy* measures data to get a decision of normal or abnormal event frame. But any discrete value of *Entropy* which exceeds a predefined threshold T_E is not a clear evidence of abnormal event frame. It may frequently fear that at least one attribute (an outlier) may have been severely corrupted by a mistake or error (e.g., tracking calculation errors) which would lead an erroneous decision of the normal or abnormal event frame. An outlier is a sample that is very different from the average sample in the data set. An outlier may be an ordinary sample, but of which at least one attribute has been severely corrupted by a mistake or error (e.g., tracking calculation errors). An outlier may also be a bona fide sample, that simply turns out to be exceptional. To minimize this outlier problem, a polynomial fitting would be a good solution. Runge's phenomenon [152] shows that lower-order polynomials are normally to be preferred instead of augmenting the degree of the interpolation polynomial, even if some of the badness of this interpolation may be overcome by using Chebyshev polynomials instead of equidistant points. Accordingly, we can apply some lower degree (e.g., 5) of polynomial fitting on the obtained *Entropy* measures data. As a consequence a more reliable, workable, palatable, and much less erroneous measures over the originally obtained *Entropy* measures data for a decision of normal or abnormal frame can be gained.

3.7.5 Threshold Estimation

The decision of normal or abnormal frame can be taken either *static way* by comparing with polynomial fitting data with a predefined threshold T_E or *dynamic way* by detecting considerable sudden changes of the polynomial fitting data over time. In static way, a predefined threshold T_E, calculated from video which contains exclusively normal activities, can differentiate each frame with respect to its estimated $Entropy$ whether it is normal or exceptional. An abnormal frame can be detected *if* & *only if* $Entropy > T_E$, otherwise normal frame. The T_E (also $Entropy$) depends on the controlled environment (video stream V_s), specifically the remoteness of the camera to the scene, the orientation of the camera, the type and the position of the camera, lighting system, density of the crowd, etc. In general, the more is the remoteness between the camera and the scene, the less is the considerable amount of optical flows and blobs. In case of escalator, T_E also places trust on the escalator type and position. Looking on these facts, we have at least one threshold for a video stream. If we have $\mathscr{N}$ video streams, which are the case in sites e.g., airports, malls, banks, subways, stations, hospitals, hotels, schools, concerts, cinema halls, parking places, sporting events, political events, town centers, etc., then we put forward at least $\mathscr{N}$ thresholds. If the video stream V_{s-1} (where s-1 $\in \mathscr{N}$) leaves for another V_s (where $s \in \mathscr{N}$), then the threshold T_E of V_s will be made over by means of:

$$[T_E]_{V_s} = arg \max_{f=1...m} [Entropy]_f + arg \min_{f=1...m} \left[\frac{1}{(2\pi)^2} \sum_{k=0}^{\infty} \frac{(-1)^k (Entropy)^{2k+1}}{k!(2k+1)} \right]_f \tag{3.97}$$

where m is the number of frames in the video V_s and second term indicates some minimum *Gaussian error*, which is added for a good estimation of the threshold.

3.7.6 Experimental Results and Discussion

In this subsection, we have presented experimental results a bit detailed as compared to other approaches. To conduct experiments, we have mainly used the *Escalator dataset* [132] and the two datasets as operated by [131] so called, respectively, the *UMN dataset* [137] and the *Web Dataset* [131]. Routinely ζ limits $0.65 \leq \zeta \leq 1$ and $n = 2000$. Adjacent to camera region, $\zeta = 0.65$ suits well while ζ bears 1 at opposite end. Shape and scale parameters have been friendly selected as $\nu = 2$ and $\lambda = 0.5$, consecutively.

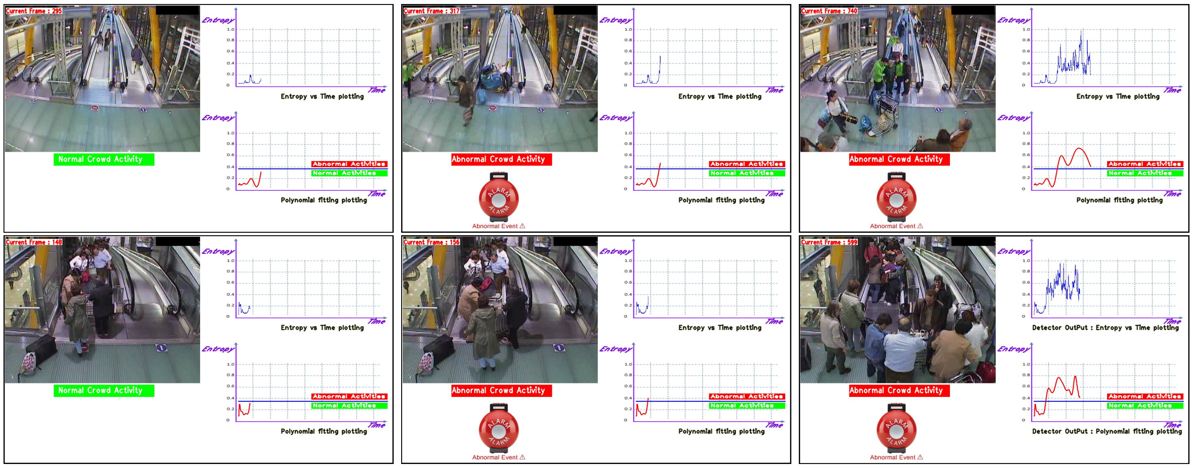


Figure 3.33: Two sample videos from the escalator dataset: first row concerns a person falling episode on the escalator egress; second row presents an aberrant situation caused by a wheel broken trolley.

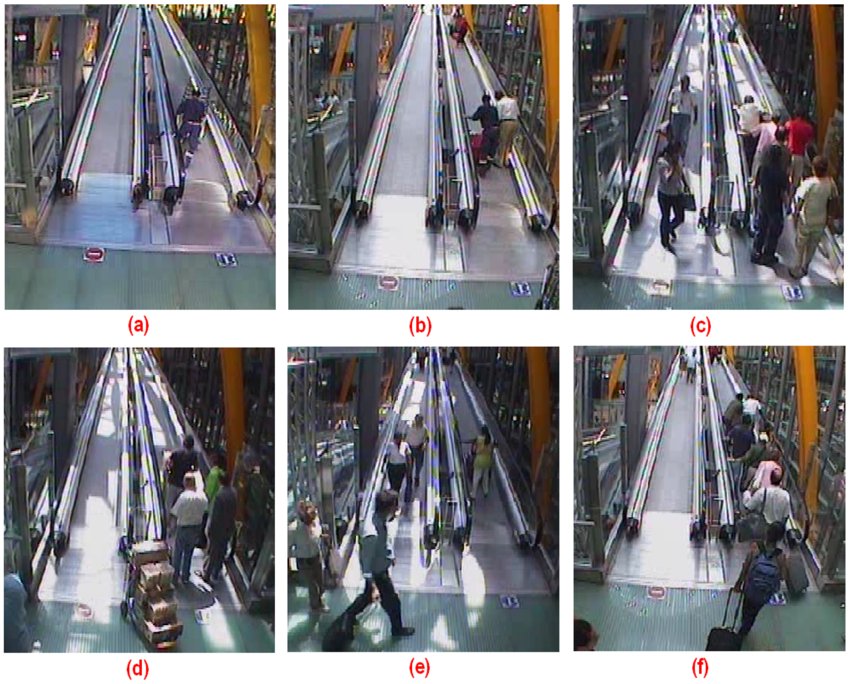


Figure 3.34: Method has hardly effect on handling occlusion anomalies e.g., (a), (b), (c), (d), (e), (f).

3.7.6.1 The Escalator Dataset

The complete *Escalator dataset* [132] consists of 29 real videos, taken in spanning days and seasons, of frame size 640×480 pixels, collected by cameras installed in an airport to monitor especially the escalator exits, provided by a video surveillance company[2]. The videos were used to provide informative data for the security team. Each video stream consists of normal and abnormal events. The normal situations correspond to crowd flows without any eccentric event on the escalator elsewhere. Eccentric events correspond to videos which contain collapsing events mostly in the escalator egresses. Generally, in the videos we have two escalators corresponding to two-way-traffic of opposite directions. Images in Fig.3.33, are the output of the abnormal event detector, depict two crowd scenarios of collapsing events on the escalator exits. First row (v_1 listed on the Table 3.2) depicts a scenario where two persons were standing on the moving escalator and suddenly a trolley became unbalanced and rushed out toward them. One person got away by running and was not run down under the force of trolley, while other was ill-fated. Hence the non-escapee was run down by the runaway trolley, and subsequently fell down at the exit point of the moving escalator. Second row (v_2 listed on the Table 3.2) describes another inconsistent circumstances on the exit point where a wheel from the trolley has suddenly been broken off by the friction during its travel over the escalator. Most of the inconsistent situations were detected by the proposed approach. The detailed evaluation of the proposed algorithm considering static method of thresholding for the provided escalator dataset has been listed on the Table 3.2. The algorithm has scarcely effect on the video streams 6th, 9th, 14th, 17th, 22nd, and 28th listed on the Table 3.2 as shown their sample frames in Fig.3.34 (a), (b), (c), (d), (e), and (f), respectively. This is due to the fact that the video sequences include abnormal events occur with occlusion. Thus the estimated $Entropy$ obtained from the quantity of extracted information is insufficient to draw out anomalous frames. It is well known that occlusion handling is an arduous part of optical flow technique. The fact is that occluded pixels violate a major assumption of optical flow that each pixel goes somewhere. On the Table 3.2, except six videos, the first detected abnormal frame D_{V_s} of some video V_s has been compared with the respective ground truth G_{V_s} and thereof *root mean squared error* Ψ and *mean absolute error* Φ have been estimated for 23 out of 29 videos. Ground truth is the process of manually marking what an algorithm is expected to output. The estimation of $\Psi = 0$ and $\Phi = 0$ corresponds to perfect detection or ideal case or ground truth. However, the estimated $\Psi \approx 5$ and $\Phi \approx 5$ fall within the fitting range of many computer vision applications along with escalators.

[2] Thanks to the MIAUCE project, the EU Research Programme (IST-2005-5-033715).

Table 3.2: *Performance evaluation of the method using escalator dataset.* G_{V_s} *and* D_{V_s} *mark ground truth and first detected atypical frames of some video* V_s*, respectively.* $[T_E]_{V_s}$ *denotes* T_E *of* V_s*.*

Various Measures	Video Streams $(V_s)_{s=1...29}$																												
	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}	V_{15}	V_{16}	V_{17}	V_{18}	V_{19}	V_{20}	V_{21}	V_{22}	V_{23}	V_{24}	V_{25}	V_{26}	V_{27}	V_{28}	V_{29}
D_{V_s}	309	155	24	93	16	(a)	37	92	(b)	169	95	183	256	(c)	338	29	(d)	147	139	55	123	(e)	80	63	212	49	207	(f)	119
$[T_E]_{V_s}$	.38	.36	.34	.41	.36	.43	.39	.40	.42	.35	.37	.44	.37	.39	.42	.38	.33	.37	.43	.35	.33	.44	.45	.42	.38	.39	.41	.45	.37
G_{V_s}	312	158	28	99	13	54	31	97	62	175	99	179	261	923	331	33	553	151	144	52	128	71	86	67	217	56	211	411	125
Δ_{V_s}	9	9	16	36	9	—	36	25	—	36	16	16	25	—	49	36	—	16	25	9	25	—	36	16	25	49	16	—	36
Mean Errors	***Root Mean Squared Error*** $(\Psi) = \sqrt{\frac{1}{29}\sum_{s=1}^{29}\Delta_{V_s}}$ where $\Delta_{V_s} = (G_{V_s} - D_{V_s})^2 \Rightarrow \Psi = \sqrt{\frac{571}{23}} \approx 5$ ***Mean Absolute Error*** $(\Phi) = \frac{1}{29}\sum_{s=1}^{29}\lvert G_{V_s} - D_{V_s}\rvert \Rightarrow \Phi = \frac{111}{23} \approx 5$																												

3.7.6.2 The UMN Dataset

The publicly available dataset of normal and abnormal crowd videos from University of Minnesota [137] comprises the videos of 11 different scenarios of an escape event in 3 different indoor and outdoor scenes. Each video (frame 320×240 pixels) consists of an initial part of normal behavior and ends with sequences of the abnormal behavior. The qualitative results of the abnormal behavior detection for four sample videos (we named d_1, d_2, d_3, d_4 from top to bottom) of UMN dataset have been presented in the Fig. 3.35. In all the sample videos, abnormal motion includes a sudden situation when the group of people start running the measured *Entropy* will be higher than that of any other before estimated. Gaussian like curves present the abnormal motions when those groups of people are trying to leave their places with atypical motions. Results report that the proposed method performs something to a greater degree to distinguish abnormal sequences. The results are likely a bit superior to [131] in the sense that there is no reported false positives on the proposed method. Table 3.3 provides the quantitative results of a comparison with Mehran et al.$'$s [131] results for the same four sample videos.

3.7.6.3 The Web Dataset

We also conducted the experiment on the same challenging set of videos that has been used by [131] and collected from the sites like Getty Images and ThoughtEquity (http://www.thoughtequity.com) which contain documentary and high quality videos of crowds in different urban scenes. The dataset encompasses 12 sequences of normal crowd scenes such as pedestrian walking, marathon running, and 8

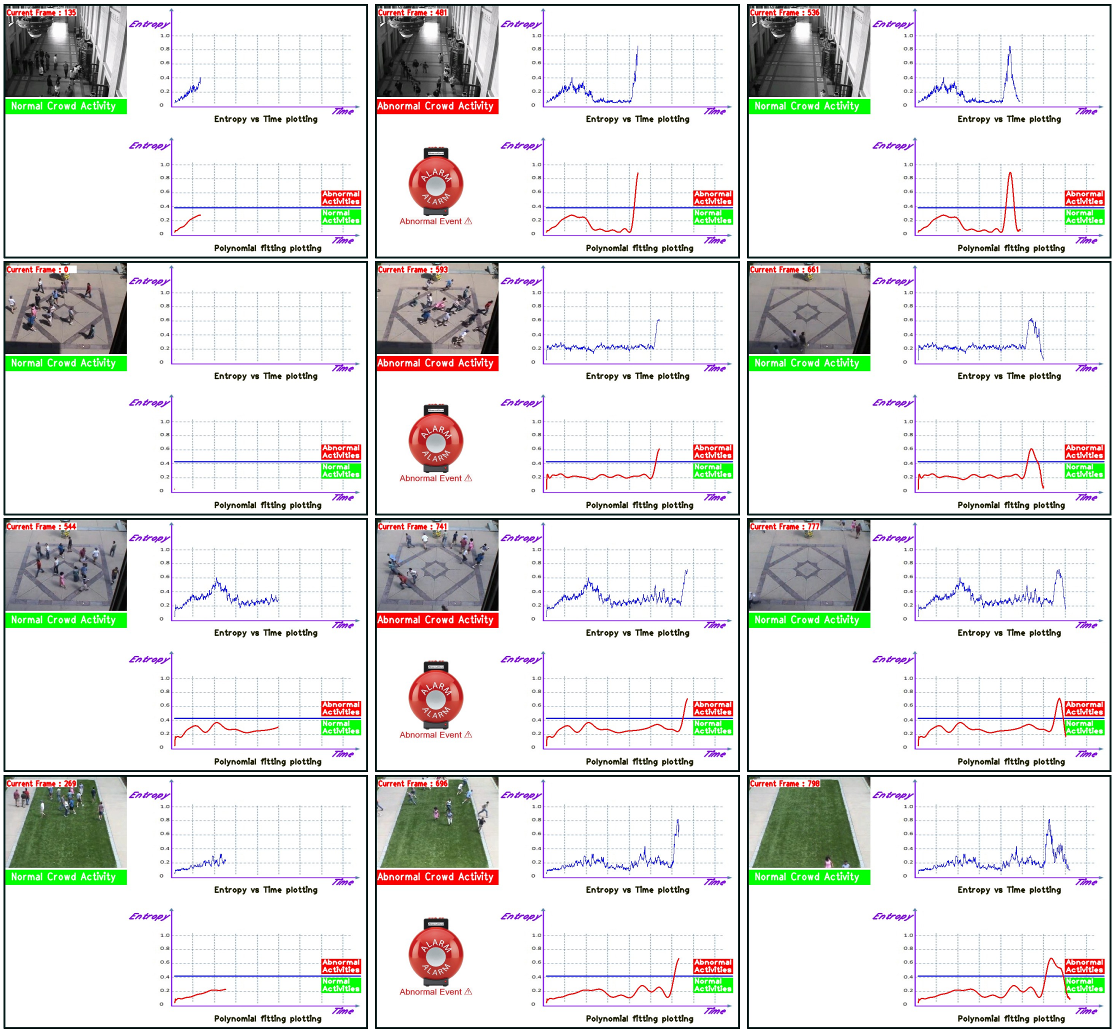


Figure 3.35: Qualitative results of abnormal behaviors detection using the proposed framework for the same four sample videos as shown in [131] from the UMN dataset.

Table 3.3: *Comparison of Mehran et al.'s [131] results*

Approaches	d_1	d_2	d_3	d_4	Ψ	Φ	False Positives
Mehran et al. [131]	482	593	741	696	16	16	**6**
Proposed	461	576	718	671	6	6	**0**
Ground Truth case	466	581	724	678	0	0	**0**

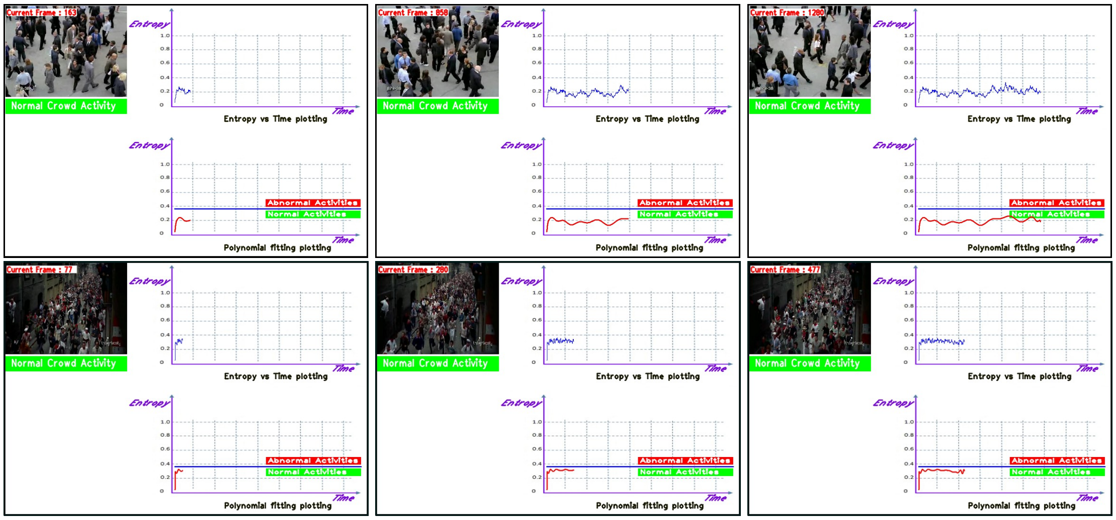


Figure 3.36: Qualitative results of normal behaviors detection. First and second rows concern normal activities of pedestrian walking and marathon running, respectively.

scenes of escape panics, protesters clashing, and crowd fighting as abnormal scenes. All frames have been resized to the 320 × 240 pixels.

Fig.3.36 shows the qualitative results of normal behaviors detection from two sample videos of the Web dataset [131]. The videos concern pedestrian walking and marathon running. Fig.3.37 shows the qualitative results of abnormal behaviors detection from three sample videos of the Web dataset [131]. The videos bear reference to crowd fighting on the street and overwhelming feeling of fear and anxiety.

Beyond the crowd aberrant activities detection, the algorithm can monitor illegal traffic activities on the high-way, e.g., car making illegal U-turn. Fig.3.38 depicts an illegal U-turn situation which has been picked up by the algorithm. Since the approach considers circular variance C_v (as discussed in 3.7.3.1), it is too accurate and reliable to report any angular change as compared to linear measure.

3.8 Discussion

We have adopted six approaches namely Covariance (3.2), NCRIM (3.3), Mahalanobis metric (3.4), Bhattacharyya metric (3.5), Enumerated Entropy (3.6), and Shannon Entropy (3.7) by first performing a global-level motion analysis within each frame's region of interest that provides the knowledge of crowd's multi-modal behaviors in the form of complex spatiotemporal structures. These structures are

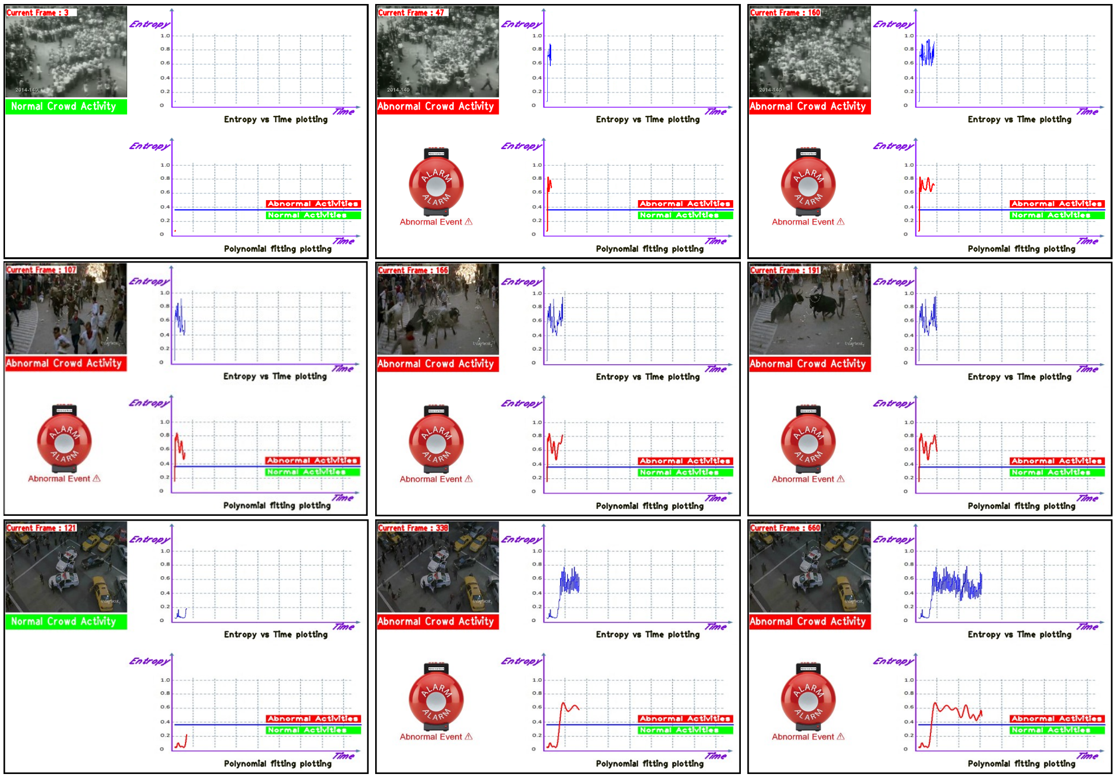


Figure 3.37: Qualitative results of abnormal behaviors detection. The 1st row demonstrates crowd fighting on the street, while the 2nd and 3rd rows touch upon escape panics.

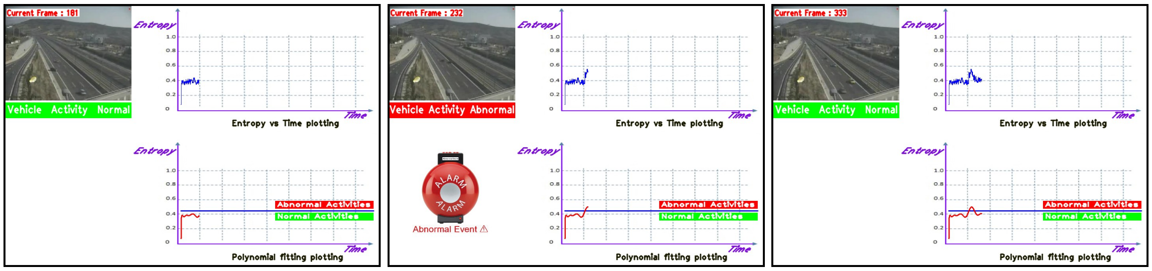


Figure 3.38: Example video in which cars are ensuing the regular traffic flow which hints that Entropies are normal; while a car making an illegal U-turn which infers that Entropies are higher and consequently the illegal traffic activity has been picked up by the pointed approach.

then employed in the detection of unusual surveillance events within crowds. The global level analysis eliminates the need for low level change detection algorithms.

We have employed three types of region of interest namely *motion heat map* (MHM), *spatiotemporal region of interest* (ST-RoI), and *region of interest image map* (RIIM). Basically, their functions are the same, only difference in construction. The approaches have been tested on videos of single camera data-set. To conduct experiments, we have primarily relied on the *Escalator dataset* [132], *UMN dataset* [137], and *Web dataset* [131]. Ignoring their own restrictions, the average performance of the approaches is nearly close to each other. However, the approach Shannon Entropy has been test both simple simulated data and real data-sets and can be taken into account the most effective one.

3.8.1 Pros and Cons of Different Approaches

Since all of the approaches are based on spatiotemporal information, hence there are some unique global pros and cons reflect on them. The main noticeable excellences and disadvantages are listed below.

Advantages: The main benefits of the proposed frameworks are stated as:

- They do not expect low level change detection algorithms
- They are simple and easy to understand and implement
- They do not need explicit learning process and training data
- They work all directional flow of movers without imposing a condition of their numbers in videos
- They reduce processing time by considering a region of interest

Disadvantages: There are three main disadvantages widely be seen on the proposed approaches as:

- They expect a predefined threshold, which maybe either static or dynamic, to make a decision.
- They are based on optical flow concept, where occlusion handling is not considered and hence the approaches do not pay any hint on occlusion handling.
- They detect abberations but do not localize where the abnormalities on the video frames are.

3.8.2 Comparison with Internal Issues of Different Approaches

In this subsection, we have shown a comparison which is based on various internal issues of the different approaches, as listed on the Table 3.4. The developed algorithms accommodate some of the challenges encountered in videos of crowded environments to a certain degree, while challenge like occlusion does not concern at all. In the following, we have explained briefly how the developed algorithms accommodate some of the challenges encountered in videos of crowded environments to a certain degree.

3.8.2.1 Depiction of abnormality

Interactions between people are indiscernible in crowded scenarios, and as a result, individual centric representation of abnormal events in crowds is very unlike. Furthermore, an abnormal event or situation in a high density crowded scene often spreads very abruptly, which makes it even more challenging to develop a general appreciation of the abnormal situation by gleaning information from an individual's behavior. Some existing works like Mehran et al. [131] localized the abberation on the video frames, yet their final output is globally marked each frame either normal or abnormal. We have considered a frame based evaluation of abnormal event without locating its real position on the frame. Having the knowledge of threshold from the context of normal videos, if a frame exceeds the required threshold level, then it is encountered as an abnormal frame. This helps to minimize the depiction of abnormality in certain degree. For example, people running in Marathon is normal, if we have a standard threshold of running level, then that could be apply to detect people walking level and vice versa.

3.8.2.2 Threshold estimation

In our approaches, the decision of normal or abnormal frame has been taken into account in the static way. A predefined threshold has differentiated each frame with respect to its estimated distance value. The theoretical aspect of estimating such threshold is that we have considered the maximum number of estimated distance value in large videos those contain exclusively normal events and then added with it some minimum *Gaussian error*, which helps to make a better estimation of the threshold. This threshold technique minimizes some small degree of the threshold estimation challenges. Vast majority of the problem remains yet. For example, such threshold heavily depends on the controlled environments (video streams), which are the case in sites such as airport, mall, bank, play ground, subway, concert, school, hospital, parking place, town center, political event, etc. If the environment changes, then the

Table 3.4: *Comparison of different approaches: symbols* ⊕ *and* ⊖ *denote* Yes *and* No *, respectively.*

Different Internal Issues	Different Approaches					
	Cov. 3.2	*NCRIMA* 3.3	*Mahal.* 3.4	*Bhatta.* 3.5	*E.Entropy* 3.6	*S.Entropy* 3.7
Region of interest used?	⊕	⊕	⊕	⊕	⊕	⊕
Serious effect on NaN?	⊕	⊕	⊕	⊕	⊖	⊖
Outlier effect concerned?	⊖	⊖	⊕	⊕	⊖	⊖
Handling occlusion scenarios?	⊖	⊖	⊖	⊖	⊖	⊖
Singularity problem concerned?	⊕	⊕	⊕	⊖	⊖	⊖
Static threshold concerned?	⊕	⊕	⊕	⊕	⊕	⊕
Classification explicitly used?	⊖	⊖	⊕	⊕	⊖	⊖
Matrix explicitly used?	⊕	⊕	⊕	⊕	⊖	⊖
Matrix inversion involved?	⊕	⊕	⊕	⊕	⊖	⊖
Eigenvalues explicitly reckoned?	⊕	⊕	⊖	⊖	⊖	⊖
Eigenvectors explicitly reckoned?	⊖	⊖	⊖	⊖	⊖	⊖
Direction explicitly used?	⊕	⊕	⊕	⊕	⊕	⊕
Velocity/Displacement directly used?	⊕	⊖	⊕	⊖	⊕	⊕
Mean/Variance explicitly used?	⊕	⊖	⊕	⊕	⊕	⊕
Circular variance considered?	⊖	⊖	⊖	⊖	⊖	⊕
Segmentation explicitly used?	⊖	⊕	⊖	⊕	⊖	⊖
Shadow/lighting effect minimized?	⊖	⊕	⊖	⊕	⊖	⊖
Histogram concept considered?	⊖	⊖	⊖	⊖	⊕	⊖
For direction *atan2()* used?	⊕	⊕	⊕	⊕	⊕	⊕
Raw data normalization used?	⊖	⊖	⊕	⊖	⊖	⊖
Vertical coordinate used?	⊖	⊖	⊕	⊖	⊕	⊕
Foreground estimation estimated?	⊖	⊕	⊖	⊕	⊖	⊖
Rank of matrix directly counted?	⊖	⊕	⊖	⊖	⊖	⊖
Interest points clustering concerned?	⊖	⊖	⊖	⊕	⊖	⊖
Metric classification employed?	⊖	⊖	⊕	⊖	⊖	⊖
Final distance innerly normalized?	⊖	⊕	⊖	⊖	⊕	⊕
Color silhouetted region considered?	⊖	⊕	⊖	⊖	⊖	⊖
Frame-based abnormality concerned?	⊕	⊕	⊕	⊕	⊕	⊕

threshold should be regenerated. Along with probabilistic model, a dynamically estimated threshold, which would be estimated by detecting significant sudden change of the estimated distance values over frame, would perform a bit more. Future directions would include this improvement.

3.8.2.3 Handling of occlusion

The phenomenon of occlusion is very frequent in crowd videos due to high density of distracting targets in the scene. Sometimes physical features of the scene and camera motion may cause occlusion, resulting in the loss of visibility of the tracked target. Optical flow can be reliably estimated between areas visible in two images, but not in occlusion areas, which disappear in the other images. The basic idea of optical flow computation is maintaining the brightness constancy assumption, which relates the image gradient. Occlusion handling is a major problem in optical flow techniques. There are two kinds of occlusion may happen in optical flow estimation, in general. One is from motion occlusion, e.g., occlusion generates due to object motion and the occluded areas from two frames which are not overlapped at the same location. The second one is from mismatching e.g., the occluded regions from different images are overlapped at the same position. The mismatching may happen under different conditions, such as object appearing/disappearing, shadow, color change, or large object deformation (shrinking or expanding), etc. However, occlusion handling is difficult as occluded pixels violate a major assumption of optical flow that each pixel goes somewhere. In theory, the pixels at the occlusion area should not be assigned any flow vector since there is no correspondence available in the other frame. Since our proposed approaches are based on optical flow techniques, no flow vector can be obtained from occlusion areas and as a result the problem of occlusion still remains.

3.8.2.4 Few pixels on targets

Harris corner detector estimates image feature *points of interest* and optical flow techniques track those points over frames. However, the computed features such as interest points may become noisy and unreliable. To overcome this shortcoming, we have contended that in a scene of a high density crowd, detection of individual objects cannot be necessary, and consequently, modeling the crowds at a global level is more practical. The proposed approaches expect a region of interest, which improves the quality of the results and reduces processing time. To get comprehensible information of optical flow from each frame, the region of interest of each frame is separated into small blocks. This helps to minimize the few pixels on target problem a bit more.

3.8.2.5 Effect of Lights and Shadows

In outdoor installations illumination varies significantly such as extremely strong lighting condition (e.g., sunlight) which causes shadows of respective objects. This shadow, causes sturdy noise, serves as an inherent structure of the object which is almost impossible to isolate in reality and this effect does not concern in our approaches. On the other hand, in many indoor installations there may not be significant illumination variation but sometimes some light reflection can appear in the scene. Some of our proposed algorithms (e.g., *NCRIMA* 3.3 and *Bhatta.* 3.5) minimize this effect.

3.8.3 Comparison with some State-of-the-art and Proposed Approaches

Our proposed algorithms have several important differences from the most related and recent and also frequently cited body of works, e.g., Andrade et al. [7, 8], Ali et al. [5], Mehran et al. [131], etc. A brief overview of some important issues of these research works along with one of our approaches (Shannon Entropy 3.7) have been listed on the Table 3.5.

Most of these methods require a learning period to estimate various parameters of the system, and hence reliable learning of unknown parameters is not always accurately possible which could potentially increase the rate of false alarms. For instance, Mehran et al. [131] have introduced a method to detect and localize abnormal behaviors in crowd videos using social force model. They have presented that their estimated social force model is capable of detecting the governing dynamics of the abnormal behavior, even in the scenes that it is not trained. But the false positive detections in their model are result of incorrect estimation of social forces. This is a severe shortcoming of their approach. In a contrary manner, there is no explicit learning period in our approaches, consequently, the false alarm rates are significantly low as compared to [131] (e.g., see Table 3.3).

For crowd segmentation, the method of Ali et al. [5] has been taken into account the goal-directed nature of human crowds, where the members of the crowds have clear knowledge of what and where their goals rest, e.g., extremely large number of people at sporting events, religious festivals, etc. This goal-directed nature has been implemented on to the crowd segmentation framework, where segments are distinguished from each other on the basis of the fate of the particles belong to that segment. The particles with similar fate have similar goals, and, thus, characterize a distinct group of the crowd in a given scene. Results showed some satisfactory results for extremely high crowded scenes but high or medium or low crowded scenes such segmentation would be gone in vain. However, in such case, approaches like Mehran et al. [131], Andrade et al. [7, 8], as well as our framework would show good performance.

Like our approaches, in the work of Andrade et al. [7, 8] crowd behavior has been characterized at a global level by using the optical flow of the video sequence. Unlike our approaches, during the learning stage, a reduced order representation of the optical flow was generated by performing PCA on the flow vectors. Afterwards, top few eigenvectors were used as the representative features and spectral clustering was performed to identify the number of distinct motion patterns present in the video. The features in the clustered motion segments were used to train different HMMs which were then used for event detection in crowds. The method was only tested by data obtained from simulation. A general limitation of simulation is that models are typically unstructured and must be developed for problems that are also unstructured. It is often impractical to realistically validate simulation results all of the above. Besides, model building for simulation is often costly and time-consuming.

Table 3.5: *Comparison of our best method with some state-of-the-art approaches in the direction of abnormality detection from crowded scenes: symbols* ⊕, ⊖, *and* ○ *denote* Yes, No, *and* Unknown, *respectively.*

Different Issues	Different Approaches			
	Andrade et al. [7, 8]	Ali et al. [5]	Mehran et al. [131]	Our approach (e.g., 3.7)
Detected anomaly?	⊕	⊕	⊕	⊕
Training data used?	⊕	⊕	⊕	⊖
Learning process used?	⊕	⊕	⊕	⊖
Region of interest used?	⊖	⊖	⊖	⊕
Low rate of false alarm?	○	○	⊖	⊕
Fixed camera employed?	○	⊕	⊕	⊕
Tested with real data-set?	⊖	⊕	⊕	⊕
Tested with simulated data?	⊕	⊖	⊖	⊕
Multi-camera data-set used?	○	⊖	⊖	⊖
Localized where anomaly was?	⊖	⊕	⊕	⊖
Occlusion handling concerned?	⊖	⊖	⊖	⊖
Automatic threshold concerned?	⊖	⊖	⊖	⊖
Single target tracking concerned?	⊖	⊖	⊖	⊖
Usually affected with number of movers?	⊖	⊖	⊖	⊖
Fitted for extremely high crowded scenes?	○	⊕	○	○
Fitted for low/medium/high crowded scenes?	⊕	○	⊕	⊕

Chapter 4

Detection of Usual Video Events

Contents

4.1 Overview

Event detection in video surveillance is an important task for the places of both private and public. As huge amount of video surveillance data makes it an exhausting work for people to keep watching and finding anomalous events, an automatic surveillance system is strongly needed for detecting suspicious events. A video event is defined to be an observable action or change of state in a video stream that would be important for the security management.

In this chapter we (in [SD09b]) have presented a system that generates automatically *pseudo Euclidian distance* (PED) from the trigonometrically treatment of *motion history blob* (MHB) obtained from *motion history images* (MHI) to extract efficient image features, which are pertinent to *video event detection* (VED). Given a point with its direction of motion where the point coincides the center of a circle. *How far the point can virtually travel inside the circle with that direction?* That virtual distance is called *pseudo Euclidian distance*. PED, would be potentially used in wide variety of computer vision applications, remains a great contribution of the Thesis. To show the interest of the usage of PED, we have proposed a PED based methodology for VED and the detection results of some events at *TRECVID2008*[43] in real videos have been demonstrated. Since the surveillance video for events detection is captured from an airport, it is unconstrained and has the characteristics, e.g., highly clutter, massive population flow, heavy occlusion, reflection, shadow, fluctuation, varying target sizes, sometimes low video quality, etc. In spite of these difficulties, we have striven to extract efficient image features that are pertinent to events of interest. As different individuals may have dramatically different appearances, the most relevant image features of events are motion patterns (e.g., direction and position of the motion history blob, etc.). Moreover, all the videos are taken from surveillance cameras which means the position of the cameras is fixed and cannot be changed. As a result, all the motion information extracted from the surveillance videos can be caused only by the activation of people in the videos. No motion implies no activity on the video frames.

4.2 Related Works

Events may vary greatly in duration, from two frames to longer duration events that can exceed the bounds of the excerpt. In crowded environments e.g., airports, malls, etc., objects merge and occlude each other very frequently, as a result conventional background subtraction methods do not work as appointively. Many single frame detection algorithms based on transfer cascades [170, 114] or recognition [41, 157, 20] have demonstrated some high degree of promise for pedestrian detection in real world busy scenes with occlusion. To detect pedestrian histogram of gradients was used in [41], while authors in [157, 20] used biological inspired model for recognizing different classes including pedestrian. However, most of those pedestrian detection algorithms are significantly slow for real time applications. For example, authors in [184] noted that the state-of-the-art algorithms for pedestrian detection e.g., [41] takes around 0.5 seconds for recognition of 128×64 size image frame, [157] takes 2 seconds/frame, and [20] takes about 80 seconds/frame. A target detection and tracking algorithm based on the measurements of a stereo audio and cycloptic vision sensor has been presented in [192]. To detect events in the *TRECVID'08* many algorithms have been proposed, e.g., based on: change detection [188], analysis of trajectory [109], trajectory and domain knowledge [55], spatio-temporal video cubes [53], Haar based pedestrian detection and histogram matching [184], optical flow concepts [61, 97, 58], etc. In the work of [109] people meeting (PeopleMeet) event was detected mainly by analyzing pedestrian trajectories. They detected and tracked people in the scene by using the method described in [81]. To get reliable pedestrian trajectories for people meeting event detection job, they suggested a detection-based hierarchical association method which was capable of robustly tracking multiple pedestrians under such challenging conditions. Their method produced pedestrian trajectories by the help of progressively associating detection responses given by the pedestrian detector as introduced by [176]. A combination of trajectory and domain knowledge based subsystems can be found in [55]. The trajectory-based subsystem implements human detection and tracking to generate trajectory and three-level trajectory features are used to detect PersonRuns, PeopleMeet, PeopleSpiltUp, and Embrace. The domain knowledge-based subsystem constructs specific models for PeopleMeet, Opposingflow, and ElevatorNoEntry depending on domain knowledge. Nevertheless, vast diversity of one event viewed from different view angles, different scales, different degrees of partial occlusion, etc., make challenge for performance of the event detectors; hence it is necessary to greatly improve their effectiveness by further investigation.

We have proposed a methodology based on *pseudo Euclidian distance*, PED, for *video event detection* (VED). To extract image features, optical flow estimation would be a superior grade for the crowd scene but it is too sensitive to the small noise because of the broadness of the camera view. If there are many people in the videos, the existence of small motion noise will be extremely negative and unreliable. Therefore, estimating the movement of objects by optical flow is difficult. Deeming this fact we rest with confidence on motion history images (MHI), *motion history blob* (MHB), and trigonometric treatments of MHB which generate PED to extract efficient image features which are pertinent to event of interests. The MHI is a representation of the history of pixel-wise changes, yet remains a computationally inexpensive method for analysis of object motions and effectively only previous frame needs to be stored. We segment the MHI to grasp the essential sequence of motion components, *object of interest* (OoI) or MHB, which are then tracked by using PED. Generation and usage of PED are the unique contribution of our current investigation. There are several state-of-the-art algorithms for tracking OoI, e.g., particle filtering [11], hybrid strategy [31], etc. Since occlusions happen frequently in limited camera scope, particle filtering may archive a commendable performance. But particle filtering is a time-consuming process, especially when the object tracked is large. It is difficult to complete the test on evaluation data within the limited time. Henceforth, we have taken up PED for MHB tracking with the target for different kinds of VED.

4.3 Calculation of Pseudo Euclidian Distance (PED)

4.3.1 Extraction of Motion History Blobs (MHB)

The strength of the motion history images (MHI) is that although it is a representation of the history of pixel-wise changes, only previous frame needs to be stored. It is easy to implement and adds little computational cost to the real-time system. In a motion history image, $H_\tau(x,y,t)$, pixel intensity is a function of the temporal history of position or motion at that point. The previous method of MHI as described in [46] was based on frames rather than time. Currently, a simple replacement and duration operator based on time-stamping is used [22]:

$$H_\tau(x,y,t) = \begin{cases} \tau & if\ \psi(x,y,t) = 1 \\ max(0, H_\tau(x,y,t-1) - \delta) & otherwise \end{cases} \tag{4.1}$$

where x, y, and t demonstrate the position and time; τ is the current time-stamp; δ is the maximum time duration constant (e.g., few seconds) associated with the template; $\psi(x,y,t) = 1$ signals object presence or motion in the current video image. The $\psi(x,y,t)$ can be computed from background subtraction, frame differencing, optical flow, edges, stereo-depth silhouettes, flesh-colored regions, etc.

The use of time-stamps allows for a more consistent port of the system between platforms where speeds may differ. System time is consistent during processing where frame rate is not. Thus time is explicitly encoded in the motion template. The Eq. 4.1 indicates that the MHI pixels where motion occurs are set to the current time-tamp τ, while the pixels where motion happened far ago are cleared. The above update function is called each time a new image is received and the corresponding silhouette image is formed. The result of the function is a scalar-valued image where more recently moving pixels are brighter; only we wish to deal with those brighter part (region of motion component) which we called *motion history blob* (MHB) or *silhouetted region of motion component* (SRMC). We get absolute difference between two frames and threshold it and using the thresholded frame update function of Eq. 4.1 to get MHB:

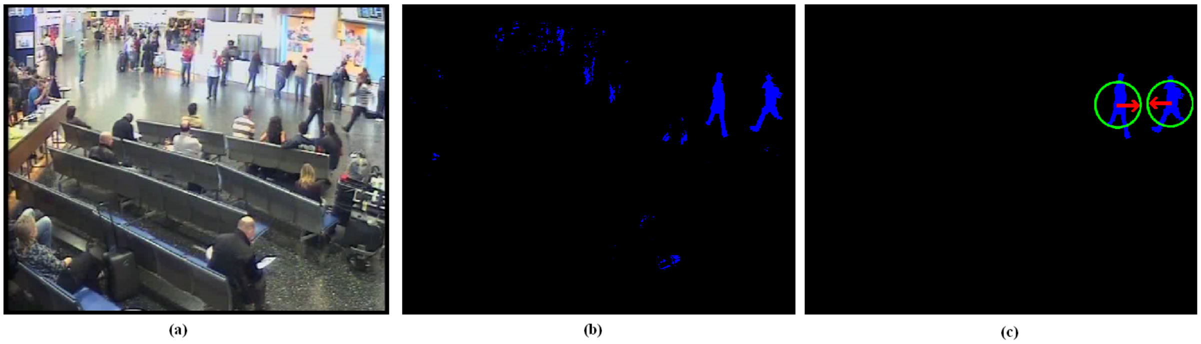


Figure 4.1: (a): camera view; (b): blue regions are the current silhouettes (motions mask) or *motion history blobs* (MHBs); (c): view after suppression of the little MHBs from (b), and red arrows point towards global motion orientations of the rest motion components.

$$\psi(x,y,t) = \begin{cases} 1 & if\ D(x,y,t) \geq \eta \\ 0 & else \end{cases} \tag{4.2}$$

where η be a threshold and signal $D(x,y,t)$ with difference distance δ can be estimated by dint of:

$$D(x,y,t) = |I(x,y,t) - I(x,y,t \pm \delta)| \tag{4.3}$$

where $I(x,y,t)$ is the intensity value of pixel location with coordinate (x,y) at the t^{th} frame of the video. Since the motion history image encodes in a single image the temporal nature for the motions over some time interval, the motion segmentation should be easier than in methods those attempt to segment and propagate motion between frames. Fig. 4.1 (a) and (b) depict a snapshot of original image and the current silhouette motion or MHB respectively. To get sequence of motion components, it is important to segment motion regions which were produced by the movement of parts or the whole of the object of interest. On getting sequence of silhouetted motion component (MHB or SRMC), we calculate number of points within each component so that it will be easy to check out the case of little motion, which will be neglected under an experienced threshold T_h (say 50-point). On filtering, we estimate the center of each remaining motion component (motion history blob) by applying Hu's moments [80]. A circle with fix radius is drawn coincident with each center of motion history blob as marked by a green colored circle in Fig. 4.1 (c). The centroid and global motion orientation Φ of each motion component (marked red arrow in Fig.4.1 (c)) have been estimated in the following subsections. Finally, each region of motion component or motion history blob has an explicit center (e.g., $P(x_0, y_0)$ in Fig.4.2) and a global motion of orientation Φ. After the trigonometrical treatments of the circle of the each motion history blob, the position and angle information ($P(x_0, y_0)$, Φ) will be used to generate *pseudo Euclidian distances* (PED). In our current investigation PED will be measured in terms of pixels length which will play the vital role for tracking the region of motion components (MHB) in video frames.

4.3.2 Estimation of the Centroid of Motion History Blob

Centroid is the term given to the center of a region, area, etc. Moments give us an indication of the center, spread, skewness etc. of what we are measuring (e.g., pixel values and locations). Image moments are useful to describe objects after segmentation. As we are dealing with image regions, we need to work with two dimensional moments. The use of moments for image analysis and object representation was introduced in 1962 by Hu [80]. Moments can give us a highly compressed indication of the shape of the object being measured in our image.

The two-dimensional moment m_{pq} of order $p+q$ of a density distribution function $\rho(x,y)$ (e.g., image intensity) are defined in terms of Riemann integrals as:

$$m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q \rho(x,y) dxdy. \tag{4.4}$$

The two-dimensional moment for a discretized image $\rho(x,y)$ is given by:

$$m_{pq} = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} x^p y^q \rho(x,y). \tag{4.5}$$

A complete moment set of order n consists of all moments m_{pq}, such that $p+q \leq n$. Hu's uniqueness theorem states that if $\rho(x,y)$ is piecewise continuous and has nonzero values only in a finite region, then the moments of all orders exist. Moreover, the theorem proves that the moment set $\{m_{pq}\}$ is uniquely determined by $\rho(x,y)$ and oppositely, $\rho(x,y)$ is uniquely determined by $\{m_{pq}\}$. Since an image segment (e.g., MHB) has a finite area and, in the worst case, is piece-wise continuous, moments of all orders exist, and a moment set can be computed that will uniquely describe the information contained in the image segment. To characterize all of the information contained in an image segment requires a potentially infinite number of moments. The challenge is to select a meaningful subset of moments that contain sufficient information to accurately characterize the image. The definition of the zero-th order moment m_{00} of the image $\rho(x,y)$ is

$$m_{00} = \sum_{-\infty}^{\infty}\sum_{-\infty}^{\infty}\rho(x,y). \tag{4.6}$$

This moment represents the total mass of the given image. When computed for a silhouette image, the zero-th moment represents the total object area. The two first order moments $\{m_{10}, m_{01}\}$ are used to locate the *center of mass* (centroid) of the object. The centroid defines a unique location with respect to the object that may be used a reference point to describe the position of the object within the field of view. The coordinates of the centroid can be defined through moments as shown:

$$\widehat{x} = \frac{m_{10}}{m_{00}} \tag{4.7}$$

$$\widehat{y} = \frac{m_{01}}{m_{00}}. \tag{4.8}$$

Using Eq. 4.7 and 4.8, the central moments μ_{pq} are defined as:

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(x-\widehat{x})^p(y-\widehat{y})^q\rho(x,y)d(x-\widehat{x})d(y-\widehat{y}). \tag{4.9}$$

Using Eq. 4.5 yields:

$$\mu_{pq} = \sum_{-\infty}^{\infty}\sum_{-\infty}^{\infty}(x-\widehat{x})^p(y-\widehat{y})^q\rho(x,y). \tag{4.10}$$

It is well-known that under the translation of coordinates, the central moments do not change, and are therefore invariants under translation. It is quite easy to express the central moments μ_{pq} in terms of the ordinary moments m_{pq}. For the first four orders, we have:

$$\mu_{00} = m_{00} \tag{4.11}$$

$$\mu_{10} = 0 \tag{4.12}$$

$$\mu_{01} = 0 \tag{4.13}$$

$$\mu_{20} = m_{20} - m_{00}\widehat{x^2} \tag{4.14}$$

$$\mu_{11} = m_{11} - m_{00}\widehat{xy} \tag{4.15}$$

$$\mu_{02} = m_{02} - m_{00}\widehat{y}^2 \tag{4.16}$$

$$\mu_{30} = m_{30} - 3m_{20}\widehat{x} + 2m_{00}\widehat{x}^3 \tag{4.17}$$

$$\mu_{21} = m_{21} - 3m_{20}\widehat{y} - 2m_{11}\widehat{x} + 2m_{00}\widehat{x}^2\widehat{y} \tag{4.18}$$

$$\mu_{12} = m_{12} - m_{02}\widehat{y} - 2m_{11}\widehat{y} + 2m_{00}\widehat{xy}^2 \tag{4.19}$$

$$\mu_{03} = m_{03} - 3m_{02}\widehat{y} + 2m_{00}\widehat{y}^3 \tag{4.20}$$

To achieve invariance with respect to orientation and scale, we normalize central moments as follows:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{[(p+q)/2]+1}} \tag{4.21}$$

where $(p+q) \geq 2$. For arbitrary shapes, a potentially infinite number of moments $\{\eta_{pq}\}$ can uniquely describe that shape. These image moments are invariant with respect to translation and scale operations, but an infinite number of moments are obviously impractical, and in any case, we want a compressed representation of shape. Since objects in video images contain a large amount of noise, only moments up to the third order are generally practicable. There are ten moments up to the third order, but scale normalization and translation invariance fix three of these moments at constant values. Rotation invariance takes away one more degree of freedom leaving us with six independent dimensions. Hu uses seven variables however: six to span the six degrees of freedom, and a final seventh variable who's sign removes reflection invariance. Only the first six of the Hu variables below give reflection invariance. The seven moment-based features proposed by Hu that are functions of normalized moments up to the third order are:

$$\psi_1 = \eta_{20} + \eta_{02} \tag{4.22}$$

$$\psi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{4.23}$$

$$\psi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{4.24}$$

$$\psi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{4.25}$$

$$\psi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})^2[(\eta_{03} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} + \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \tag{4.26}$$

$$\psi_6 = (\eta_{20} + \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \tag{4.27}$$

$$\psi_7 = (3\eta_{21} + \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \tag{4.28}$$

Equations ψ_1 and ψ_2 provide scale and translation independence, ψ_3 to ψ_6 ensure rotation with reflection invariance, and ψ_7 provides reflection discrimination in it's sign.

4.3.3 Global Motion Orientation Φ Estimation

We calculate global motion orientation Φ of each remaining component or motion history blob as used in [45]:

$$\Phi = 2\pi - \Phi_{ref} - \frac{\sum_{x,y} angDiff(\Phi_{con}(x,y), \Phi_{ref}) \times N(\tau, \delta, H_\tau(x,y,t))}{\sum_{x,y} N(\tau, \delta, H_\tau(x,y,t))} \tag{4.29}$$

where 2π accommodates the adjustment for images with top-left origin; Φ_{ref} be the base reference angle (peaked value in the histogram of orientations); $N(\tau, \delta, H_\tau(x,y,t))$ be a normalized motion history image value (linearly normalizing the motion history image from 0-1 using the current time-stamp τ and duration δ); $angDiff(\Phi_{con}(x,y), \Phi_{ref})$ be the minimum signed angular difference of an orientation from reference angle; and $\Phi_{con}(x,y)$ be the motion orientation map found from gradient convolutions. For the convolution, the Sobel gradient masks can be used as:

$$F_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{4.30}$$

$$F_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}. \tag{4.31}$$

With the gradient images $F_x(x,y)$ and $F_y(x,y)$ calculated from the motion history image, it is a easy matter to obtain the gradient (motion) orientation for a pixel by:

$$\Phi_{con}(x,y) = tan^{-1}\frac{F_y(x,y)}{F_x(x,y)}. \tag{4.32}$$

Care should be taken, when calculating the gradient information as it is only valid at particular locations within the MHI. The surrounding boundary of the MHI layering should not be used because non-silhouette (zero valued) pixels would be included in the gradient calculation, thus corrupting the result. Only MHI interior silhouette pixels should be examined. Additionally, we should not use gradients of MHI pixels that have a contrast which is too low (inside a silhouette) or too high (large temporal disparity) in their local neighborhood [45].

4.3.4 Pseudo Euclidian Distance

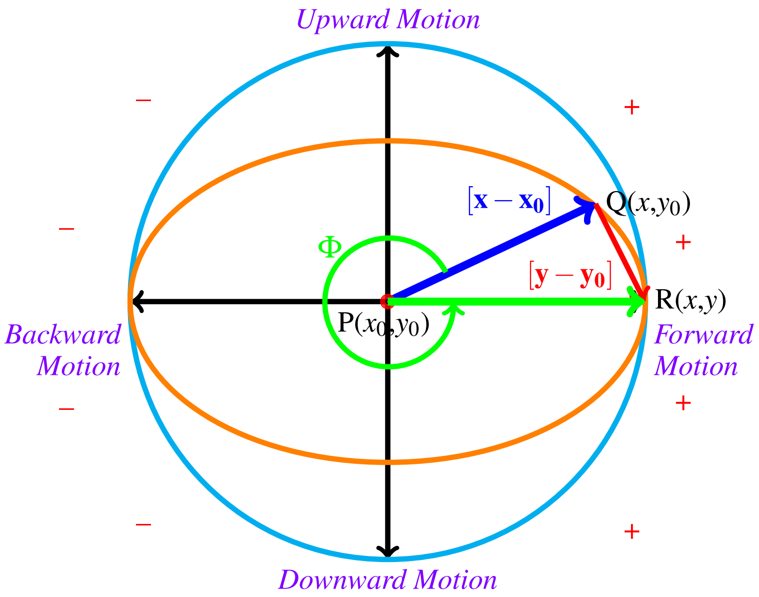


Figure 4.2: Global motion orientation Φ of a *motion history blob* inside of an ellipse. Circle is an exceptional ellipse in which the two foci are coincident at the center of the ellipse.

We take into account four motion directions, based on its previous motion directions, namely forward (+), backward (-), upward (around $+\pi/2$), and downward (around $-\pi/2$) as shown in Fig.4.2, which illustrates a situation where the center of a motion history blob $P(x_0,y_0)$ (center of circle/ellipse) moves in forward direction with respect to its previous motion direction, e.g., from point $P(x_0,y_0)$ tends to $R(x,y)$, i.e., $\overrightarrow{PR} = \overrightarrow{PQ} + \overrightarrow{QR}$, and having this fact it is evident that:

$$\Phi = tan^{-1}\frac{y-y_0}{x-x_0} \quad \Rightarrow y = y_0 + (x-x_0)tan\Phi. \tag{4.33}$$

Although we are dealing with only circle, the global motion orientation Φ has been shown inside of an ellipse for better presentation as a circle is a special case of an ellipse. An ellipse is defined as the locus of points equidistant to two fixed points called the foci. Deeming that the ellipse is centered at (0,0) and foci are located at $(\pm c, 0)$ then the standard equation of it can be formulated by dint of:

$$\frac{x^2}{a^2} + \frac{y^2}{a^2-c^2} = 1 \tag{4.34}$$

where a and $\sqrt{a^2-c^2}$ are the semi-major and semi-minor axes respectively. The foci always lie on the semi-major axis, spaced equally each side of the center of the ellipse. If the lengths of semi-major axis a and semi-minor axis $\sqrt{a^2-c^2}$ are identical, i.e., $c=0$, then both foci are coincident at the center of ellipse, explicitly, the ellipse Eq. 4.34 comes into existence an equation of a circle, $x^2+y^2=a^2$, where a is renamed as the radius of the circle. The area enclosed by the circle is π multiplied by the radius squared a^2. Since we are considering unit area, i.e., $\pi a^2=1$, the Eq. 4.34 can be rewritten as: $x^2+y^2=\frac{1}{\pi}$. Substituting y from Eq. 4.33 into this new circle equation, the following quadratic equation yields:

$$(1+tan^2\Phi)x^2+2tan\Phi(y_0-x_0tan\Phi)x+(y_0-x_0tan\Phi)^2-\frac{1}{\pi}=0. \tag{4.35}$$

On solving Eq. 4.35, we get two solutions or roots, say x^+ and x^-, formulated as:

$$x^+=\frac{-tan\Phi(y_0-x_0tan\Phi)+\sqrt{(tan\Phi(y_0-x_0tan\Phi))^2-(1+tan^2\Phi)((y_0-x_0tan\Phi)^2-\frac{1}{\pi})}}{1+tan^2\Phi} \tag{4.36}$$

$$x^-=\frac{-tan\Phi(y_0-x_0tan\Phi)-\sqrt{(tan\Phi(y_0-x_0tan\Phi))^2-(1+tan^2\Phi)((y_0-x_0tan\Phi)^2-\frac{1}{\pi})}}{1+tan^2\Phi} \tag{4.37}$$

and their corresponding y components y^+ and y^- are as follows:

$$y^+=y_0+(x^+-x_0)tan\Phi \tag{4.38}$$

$$y^-=y_0+(x^--x_0)tan\Phi. \tag{4.39}$$

There are three variables namely x_0, y_0, Φ in Eq. $4.36-4.39$ of which Φ can be easily calculated using Eq. 4.29. However, x_0, y_0 are merely the positions of the moving component which can be obtained from their x and y coordinates respectively. Since we are using unit scale, there will be a severe error if we want to use their corresponding coordinates directly. Normalization can uniquely solve that problem. Assuming that the frame size is $f_x\times f_y$ (e.g., 640×480, etc.) pixels, then to obtain workable values of x_0 and y_0 normalization can be performed as:

$$N_{xy}=\frac{\frac{x}{f_x}+\frac{y}{f_y}}{2} \tag{4.40}$$

$$x_0=\frac{2N_{xy}-1}{\sqrt{\pi}} \tag{4.41}$$

$$y_0 = \sqrt{\frac{1}{\pi} - x_0^2}\,(1 - 2N_{xy}) \tag{4.42}$$

where N_{xy} be a *pseudo number*, between 0 and 1, generated by using the x and y coordinates of any point on the frame. Take into consideration the position (x_0, y_0) and those two points (x^+, y^+) and (x^-, y^-), it is easy to compute their respective *pseudo Euclidean distance* (PED), signified λ^+ and λ^-, by dint of:

$$\lambda^+ = \sqrt{(x_0 - x^+)^2 + (y_0 - y^+)^2} \tag{4.43}$$

$$\lambda^- = \sqrt{(x_0 - x^-)^2 + (y_0 - y^-)^2}. \tag{4.44}$$

Let us give a simple example of PED calculation. Fig. 4.3 depicts the PED calculation where in video frames the center of motion history blob of a person has been moved from position (150,100) to (640,100) with global motion direction diversifications roundabout 15° and an invariant velocity of 10 pixels per frame.

Once we get PED, many routines can be employed to use the raw information for analysis or recognition. For instance, it is possible to detect video events using PED on employing some routines, as stated in the following subsections. We are confident that the future detailed investigation results of PED would work somewhat in parallel to the assumption and prediction algorithms e.g., local and/or global optical flow techniques, Kalman filter, particle filters, etc.

4.4 Video Events Detection (VED)

We wish to use PED for VED. For this aim, we need the explicit information of motion history blobs that can be gained by tracking objects of interest and thereof can get more information which will be used for specific VED, e.g., PersonRuns, ObjectPut, OpposingFlow, PeopleMeet, Embrace, PeopleSplitUp.

4.4.1 Motion History Blobs (MHB) Tracking

Assuming that the radius a is unit and consists of S number of pixels. PED in Eq. 4.44 can be expressed in term of pixels length λ^+_{pixel} & λ^-_{pixel} respectively:

$$\lambda^+_{pixel} = Number\ of\ pixels\ to\ pass\ on\ \lambda^+ = \lceil S * \lambda^+ \rceil \tag{4.45}$$

$$\lambda^-_{pixel} = Number\ of\ pixels\ to\ pass\ on\ \lambda^- = \lceil S * \lambda^- \rceil \tag{4.46}$$

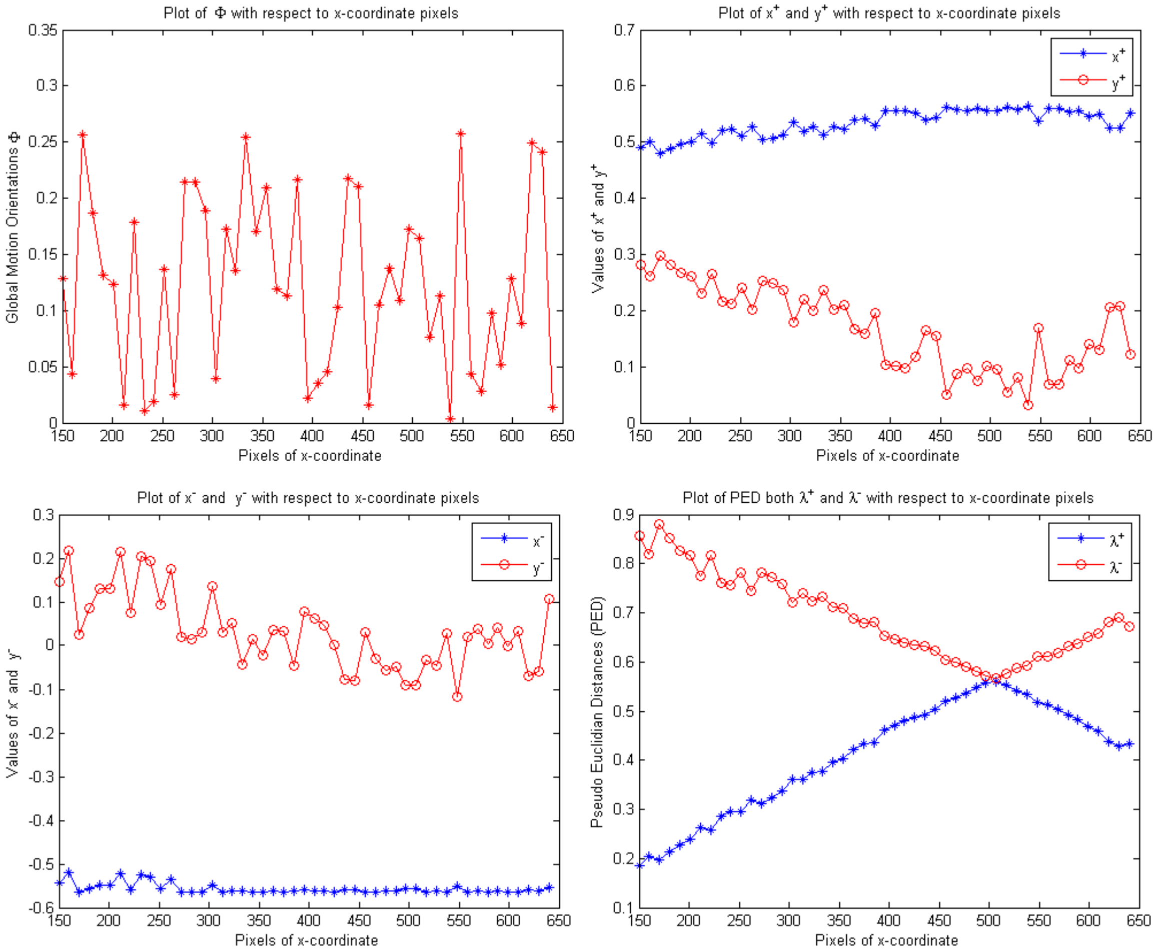


Figure 4.3: A simple example of PED calculation concerning the movement of the center of circle of the Motion History Blob of a person from (150,100) to (640,100) with global motion orientation variations about 15° and a constant velocity of 10 pixels per frame. λ^- exhibits concave up or convex cup; λ^+ substantiates concave down or convex cap.

which use as the judgement index for tracking MHB in the following algorithm.

Algorithm [*M: total number of circles in any frame f, N: total number of circles in frame $f+1$, m: circle counter in frame f, n: circle counter in frame $f+1$*]

```
1. begin
2. if $N = 0$ then exit
3. initialization: $m = 1$, $n = 1$
4. if $m \leq M$
4.1 then
4.1.1 if $n \leq N$

    (i) Calculate Euclidean distance $d(C_f^m, C_{f+1}^n)$ between two centers of circle $C_f^m$ & $C_{f+1}^n$ in $f$ & $f+1$ and store it
    (ii) Increase n by 1
    (iii) Repeat step 4.1.1

4.2 else
4.2.1 Select the minimum distance $d_{min}$, caused by two centers ${}_{min}C_f^m$ & ${}_{min}C_{f+1}^n$ with angles $\Phi_f$ & $\Phi_{f+1}$ respectively, and estimate its normalized pixel value $T_{pixel}$
```

$$d_{min} = d({}_{min}C_f^m, {}_{min}C_{f+1}^n) = arg \min_{k=1 \ldots N} \left[d(C_f^m, C_{f+1}^n) \right]_k \tag{4.47}$$

$$T_{pixel} = \left[S * \frac{1}{2} \left[1 + \frac{2}{\sqrt{\pi}} \sum_{k=0}^{\infty} \frac{(-1)^k \{d_{min}\}^{2k+1}}{k!(2k+1)} \right] \right] \tag{4.48}$$

```
4.2.2 Select $\lambda^+_{pixel_f}$ or $\lambda^-_{pixel_f}$ with respect to previous direction of movement, if same then use $\lambda^+_{pixel_f}$, otherwise use $\lambda^-_{pixel_f}$, and save current motion direction
4.2.3 The area of circle with radius $(\lambda^+_{pixel_f} + T_{pixel})$ (leading edge of the convex cap) or $(\lambda^+_{pixel_{f+1}} + T_{pixel})$ (falling edge of the convex cap) will be greater than or equal to than that of caused by $(\lambda^+_{pixel_{f+1}} - T_{pixel})$ or $(\lambda^+_{pixel_f} - T_{pixel})$, explicitly:
```

$$\pi(\lambda^+_{pixel_f} + T_{pixel})^2 \geq \pi(\lambda^+_{pixel_{f+1}} - T_{pixel})^2 \ or \ \pi(\lambda^+_{pixel_{f+1}} + T_{pixel})^2 \geq \pi(\lambda^+_{pixel_f} - T_{pixel})^2.$$

$$\textbf{\textit{If}} \textit{ there exists}: \qquad T_{pixel} \geq \left[\left| \frac{\lambda^+_{pixel_{f+1}} - \lambda^+_{pixel_f}}{2} \right| \right] \ \& \ T_{pixel} \geq 2 \tag{4.49}$$

4.2.3.1 ***then*** [in 4.2.3 the $\lambda^{-}_{pixel_f}$ has not been considered for simplicity]

(i) *A new motion of the motion history blob has been detected*
(ii) *Assign* $_{min}C^{m}_{f}$ *completely convergence to* $_{min}C^{n}_{f+1}$
(iii) *If there exists occlusion* $(T_{pixel} < 3)$ & $(\lambda^{+}_{pixel_f} = \lambda^{+}_{pixel_{f+1}})$ *then choose reasonable range of same orientation for each motion history blob and after occlusion compare its new orientation with the previous orientations.*

4.2.3.2 ***else***

The motion history blob has insignificant motion or out of frame

4.2.4 *Disregard center of circles* $_{min}C^{m}_{f}$ *and* $_{min}C^{n}_{f+1}$
4.2.5 *Decrease both M and N by* 1
4.2.6 *Increase m by* 1 *and set* $n = 1$
4.2.7 *Repeat step 4*
5. ***end***

When those explicit information of motion history blobs are available, the algorithm can easily be made suitable for different kinds of video event detection, e.g., PersonRuns, ObjectPut, OpposingFlow, PeopleMeet, Embrace, PeopleSplitUp.

4.4.2 PersonRuns (P_R)

We set three experienced T values, as thresholds, in Eq.4.49. If we use one T value then one encountered problem is that people near the camera are supposed to generate large motion and people far from the camera cannot generate such motion even when they would make very quick motion (e.g., running). To obtain an acceptable distribution of motion flow pattern, people near or far from the camera should be fairly treated. To solve this problem, we take into account three T values namely T_1 adjacent region to camera (d_1), T_2 middle region (d_2), and T_3 far region from camera (d_3) where $T_1 > T_2 > T_3$. If the distance of the camera observing region is d then the d is divided into three experienced distances d_1, d_2, and d_3 where $d_1 > d_2 > d_3$. If the camera is fixed the division can be accomplished easily. If the direction variation between two circles is about $|\Phi_f - \Phi_{f+1}| \leq \frac{\pi}{4}$ and each time Eq. 4.49 satisfies then the event is judged as P_R.

4.4.3 ObjectPut (O_P)

The O_P event is commonly characterized if there is downward motion over several frames. The down ward motion, which is stored over a period of frames, may pose variable direction between $-\frac{5\pi}{12}$ and $-\frac{7\pi}{12}$ over several frames. The approach does not consider any event as a positively detected which goes different from down ward motion (e.g., throw a bottle in dustbin). Hence it use downward motion, it can recognize the event if someone sitting down as a false positive O_P.

4.4.4 OpposingFlow (O_F)

The algorithm can easily be adapted to detect the person opposing the general flow of the scene, even without a predefined direction of opposing flow. The general direction of the scene can be calculated by considering forward motion or backward motion of the object of interests for some period in some defined region (e.g., door entry/exit). On defining the scene direction, if there exist any forward motion or backward motion with respect to it, then there is an O_F event.

4.4.5 PeopleMeet (P_M)

Assuming that people will away from each other before meeting and keep a minimum distance d_m to them during meeting. Two events may occur either crossing or meeting. The relative distance d_r between persons will be larger than d_m at the beginning of appearance and will decrease towards d_m and go beyond d_m in time and their relative orientation are in reasonable range, with these conditions if one or both persons stop (few or disappear motions) within d_m, then P_M event occurs, otherwise *crossing* (which causes false positives) occurs.

4.4.6 Embrace (E_m)

This event is close to P_M and hence assuming that E_m event happens immediately after P_M. After detecting P_M the meeting region is encompassed by a circle with approximate radius of d_m and within this region calculate d_r again. An E_m is said to be detected where d_r and orientation are below the given thresholds.

4.4.7 PeopleSplitUp (P_S)

Considering that P_S event happens after a while of detecting event P_M when one or more person will separate from a group (out of the circle). Compute and update each crowd center in consecutive frames to detect if a person is decided to leave the corresponding crowd circle. If the relative distance between the person and the crowd center is larger than d_m, then a P_S event is said to be occurred. The vast majority of the false positives are brought forth by frowzy background, occlusions of people, and sophisticated interactions among influential personages.

4.5 Experimental Results

A wide variety in the appearance of the event types makes the events detection task in the surveillance video selected for the *TRECVID2008* extremely difficult. The source data of TRECVID2008 comprise about 100 hours (*10 days * 2 hours per day * 5 cameras*) of video obtained from Gatwich Airport surveillance video data. A number of events for this task were defined. Since all the videos are taken from surveillance cameras which means the position of the cameras is still and cannot be changed. However, it was not practical for us to analyze 100 hours of video except some hours.

Different running cases in the crowd scenes have been depicted on Fig.4.4, 4.5, 4.6, 4.7, 4.8, and 4.9. All of those events have been detected as *true positive PersonRuns* P_R event. Nevertheless, the P_R event detector can not detect accurately event like a little girl is running while her father is walking on the waiting area. It can not also differentiate between a person runs and a wagon runs. These type of events have have been detected as *false positive* which occurs when we are observing a P_R event when in truth there is none. Fig.4.10, 4.11, 4.12, and 4.13 demonstrate some events detected as *false positive* P_R events. On the other hand, it has almost no effect on detection events e.g., a kid or child is running far from the camera, etc. Fig.4.14 shows some *failure or false negative* cases where persons inside red marked rectangles cannot be detected as P_R event. There are several reasons behind such cases:

- Video events have taken place significantly far distance from the camera and hence the considerable amount of motion components were insufficient to analyze over the threshold T_h.
- Sometimes the amount of motion components are enough but threshold is not applicable, e.g., a little girl is running while her father is walking on the waiting area.
- Different type of overlapping events.

The *ObjectPut* O_P detector detected several events as *true positive*. Some O_P events detection from crowd scenes have been shown on Fig.4.15, 4.16, and 4.17. Nevertheless, the O_P detector recognized the event if someone sitting down on the bench as *false positive*, e.g., Fig. 4.18. It can not detect partially overlapping events, e.g., someone is putting bottles or bags which are partially occluded by other object, etc. or non-overlapping small object e.g., somebody is putting small stuff on the shelf, etc. A considerable portion of the false positives appear also from object get. Because it is often very hard to distinguish between object put and object get. Fig. 4.19 and 4.20 demonstrate several examples of *failure or false negative* cases where the object inside red marked rectangles cannot be detected as O_P event.

Usually, people pass through a unidirectional main gate, if somebody comes out opposite of the normal direction, then such event should be detected as *OpposingFlow* O_F event. Some *true positive* detection examples of O_F event have been depicted in Fig. 4.21 and 4.22. Nonetheless, the detector has hardly effect on detection event like Fig.4.23 where a person is coming out in opposite of the normal direction and putting a stop to somewhere on the crowd.

Fig.4.24 shows few *true positive* detection examples of *PeopleMeet* P_M and *Embrace* E_m events. Fig.4.25 also demos few *true positive* detection examples of P_M and *PeopleSplitUp* P_S events. There are failure of P_S detections, for instance, a person suddenly left the meeting place by running (e.g., Fig.4.26).

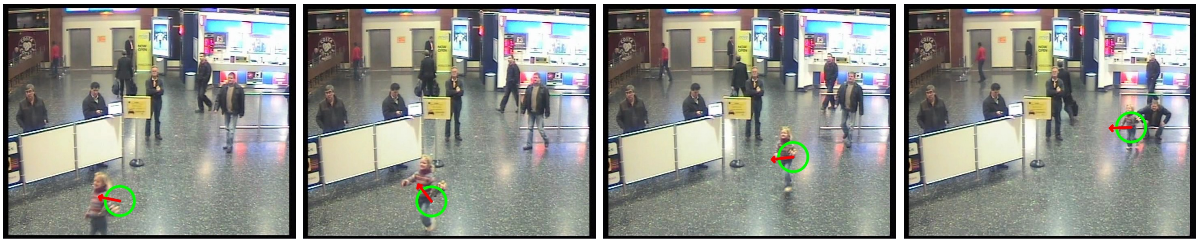


Figure 4.4: A little girl is running on the waiting area which was detected as *true positive* P_R event.

The results obtained from our methodology along with ground truth events of those videos have been demonstrated on the Table 4.1. The *sensitivity* (or *recall*) and *precision* rates of the methodology have been listed. Normally, high recall and precision rates are expected by minimizing the number of false positive and false negative events. A detector is said to be ideal if its sensitivity is equal to 1. Table 4.2 shows some selected good results concerning *PersonRuns* event detection from the TRECVID2008 [43]. Authors in [58] used a total of 314 PersonRuns events of which undetected 170 and successfully

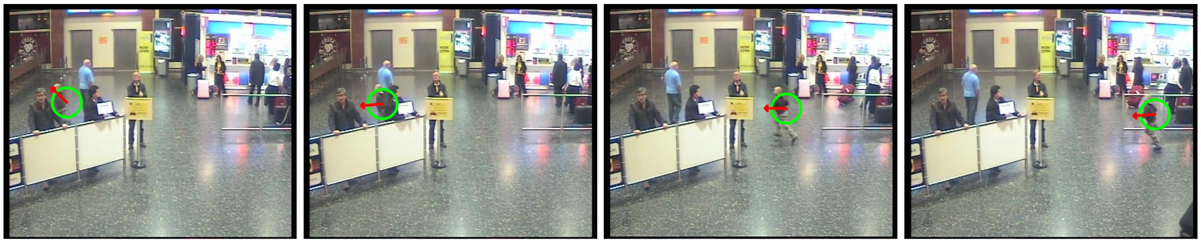


Figure 4.5: A boy is crossing the waiting area by running which was detected as *true positive* P_R event.

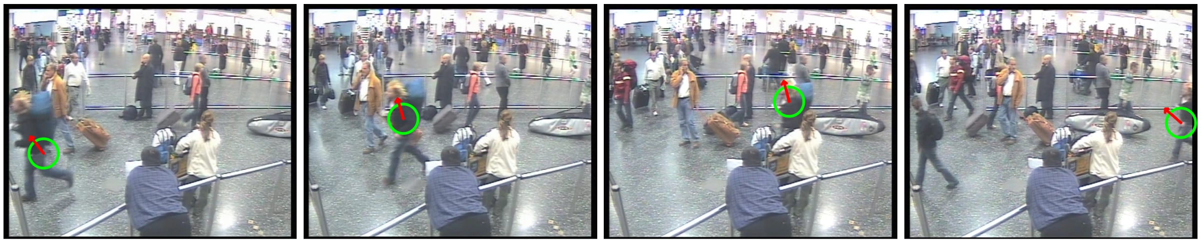


Figure 4.6: A person is running which was detected as *true positive* P_R event.

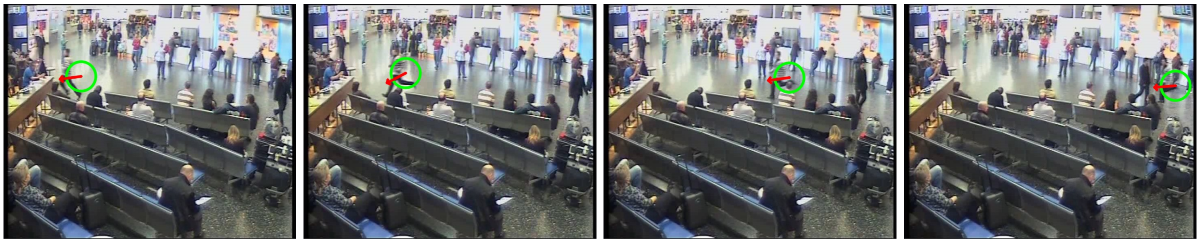


Figure 4.7: A person is crossing the region near to the waiting chair area by running which was detected as *true positive* P_R event.

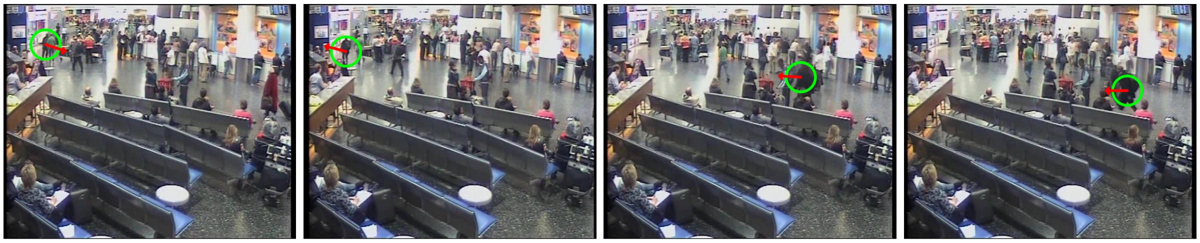


Figure 4.8: A person is also passing by running in different direction near to the waiting chair area which was detected as *true positive* P_R event.

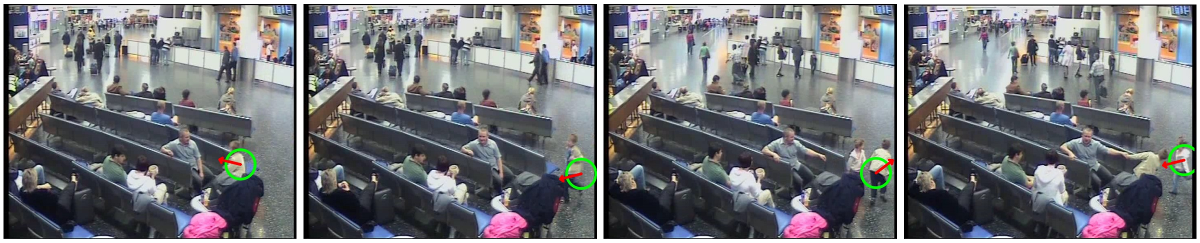


Figure 4.9: Two children are playing sometimes by running which was detected as *true positive* P_R event.

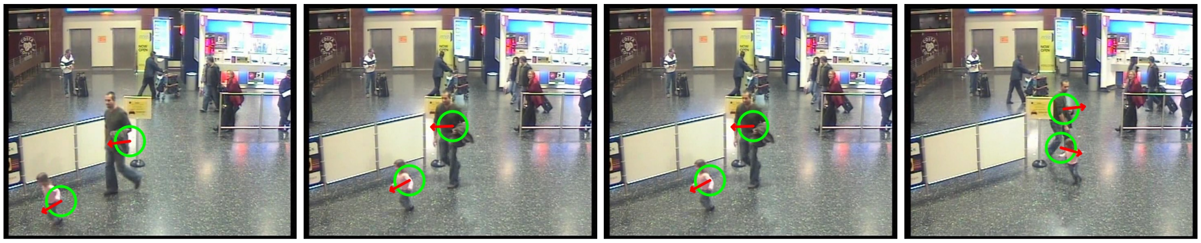


Figure 4.10: A little girl is running while her father is walking on the waiting area which was detected as *false positive* P_R event.

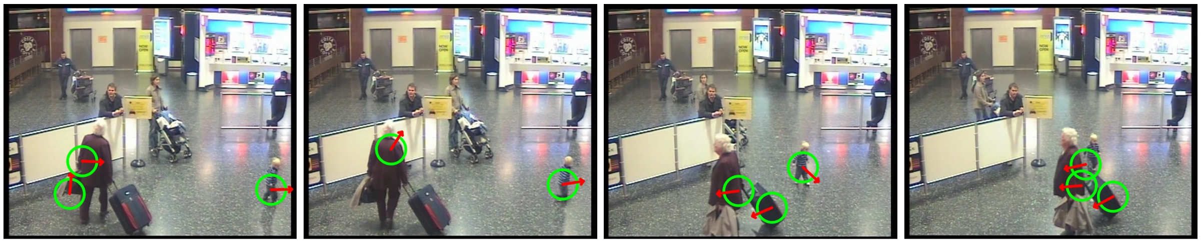


Figure 4.11: A little boy is running while a person is carrying baggages by walking which was detected as *false positive* P_R event.

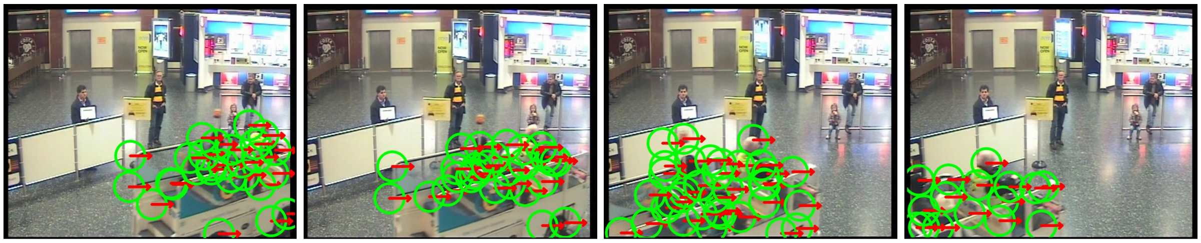


Figure 4.12: A wagon is passing on the waiting area which was detected as *false positive* P_R event.

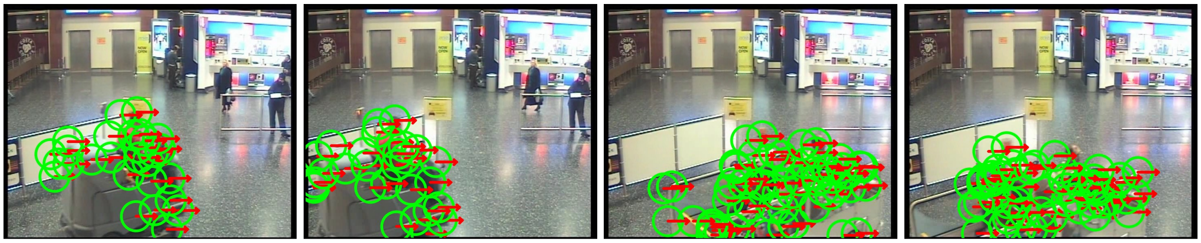


Figure 4.13: A cleaning craft is rolling quickly on the waiting area which was detected as *false positive* P_R event.

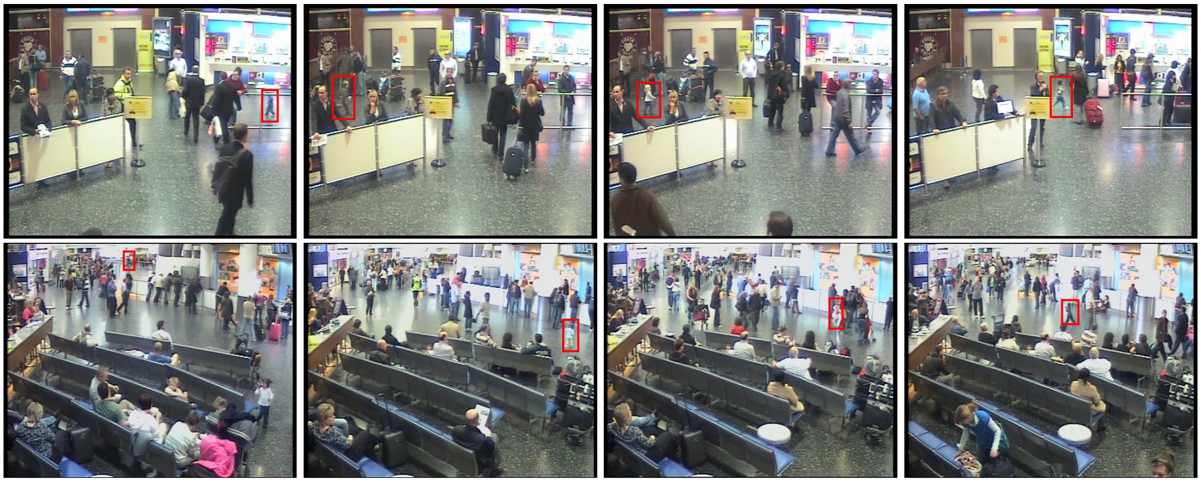


Figure 4.14: *Failure or false negative detection*: Persons inside red marked rectangles cannot be detected as P_R event.

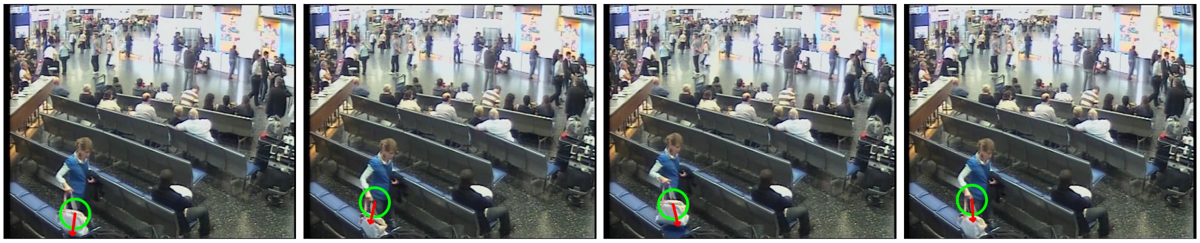


Figure 4.15: A person is putting a hand bag on the waiting bench which was detected as *true positive* O_P event.

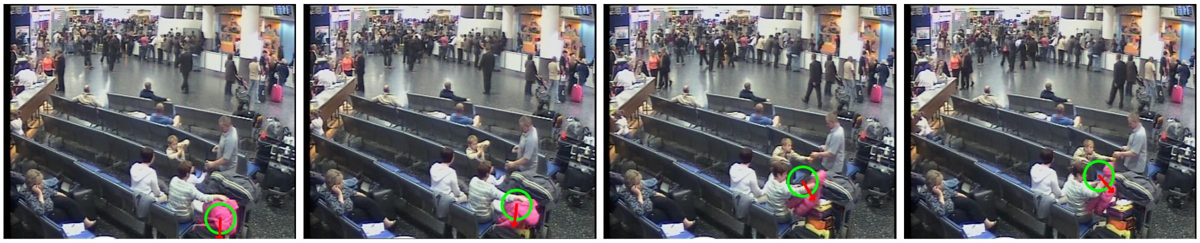


Figure 4.16: A person is putting clothes on a trolley which was detected as *true positive O_P* event.

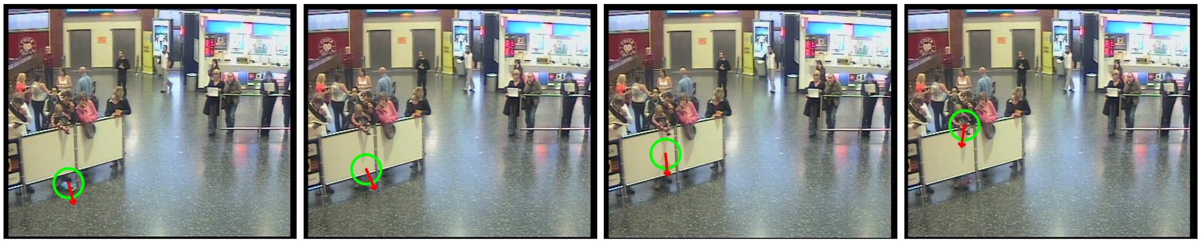


Figure 4.17: Some stuff from the hand of a baby is suddenly dropping on the floor which was detected as *true positive O_P* event.

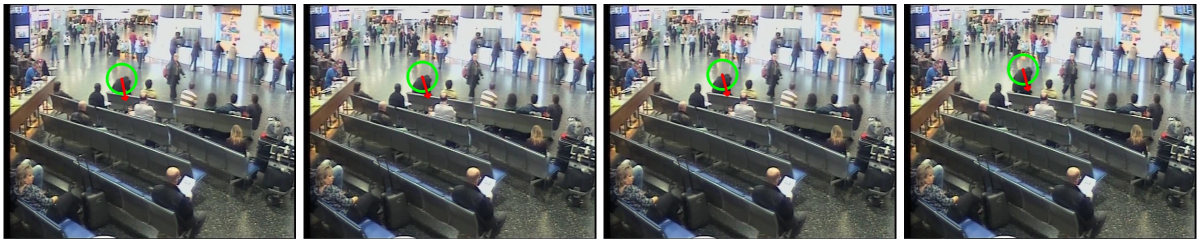


Figure 4.18: Somebody is sitting on the bench which was detected as *false positive O_P* event.

Table 4.1: *Achievement appraisal of the output of the detectors*

Different Measures	Video Events					
	P_R	O_P	O_F	P_M	E_m	P_S
Number of ground truth events ($\mathbf{g_t}$)	95	75	10	55	50	55
Number of false negative events ($\mathbf{f_n}$)	48	54	3	32	38	36
Number of false positive events ($\mathbf{f_p}$)	53	36	3	38	52	43
Number of true positive events ($\mathbf{t_p}$)	47	21	7	23	12	19
Recall rate ($\mathbf{r_r}$) = $\mathbf{t_p}/(\mathbf{t_p}+\mathbf{f_n})$ = $\mathbf{t_p}/\mathbf{g_t}$	49%	28%	70%	41%	24%	34%
Precision rate ($\mathbf{p_r}$) = $\mathbf{t_p}/(\mathbf{t_p}+\mathbf{f_p})$	47%	36%	70%	37%	18%	30%

detected 144, which considered the best *PersonRuns* detection in TRECVID2008 [43]. However, the events detection results of P_R and O_F of the proposed methodology were quite reliable along with false positives, and perhaps (specially P_R) a little bit superior to the result of Orhan et al. [58] where about 45% P_R events successfully detected. The event detectors output of P_M and P_S may have some degree of average acceptance, on the other hand, the output of E_m and O_P event detectors had performed much below than anticipations. Challenges which make circumscribe the performance of event detectors encompass mainly:

Table 4.2: *Selected good results of PersonRuns from TRECVID2008 [43]*

Different Approaches	Different Measures					
	$\mathbf{g_t}$	$\mathbf{f_n}$	$\mathbf{f_p}$	$\mathbf{t_p}$	$\mathbf{r_r}$	$\mathbf{p_r}$
Orhan et al. [58]	314	170	7291	144	45.85%	1.94%
Kawai et al. [97]	314	233	1382	81	25.80%	5.53%
Guo et al. [55]	314	291	639	23	7.32%	3.47%

- a wide variety in the appearance of event types with different view angles
- divergent degrees of imperfect occlusion
- complicated interactions among people
- unilluminated area on the video frame
- varying target sizes and poses
- massive population flow
- miscellaneous scales
- light reflection
- fluctuation
- etc.

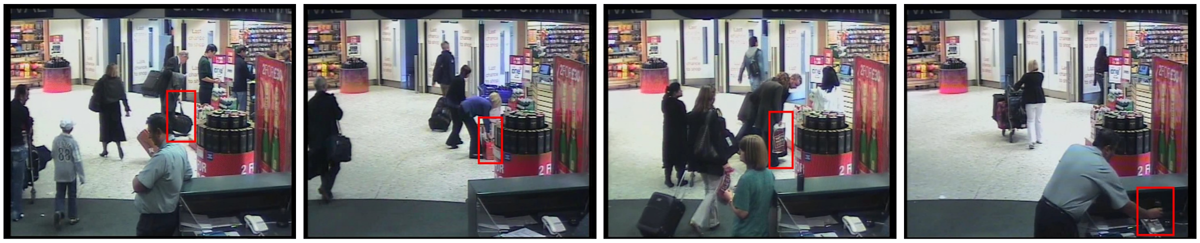


Figure 4.19: *Failure or false negative detection*: Putting object inside red marked rectangles cannot be detected as O_P event.

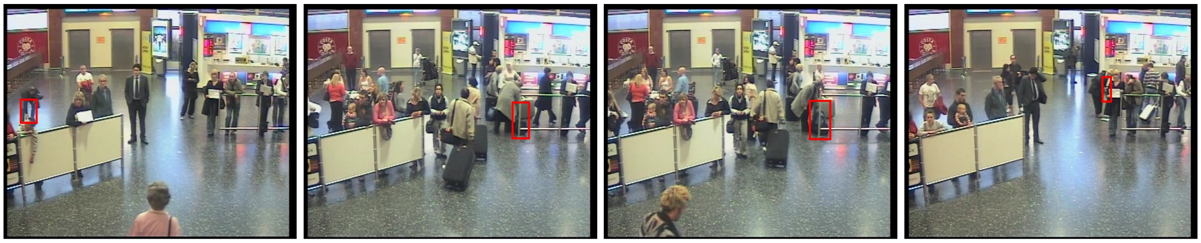


Figure 4.20: Partial occlusion: Putting object inside red marked rectangles cannot be detected as O_P event.

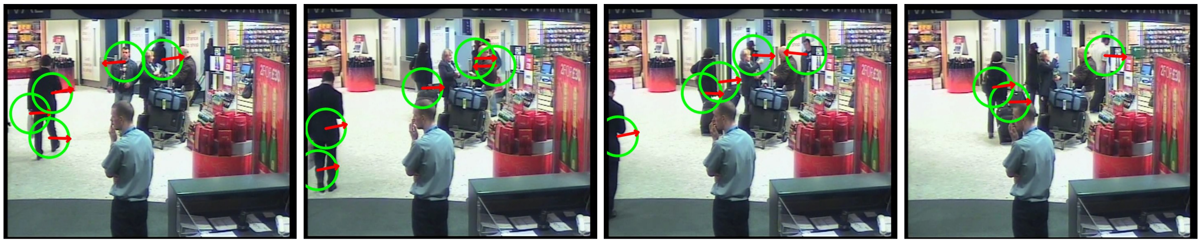


Figure 4.21: Normally people pass the main entry gate unidirectionally. But a person is following opposite of the normal direction which was detected as *true positive* O_F event.

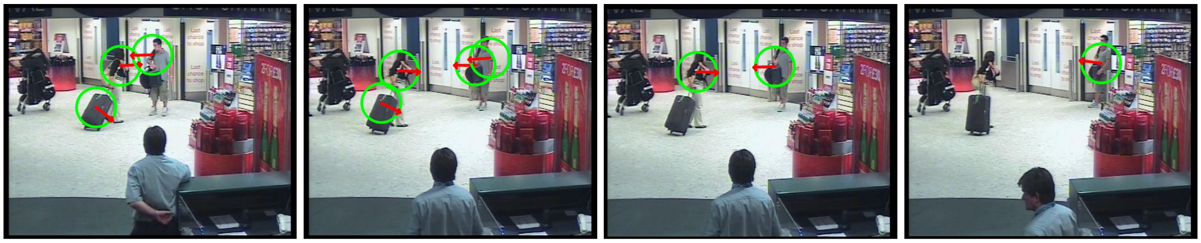


Figure 4.22: A person is slowly coming out from opposite of the normal direction of the main entry gate which was detected as *true positive* O_F event.

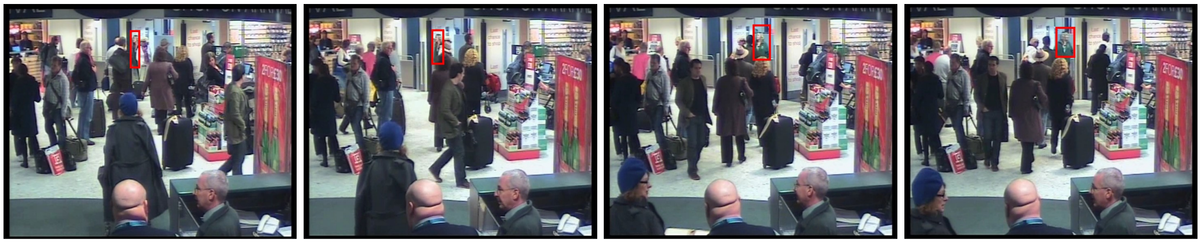


Figure 4.23: *Failure or false negative detection*: A person is coming out from opposite of the normal gate entry direction as marked by red rectangles but this event cannot be detected as O_F event.

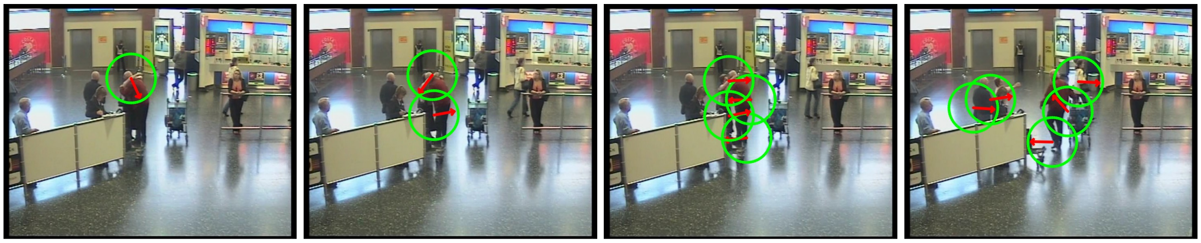


Figure 4.24: The right most image was detected as *true positive* P_M event while the rest images were detected as *true positive* E_m event.

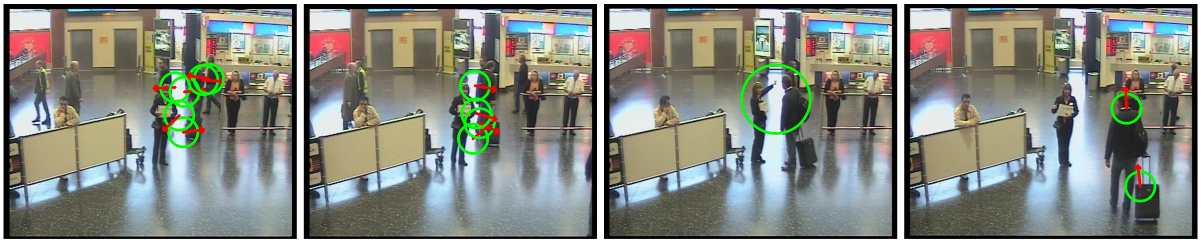


Figure 4.25: Right two images were detected as *true positive* P_M event while left two images were detected as *true positive* P_S event.

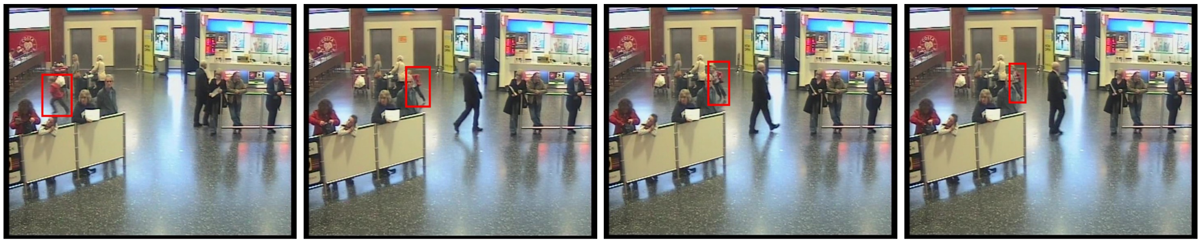


Figure 4.26: *Failure or false negative detection*: A person suddenly scattered from a meeting by a run as marked red rectangles, but this event cannot be detected as P_S event.

4.6 Conclusion

We (in [SD09b]) keyed out a new method which generates automatically *pseudo Euclidian distance* (PED) from the trigonometrically treatments of *motion history blob* (MHB) aiming for different kinds of *video events detection* (VED). The PED is defined as the virtually traveled distance of a moving point inside of a circle towards its direction when it coincides the center of the circle. The concept of PED remains one of the best contributions of this Thesis and would be used in wide variety of computer vision applications. To show the interest of the usage of PED, we proposed a PED based methodology for VED. The results based on the detection of some events at *TRECVID2008* [43] in real videos have been demonstrated. Some results show the robustness of the methodology, while the remains reflect the magnitude of the difficulty of the problem at hand. A vast majority of the false positives is induced by occlusions, cluttered background, and complicated interactions among people. Problem also includes the motion history blob tracking, which heavily depends on direction and position of the blob. Consequently, MHB tracker is inefficient where there needs more information than only direction and position, e.g., if the blob of a person occludes long time or moves long time with occlusion or does not move an elongated period of time etc. The *TRECVID2008* surveillance event detection task is a big challenge to test the applicability of such methodologies in a real world setting. Big challenges include highly clutter, massive population flow, heavy occlusion, reflection, shadow, fluctuation, varying target sizes, low video quality, etc.

Nonetheless, we take the view that we have got ahead much valuable insights to practical problems and future PED based evaluation of more effective VED methodologies have the potential to produce better results.

Chapter 5

Individual Target Tracking in Crowded Scenes

Contents

5.1 Overview

Target or object tracking, which aims at detecting the position of a moving object from a video sequence, is a challenging research topic in computer vision. Tracking algorithms are used in a wide variety of domains, such as robotics, vehicular traffic, navigation and communication systems. The main goal is to obtain a record of the trajectory of the moving object(s) over space and time by processing the sensor data. Reliable tracking methods are of crucial importance in many surveillance systems in order to enable human operators to remotely monitor activity across large environments such as: (i) transport systems (e.g., railway transportation, airports, urban and motorway road networks, maritime transportation, etc.), (ii) banks, shopping malls, car parks, and public buildings, (iii) industrial environments, and (iv) government establishments (military bases, prisons, strategic infrastructures, radar centers, hospitals, etc.). Obstacles in tracking targets can grow due to quick target motion, changing appearance patterns of both the target and the scene, nonrigid target structures, dynamic illumination, inter-target and target-to-scene occlusions, multi-target confusion, etc. Authors in [166] introduced the concept of region covariance matrices for object detection and texture classification. Region covariance matrices can be used for detection of a target in a video and the target can be tracked in the following frames using the same approach as they proposed. By inspiring the concept of [166], authors in [139, 140] proposed approaches for detection, labelling and tracking multiple targets. The targets are represented by region covariance matrices and particle filters perform the target tracking. Their approach would pay the attention but would not be workable for tracking targets separately in the cases of sparse crowd, medium density crowd, and dense crowd as the results were reported only five people scene. Henceforth, it is noteworthy for developing an algorithm capable of handling these type of crowded scenes towards tracking targets on an individual basis. In crowded scenes, tracking a running person is relatively easy as compared to track a single person who is moving at same speed with other people in the scene. The Motion History Blobs (MHB) tracking described in 4.4.1 is inefficient where only direction and position are not enough information for tracking, e.g., if the blob of a person occludes long time or moves long time with occlusion or does not move long time, etc.

In this chapter, we have directed two sort of target tracking methods, fall into 2.3.1 category, as follows: *Firstly*, we (in [SMD08b]) have investigated an object tracking method in video using covariance matrices, as proposed in [166], straightforwardly. The method is relevant to the methods of [139, 140]. Authors in [166] described covariance as a region descriptor and applied it for object detection and texture classification. Covariance matrices do not lie on Euclidean space, therefore it is useable for a distance metric involving generalized eigenvalues which also follows from the Lie group structure of positive definite matrices. *Secondly*, we (in [SDc, SD10b]) have proposed a temporal-spatial domain algorithm to track individual targets in the cases of sparse crowd, medium density crowd, and dense crowd. There are two key differences with previous MHB tracking method (PedVed). Firstly, we have proposed how to extract the region of interest over frame in time (*target*) by means of the MHI function, which uses temporal history of position or motion. The target extraction is same as MHB or *silhouetted region of motion component* (SRMC), except after getting center of each MHB or SRMC, we consider the original video frame and get that rectangular MHB from the video frame as a target region. Secondly, we have introduced how to use the phase-correlation techniques for *targets* detection and tracking using distinct sharp peaks from the obtained peaks of target regions and the next frame's candidate regions. If two candidates or targets are similar, then their phase-correlation function gives a distinct sharp peak. Conversely, the peak of two dissimilar targets or candidates drops significantly. The inspiration of the usage of phase-correlation technique is the fact that unlike many spatial-domain algorithms, the phase-correlation means is resilient to noise, occlusions, and other defects typical of medical or satellite images.

5.2 Target Tracking using Covariance Matrices

The goal of the detection drudgery is to identify the presence and possibly the location of a given object in a video sequence, whereas the goal of the tracking task is to estimate the successive positions of an object or region using discriminating features through video frames. Tuzel et. al. [166] proposed to use the covariance of several image statistics computed inside a region of interest, as the region descriptor. They used integral images for fast covariance computation. Integral images are intermediate image representations used for fast calculation of region sums [170]. We have investigated this covariance-based descriptor for tracking objects in a video sequence.

5.2.1 Image features

Good feature selection plays a critical role for detection and classification issues. Color, edges, optical flow, texture, gradient, and filter responses are some common example of features. Image features such as color, gradient, and filter responses, used in [28, 130], are not robust for tracking in the case of illumination changes or non-rigid motion. As opposed to rigid motion, which is a transformation consisting of rotations and translations that leaves a given arrangement unchanged. Besides, effective matching algorithms are restricted by the high dimensional feature representation. AdaBoost [66] selects a small number of critical visual features from a larger set and yields extremely efficient classifiers. Low dimensional projections have been used for classification in [165] and tracking in [21]. In [37], histograms were widely used for non-rigid object tracking. Fast histogram construction methods were explored to find an optimum and complete solution for the histogram-based search problems [143]. However, the joint representation of several different features through histograms is exponential with the number of features [166]. The integral images were used for fast calculation of region histograms in [143] and for fast calculation of region covariances in [166]. The idea of integral image was first commenced in [170] for the computation of Haar-like features very quickly. Although they found better performance for face detection, the algorithm requires a long training time for the object classifiers. In [122], scale-space extrema are detected for keypoint localization and arrays of orientation histograms were used as keypoint descriptors. The descriptors are very effective in matching local neighborhoods but miss global context information [166].

5.2.2 Covariance as a Region Descriptor

Covariance is a statistical measure of correlation of the continual changes from one point – or condition – to another of two different quantities. A covariance matrix is merely collection of various covariances in the form of a square matrix. There are several benefits of using covariance matrices as region descriptors. A single covariance matrix extracted from a region can match the object in dissimilar views and poses, assuming that the covariance of a distribution is enough to discriminate it from other distributions. Let I be either one dimensional intensity or three dimensional color image. Assume F be the $W \times H \times d$ dimensional feature image, where W, H, and d are the width, height, and dimension (number of color channels) of the feature points of the image respectively, extracted from I. For each feature point, $F(x,y) = \phi(I,x,y)$, where the function ϕ can be any mapping such as intensity, color, gradients, filter responses, etc. For a given rectangular region $R \subset F$, deeming that z_k where $k \in [1,n]$ is the d-dimensional feature points inside region R where n be the total number of points in R. The region R with the $d \times d$ covariance matrix of the feature points can be presented as:

$$C_R = \frac{1}{n-1} \sum_{k=1}^{n} (z_k - \mu)(z_k - \mu)^T \tag{5.1}$$

where μ is the mean of the points. The C_R is invariant regrading the ordering and the number of points in the region of R. This signifies a certain scale and rotation invariance over the regions in different images. The covariance matrix provides a natural way of melting multiple features which might be correlated [166]. The diagonal entries of the covariance matrix represent the variance of each feature and the non-diagonal entries represent the correlations. The covariance matrices are low-dimensional compared to other region descriptors [166] and due to symmetry C_R has only $\frac{d^2+d}{2}$ different values. Expanding the mean and rearranging the terms of the above equation, it is easy to write the (i,j)-th element of the covariance matrix of the feature points as:

$$C_R(i,j) = \frac{1}{n-1}\left[\sum_{k=1}^{n}(z_k(i)z_k(j) - \frac{1}{n}\sum_{k=1}^{n}z_k(i)\sum_{k=1}^{n}z_k(j)\right]. \tag{5.2}$$

To find the covariance in a given rectangular region R, it is important to compute the sum of each feature dimension, $z(i)$ where $i \in [1,n]$, as well as the sum of the multiplication of any two feature dimensions, $z(i)z(j)$ where $\{i,j\} \in [1,n]$. We can construct $d+d^2$ integral images for each feature dimension $z(i)$ and multiplication of any two feature dimensions $z(i)z(j)$. Integral images could be used to calculate either region histograms or region covariances. Each pixel of the integral image is the sum of all the pixels inside the rectangle bounded by the upper left corner of the image and the pixel of interest [166]. For an intensity image I, its integral image is defined as:

$$Integral\,Image\,(x',y') = \sum_{x<x',y<y'} I(x,y). \tag{5.3}$$

5.2.3 Target Tracking

We track a given object from a video frame by estimating its positions in the next video frame where the object may appear with a change in pose, or after a non-rigid transformation. Feature matching, which is a simple nearest neighbor search under distance metric, is accomplished quickly using the integral images. In the search process, we use a variable size sliding window at nine different scales (four smaller, four larger, with a 15% scaling factor between two consecutive scales). For the smallest size of the window, we jump 3 pixels horizontally or vertically between two search locations. For larger windows, we jump 15% more and round to the next integer at each scale, as recommended in [1]. Since covariance matrices do not lie in an Euclidean space, it is often appropriate to use a distance metric involving generalized eigenvalues. To increase robustness towards possible occlusions and large illumination changes, we consider 5 covariance matrices extracted from overlapping regions of the object feature image, corresponding to (1) the whole region, (2-3) the half-left/right sub-regions, and

(4-5) the half-up/down subregions. We search the target frame for a region having the closest covariance matrix and the dissimilarity is measured through following equations. The distance measure proposed in [65] to measure the dissimilarity of two covariance matrices is:

$$\rho(c_1, c_2) = \sqrt{\sum_{i=1}^{n} ln^2 \lambda_i(c_1, c_2)} \tag{5.4}$$

where $\{\lambda_i(c_1, c_2)\}$ and $i \in [1, n]$ are generalized eigenvalues of c_1 and c_2, computed from:

$$\lambda_i c_1 x_i - c_2 x_i = 0 \tag{5.5}$$

where $i \in [1, d]$ and $x_i \neq 0$ are the generalized eigenvectors. The dissimilarity of the object model and a target region is computed by:

$$\rho(O, T) = arg\,min_j \left[\sum_{i=1}^{5} \rho(c_i^O, c_i^T) - \rho(c_j^O, c_j^T) \right] \tag{5.6}$$

where O, T, c_i^O, and c_i^T indicate the object, the target, the object covariance and target covariance for the 5 sub-regions, respectively. Target region with smallest dissimilarity is selected as the matching region.

5.2.4 Experimental Results

To conduct experiment, we have mainly relied on the people tracking data set of the *PETS 2009 Benchmark Data* so-called *Dataset S2: People Tracking* [42], which is organized as sparse crowd, medium density crowd, and dense crowd.

Fig.5.1 depicts the tracking results of a person inside the red marked rectangle from some video

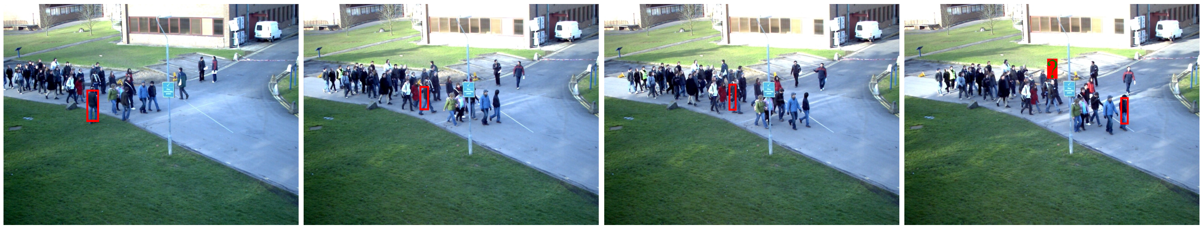


Figure 5.1: The images concern the possible extended method of [166] as implemented in [SMD08b].

sequences of [42]. The results make noticeable that a single covariance matrix extracted from a region of interest can match the region in some else views and poses. Nevertheless, if there are large scale, orientation, and illumination changes, the detection is erroneous as can be seen the last image where the target is undetected (marked as "?").

5.3 A Temporal-spatial Framework

We have proposed a temporal-spatial domain algorithm to track individual targets in the cases of sparse crowd, medium density crowd, and dense crowd. There are two key contributions within this approach.

Firstly, we have proposed how to extract the target (region of interest over frame in time) and candidate (region of possible target over next frame in time) regions from the silhouetted structures of more recently moving pixels of the object of interest by combining two techniques, namely *motion history image* MHI [22] and *Hu's moments* [80]. MHI, which uses temporal history of position or motion, helps to create a *silhouetted region of motion component* (SRMC) while Hu's moments find the center of gravity of SRMC. A key advantage behind of this hybrid technique is that it is not necessary to search the possible target region everywhere on the candidate frame except for the candidate regions. Consequently, the searching process becomes overpowering rapid.

Secondly, we have introduced individual target tracking techniques using distinct maximum peaks, obtained by phase-correlation techniques, from the resulting peaks of target regions and the candidate frame's candidate regions. When two target and/or candidate regions are similar, their phase-correlation function gives a distinct sharp peak (see Fig 5.3 (a) and (b)). Conversely, the peak of two dissimilar target and/or candidate regions drops in a clearly noticeable manner (see Fig 5.3 (c)). The motivation of the usage of phase-correlation techniques is the fact that unlike many spatial-domain algorithms, the phase-correlation means is resilient to noise, occlusions, and other defects typical of medical or satellite images.

In temporal-spatial framework, we ([SDc]) focus on the following algorithmic steps:

- Foreground is estimated to get SRMC;
- Segmentation is performed on the obtained SRMCs to get a sequence of SRMC;
- Center of mass or center of gravity or centroid is estimated to get the center of each SRMC (*Hu center*);

- Target and/or candidate regions are estimated by the coordinates of the centroid;

- Phase-correlation techniques are performed to get the highest peak heights for target and/or candidate;

- The highest peak heights are processed by a tracking algorithm, which tracks target based on highest geometric mean. If the highest geometric mean is greater than an experienced dynamic cut-off then tracking is typically performed, else the target is motionless or the state of being occluded or out of the video frame.

In evaluation method, if a ground truth ellipse interacts with the system output rectangle, then we said a correct detection has been performed. The video sequences of PETS 2009 Benchmark data have been considered for performance evaluation of the approach.

5.3.1 Foreground Estimation and Segmentation

In video surveillance system, the first step is to have a good process for detection of foreground objects. The background image is associated to the static constituent of the scene and the foreground image is associated to the dynamic constituent of the scene. Consequently, foreground objects are the moving objects on the scene. Instead of using the mixture-of-Gaussian based adaptive background modeling method [161] to generate foreground mask for each frame, we use on the silhouetted structures of more recently moving pixels of the object of interest. With this end, we rely on the motion history image (MHI) function which was introduced by [22]. The strength of the function is that it is easy to implement and adds little computational cost, a detailed about MHI has been discussed in the subsection 4.3.1. The result of the function is a scalar-valued image where more recently moving pixels are brighter. The function is called each time a new image is received and the corresponding silhouetted image is formed. We only deal with that silhouetted brighter part while suppressing other parts, which we named *silhouetted region of motion component* (SRMC) or *motion history blobs* (MHB), which is produced by more recently movement of part or the whole of the object of interest. To get sequence of SRMCs, it is important to segment the obtained SRMCs. On segmentation, we count number of points within each SRMC so that it will be easy to check for the case of very little motion (e.g., noise), which will be neglected under an experienced threshold. Finally, we estimate the *center of mass* for each remaining SRMC.

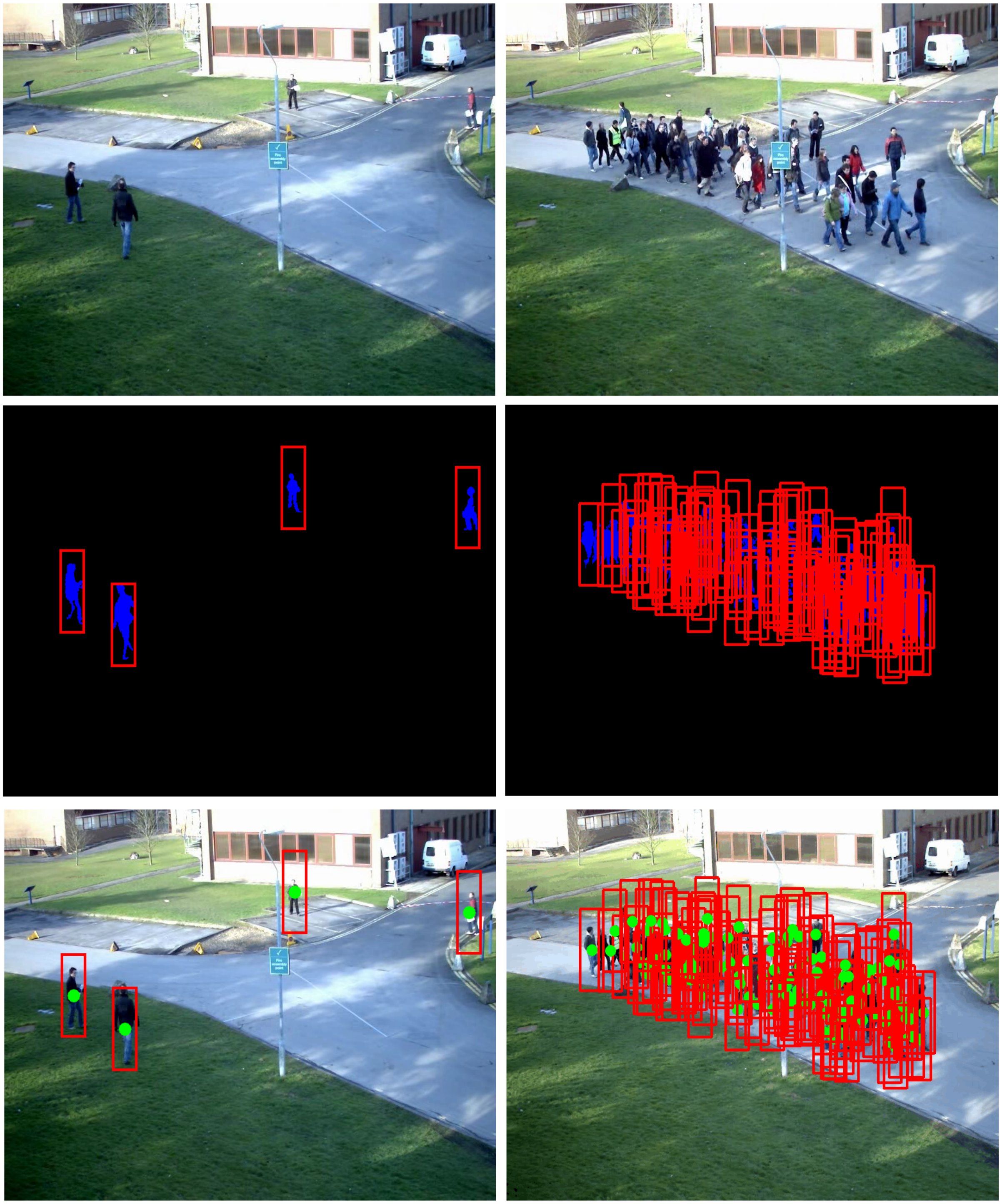


Figure 5.2: The 1st, 2nd, and 3rd rows depict, respectively, the camera view, the *silhouetted region of motion components* hedged in red colored fixed rectangles, and the target regions enclosed by identical rectangles. Green points (*Hu centers*) are the *centers of mass of SRMCs* estimated on applying Hu's moments.

5.3.2 Center of Mass Estimation

Center of mass or center of gravity or centroid is the term given to the center of a region, area, etc. Image moments are useful to describe objects after segmentation. Moments give us an indication of the center of SRMC (e.g., pixel values and regions). The use of moments for image analysis and object representation was introduced by Hu [80]. The moments of Hu are fast and accurate to represent shapes than most of the other methods. A detailed investigation of centroid estimation has been described in the subsection 4.3.2. The *coordinates of the Center of Mass* can be defined through moments as:

$$\widehat{x} = \frac{m_{10}}{m_{00}} \quad \widehat{y} = \frac{m_{01}}{m_{00}} \tag{5.7}$$

To achieve invariance with respect to orientation and scale, the normalized central moments are formulated as follows:

$$\eta_{pq} = \frac{1}{\mu_{00}^{[(p+q)/2]+1}} \sum_{-\infty}^{\infty}\sum_{-\infty}^{\infty}(x-\widehat{x})^p(y-\widehat{y})^q\phi(x,y) \tag{5.8}$$

where $(p+q) \geq 2$. For any SRMC, a potentially infinite number of moments $\{\eta_{pq}\}$ can uniquely describe SRMC. These image moments are invariant with respect to translation and scale operations, but an infinite number of moments are obviously impractical and only ten moments up to the third order are practical. Since the centroid of SRMC defines a unique location with respect to the object, it can be used a reference point to describe the position of the object on the camera view image. As a result, we represent all coordinates of the *centroid of SRMC* (*Hu center*) to their corresponding positions on the original image with fixed rectangles along with identical centers. Such representation of object shapes on the camera view frame are highly commendable for the phase-correlation techniques. The Fig. 5.2 hints the process for sparse (in left side) and dense (in right side) crowded scenes. The superiority of this technique is that it is not important to search the object through the whole image during tracking. As a result, the searching process becomes swift overwhelmingly.

5.3.3 Phase-correlation Techniques

The input to a phase-correlation algorithm [107] may consist of a pair of images or a pair of co-sited rectangular blocks of identical dimensions belonging to consecutive frames or fields of a moving sequence sampled at successive time interval; and the output commonly concerns single *phase-correlation peak*. Let $I_0(x,y)$ and $I_1(x,y)$ be the two images which differ only by a displacement of $(\Delta x, \Delta y)$, i.e.,

$I_1(x,y) = I_0(x-\Delta x, y-\Delta y)$. Their corresponding Fourier transformations F_0 and F_1 will be related by:

$$F_1(u,v) = F_0(u,v)e^{-2\pi j(u\Delta x+v\Delta y)}. \tag{5.9}$$

One can then calculate the normalized cross-power spectrum to factor out the phase difference:

$$P(u,v) = \frac{F_1(u,v)F_0^*(u,v)}{|F_1(u,v)F_0^*(u,v)|} = \frac{F_0(u,v)F_0^*(u,v)e^{2\pi j(u\Delta x+v\Delta y)}}{|F_0(u,v)F_0^*(u,v)e^{2\pi j(u\Delta x+v\Delta y)}|} = e^{2\pi j(u\Delta x+v\Delta y)} \tag{5.10}$$

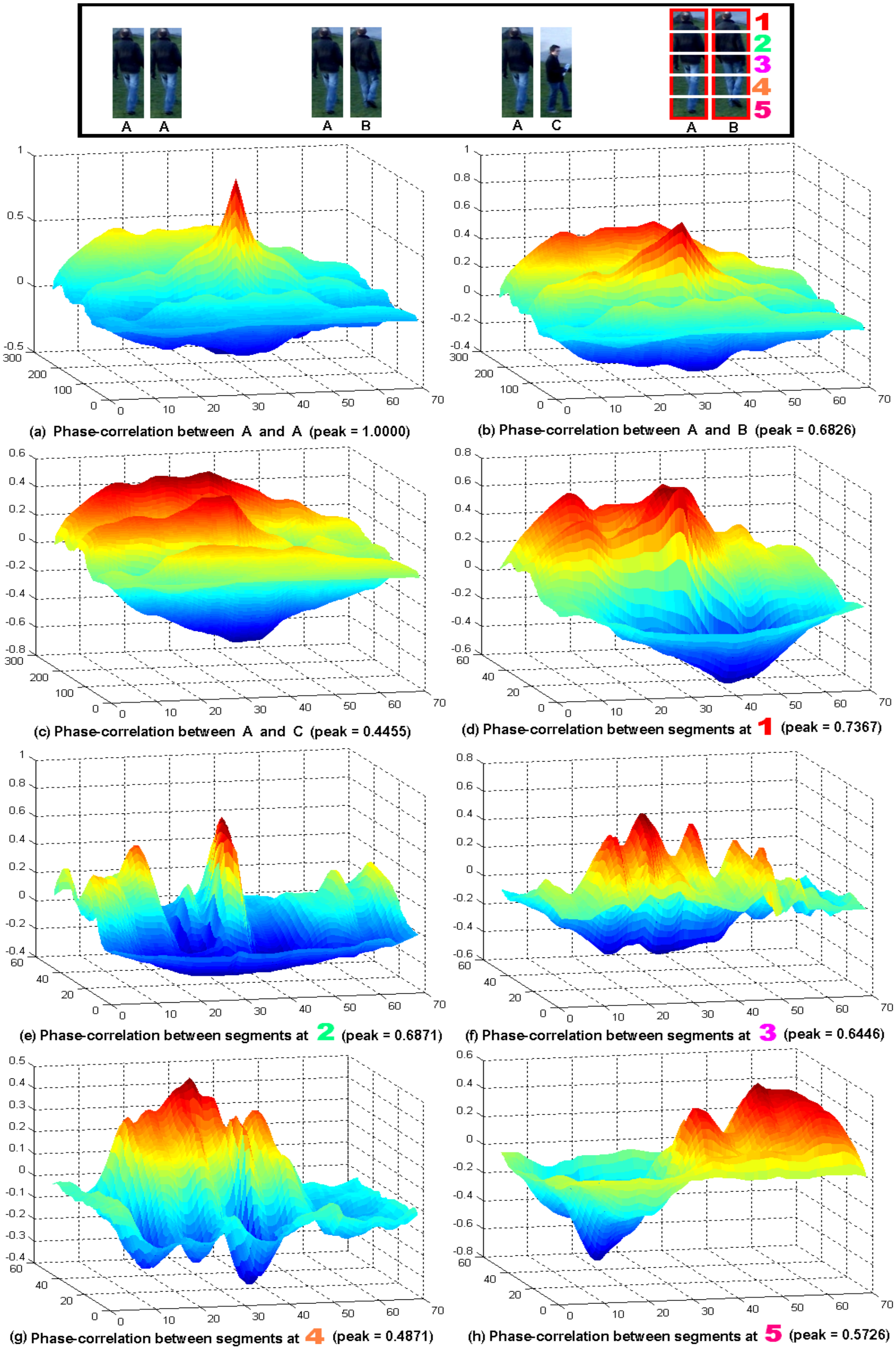
where F_0^* is the complex conjugate of F_0. It directs that the phase of the normalized cross-power spectrum is equivalent to the phase difference between the images I_0 and I_1. If the two images are identical but shifted, the result will be an impulse at $(\Delta x, \Delta y)$ which represents the translation displacement between the two images. By computing the linear phase of $P(u,v)$, the translation displacement can be determined. Furthermore, the inverse Fourier transform of a complex exponential is a Kronecker delta, i.e., a single peak. By taking the *Inverse Fourier Transform* of $P(u,v)$, it is easy to obtain its spatial representation:

$$p(u,v) = IFT(P(u,v)) = \delta(x+\Delta x, y+\Delta y) \tag{5.11}$$

where $p(u,v)$ is a *Kronecker delta* function which is zero everywhere except at the displacement, namely the *phase-correlation peak*. This result would have been brought together calculating the cross-correlation directly. But the boon of this phase-correlation method is that the discrete Fourier transform and its inverse can be performed using the fast Fourier transform, which is much faster than correlation for large images. Over and above computational efficiency, phase-correlation puts up key plus points in terms of its convincing response to edges and outstanding picture features, its immunity to illumination changes and moving shades, and its finesse to assess notable dismissals.

5.3.4 Tracking Techniques

To track target in the scene, knowledge about what the target looks like is absolutely essential. Such cognition can be gained from the rectangular template in the original video frame as marked on Fig.5.2. On defining the target region T, we calculate the phase-correlation between the target region and each rectangular template from the next frame (candidate frame) e.g., Fig.5.3 (b) and (c). We search for the first order candidate peak height which is the highest peak height among the resulting peak heights. We may select a second order candidate peak height, which is supposed to be the best match of the target in



(a) Phase-correlation between A and A (peak = 1.0000)

(b) Phase-correlation between A and B (peak = 0.6826)

(c) Phase-correlation between A and C (peak = 0.4455)

(d) Phase-correlation between segments at 1 (peak = 0.7367)

(e) Phase-correlation between segments at 2 (peak = 0.6871)

(f) Phase-correlation between segments at 3 (peak = 0.6446)

(g) Phase-correlation between segments at 4 (peak = 0.4871)

(h) Phase-correlation between segments at 5 (peak = 0.5726)

Figure 5.3: **Peak** value shows the highest peak height and **A** is target while **B** & **C** are candidates.

the next frame, by finding the highest peak height from all of the calculated first order candidate peak heights. But there maybe more than one first order candidate peak heights which are close to each other and cause ambiguity and difficulty in selection process. To minimize this dismay, we count each first order candidate peak height which has exceeded a 50% cut-off of the current target peak height; namely the second order candidate peak height $\mathbf{C}$. Such a cut-off makes the processing faster by suppressing the first order candidate peak heights which do not exceed the 50% of the current target peak height. We segment each $\mathbf{C}$ as well as $\mathbf{T}$ to get their respective segmented phase-correlation highest peak heights. Afterwards, we estimate the *geometric mean* of the obtained unsegmented and segmented highest peak heights. Suppose that there are n numbers of $\mathbf{C}$ and each $\mathbf{C}$ is segmented into s number of small regions. Upon applying phase-correlation technique, let $\mathbf{T}\textcircled{$\oint$}\mathbf{C}$ denote the obtained highest peak height between $\mathbf{T}$ and $\mathbf{C}$. Similarly, let $\mathbf{T_i}\textcircled{$\oint$}\mathbf{C_i}$ denote the highest peak height between segments $\mathbf{T_i}$ and $\mathbf{C_i}$, where $i \in s$. Thus, a set of highest peak height $\Omega_{s+1} = \{\mathbf{T}\textcircled{$\oint$}\mathbf{C}, \mathbf{T_1}\textcircled{$\oint$}\mathbf{C_1}, \mathbf{T_2}\textcircled{$\oint$}\mathbf{C_2}, \ldots, \mathbf{T_s}\textcircled{$\oint$}\mathbf{C_s}\}$ can be gained and their geometric mean can be estimated by means of:

$$G_m = \left[\prod_{i=1}^{s+1} \Omega_i\right]^{\frac{1}{s+1}} = exp\left[\frac{1}{s+1}\sum_{i=1}^{s+1} log_e \Omega_i\right]. \tag{5.12}$$

Consequently, we have n numbers of G_m of which we may have to select the best match for T. An example of the segmented phase-correlation has been depicted on the Fig. 5.3 (d), (e), (f), (g), and (h) mooting $s = 5$ segments and their geometric mean with (b) is 0.6293. Estimation is better if the number of segments is higher. However, finally, we find the special $\mathbf{C}$ which posses the highest geometric mean as:

$$Highest\, geometric\, mean = arg \max_{k=1\ldots n} [G_m]_k. \tag{5.13}$$

If the highest geometric mean is greater than a dynamic experienced threshold then the target is said to be *detected* and we *relocate the target region* to that $\mathbf{C}$ corresponding region and repeat the process; else the target is deemed as motionless or the state of being occluded or out of the video frame. If the target does not match in some extended period of video sequences then it counts as out of the video frame; else it views as motionless or the state of being occluded and repeat the process without relocate its region. Similar way the algorithm is suitable for tracking multiple targets individually and simultaneously.

5.3.5 Experimental Results

To conduct experiment, we have heavily relied on the people tracking data set of the *PETS 2009 Benchmark Data* so-called *Dataset S2: People Tracking* [42], which is organized as sparse crowd, medium density crowd, and dense crowd. The experienced dynamic cut-off for the next frame has been deemed as 75% of the highest geometric mean of the current target region with $s = 10$ segments.

5.3.6 Evaluation Method

We have annotated the target person ourselves. In general, the annotator draws an ellipse where a target is taking place and keeps track of the target throughout the under investigation frame sequences. Each annotator works separately and the number of employed annotators depends on the number of individual targets will have to be tracked simultaneously. Normally, our system draws rectangle as a target tracking result. If a ground truth ellipse overlaps with the system output rectangle in any frame, we define it to be a *true positive* or *correct detection*. If there is no overlapping system output within the ground truth ellipse, we define it as a *false negative* or *miss detection*. If there is no ground truth but there is a system output, then it is a *false positive* or *false alarm*. If there is no ground truth ellipse as well as there is a system output rectangle, then we define it as a *true negative* or *correct rejection*. Normally, true negative occurs when the target is motionless or the state of being occluded or out of the video frame. We have considered different statistical probability measures e.g., *precision rate* (also called *sensitivity*), *recall rate* (also called *true positive rate*), *specificity* (also called *true negative rate*), and *accuracy*. Precision is the probability that can be seen as a measure of exactness or fidelity, whereas recall is also the probability that is a measure of completeness. A sensitivity of 100% means that the algorithm recognizes all actual positives, whereas a specificity of 100% means that the algorithm recognizes all actual negatives. Accuracy indicates proximity of measurement results to the true value, precision to the repeatability or reproducibility of the measurement. Normally, a measurement system is said to be valid if it is both accurate and precise.

5.3.7 Evaluation Result Analysis

Although manually labeling is a time consuming task, we have verified the approach by such labeling at different challenging video sequences involving illumination change, flash light, different scale, and low video quality. In principle, we can track multiple individual targets simultaneously. The most common cause of false alarms comes from occlusion cases in the video sequences where our system considers another person to be a target due to its highest geometric mean, which is very close to the latest highest geometric mean of the real target. Another common reason of false alarms is due to small image size of targets which are far away from the camera. In the following, we have presented the tracking results of two persons (targets) in dense crowd and six persons (targets) in medium dense crowd from two different video sequences of [42]. Two and six annotators were employed separately to create targets ground truth data from the under considering video sequences. As a result, the annotators produced 8 sets of ground truth data individually. In case of two individual persons tracking considering 114 video frames, two annotators were employed separately to produce two separate data sets containing 68 & 74 frames with ellipses and 46 & 40 frames without ellipse, for first and second persons, respectively. If a target is either motionless or in the state of occlusion or out of video frame, then the annotator does not mark ellipse on respective frame. Tracking results of 60 video frames have been presented in case of six single persons.

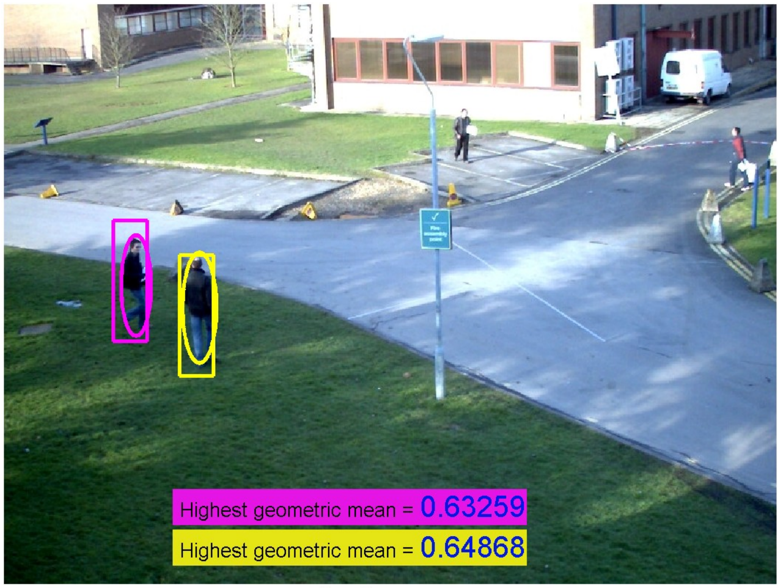

Highest geometric mean = 0.63259
Highest geometric mean = 0.64868
Frame 4

Figure 5.4: Two single persons (magenta and yellow note to A1.2 and B1.2, respectively) *true positive* tracking results in *dense crowd* from the video sequences of PETS2009 [42]. Ellipse and rectangle denote ground truth and algorithm output, respectively.

Fig.5.4, 5.5, 5.6, 5.7, and 5.8 show some example images for the results of two and six individual persons tracking. The results of the algorithm (rectangles) have been interacted most of the target

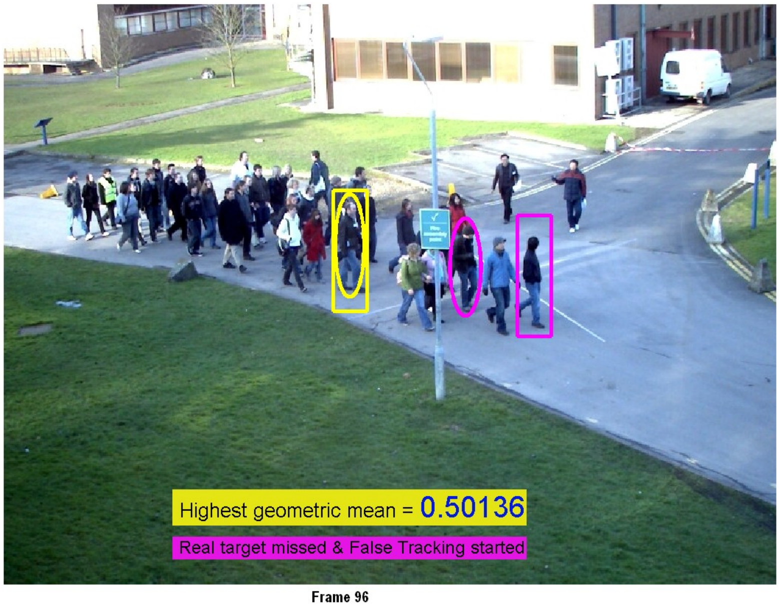

Highest geometric mean = 0.50136
Real target missed & False Tracking started
Frame 96

Figure 5.5: Tracking of person A1.2 is *true positive*, while after occlusion person B1.2 is *false negative*.

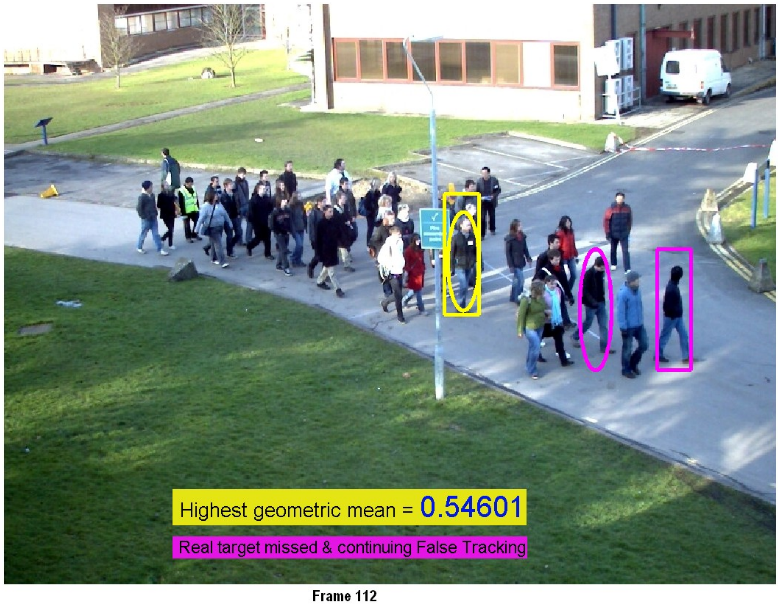

Highest geometric mean = 0.54601
Real target missed & continuing False Tracking
Frame 112

Figure 5.6: On occlusion, tracking of person A1.2 is still *true positive* and person B1.2 is a failure.

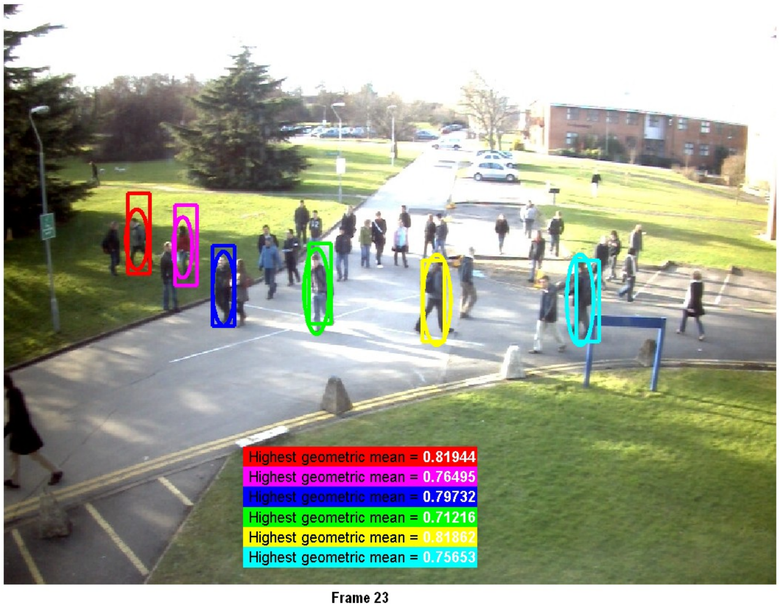

Highest geometric mean = 0.81944
Highest geometric mean = 0.76495
Highest geometric mean = 0.79732
Highest geometric mean = 0.71216
Highest geometric mean = 0.81862
Highest geometric mean = 0.75653
Frame 23

Figure 5.7: Six single persons *true positive* tracking results in *medium dense crowd* from the PETS2009 [42] video sequences. Ellipse and rectangle note to ground truth and algorithm output, respectively.

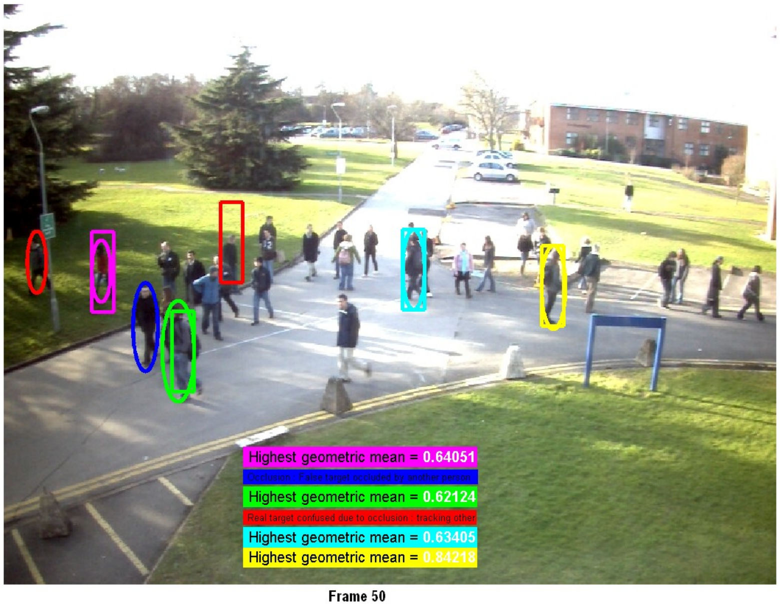

Highest geometric mean = 0.64051
Occlusion : False target occluded by another person
Highest geometric mean = 0.62124
Real target confused due to occlusion : tracking other
Highest geometric mean = 0.63405
Highest geometric mean = 0.84218
Frame 50

Figure 5.8: Two single persons tracking (red and blue) scored *failure* while rests are still *true positive*.

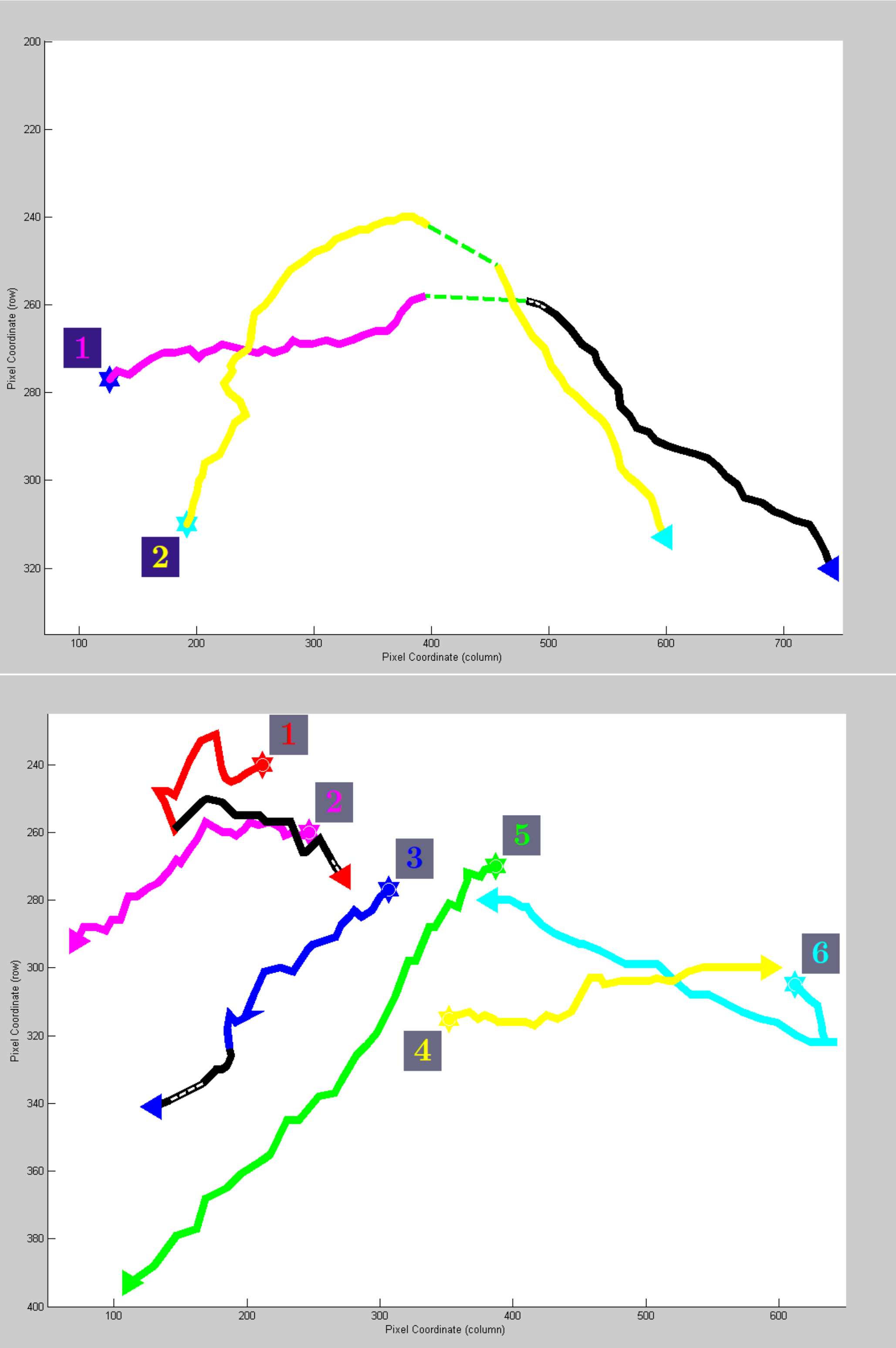


Figure 5.9: Above and underneath graphs, respectively, represent trajectories of the centers of mass for two and six single persons tracking results of the algorithm. Heavy black line shows *false negative* tracking while white dots inside it point to *false positive*. Heavy green dot lines indicate *occlusion*.

ground truth regions (ellipses) accurately. The results represent that the system is very accurate and its sensitivity to the effects of deviation in noise and lighting, which guarantees high-quality performance on fades, targets moving in and out of the shadow, and flashes of light. Fig.5.9 demonstrates the trajectories (of the centers of mass) of two and six individual persons tracking results, which concern successful detection, false alarm, missed detection, correct rejection, etc.

Table 5.1: *Tracking results analysis*

Different Measures	2-Person		6-Person					
	1	2	1	2	3	4	5	6
Number of ground truth frames (g_t)	114	114	56	60	60	60	60	60
Number of false negative frames (f_n)	26	0	32	0	26	0	0	0
Number of false positive frames (f_p)	6	0	4	0	10	0	0	0
Number of true positive frames (t_p)	42	74	24	60	24	60	60	56
Number of true negative frames (t_n)	46	40	0	0	10	0	0	4
Recall Rate = $t_p/(t_p+f_n) = t_p/g_t$	0.36	0.64	0.42	1.00	0.40	1.00	1.00	0.93
Specificity = $t_n/(t_n+f_p)$	0.88	1.00	0.00	0.00	0.50	0.00	0.00	1.00
Precision rate = $t_p/(t_p+f_p)$	0.87	1.00	0.85	1.00	0.70	1.00	1.00	1.00
Accuracy = $(t_p+t_n)/(t_p+t_n+f_p+f_n)$	0.73	1.00	0.40	1.00	0.48	1.00	1.00	1.00
Average Recall Rate	**72%**							
Average Specificity	**42%**							
Average Precision rate	**93%**							
Average Accuracy	**82%**							

The above graph of Fig.5.9, the blue and cyan stars are the starting places of 1st and 2nd persons, while heavy magenta and yellow lines belong to the trajectories of their centers of mass, respectively. In the video sequences, both persons gradually went forward to the incoming crowd and faced both light reflection and occlusion. Heavy green dot lines depict their occlusion, consecutively. False positive tracking started as soon as the 1st person had partially occluded, a region of very similar person was falsely recognized and tracked over the next video sequences. Primarily, the lighting condition helped to make the falsely recognized person similar to the real person. As a result, the highest geometric mean of that person's region was closer to than that of the real person. White dots inside heavy black line indicate the false alarm. False negative tracking has been presented by heavy black line. False negative was begun abruptly when the real person fully appeared again. But the algorithm could not detect any more the real person as it had already considered the false person as a real one. Irrespective of light reflection and occlusion, 2nd person was successfully recognized and tracked over the video sequences.

The beneath graph of Fig.5.9, stars are the starting places 1st, 2nd, 3rd, 4th, 5th, and 6th persons, while heavy red, magenta, blue, yellow, green, and cyan lines belong to the trajectories of their centers of mass, respectively. False negative tracking, as marked heavy black lines, broke out just after the partial or full occlusion of the persons. False negative became false positive when the 1st person was out of video frames as well as the 3rd person was fully occluded. Regardless of light reflection, occlusion and scale change, 2nd, 4th, 5th, and 6th persons were successfully recognized and tracked over the video sequences.

Table 5.1 makes a detail evaluation of the obtained results and the average measures are satisfactory for many applications of computer vision. Besides scale, orientation, and illumination changes, the approach can handle the occlusion with high accuracy. The framework is likely a bit superior to the similar approaches e.g., [139, 140] in the sense that it can track multiple targets irrespective of sparse crowd, medium density crowd, and dense crowd scenes. Yet, a draw back of this approach is that if the target is confused or misdirected once, then false positive or false negative detection breaks out and it is hard to get the true positive tracking result as the probability to get back the real target is limited.

5.4 Summary and Discussion

We (in [SMD08b]) have studied, a possible extended method of [166], which follows the detection of a target in a video and the target is to be tracked in the following frames using the region covariance method as by [166], can work in some extent as a single covariance matrix extracted from a region of interest can match the region in some else views and poses. We have proposed a better framework ([SDc]) for tracking individual targets in diverse crowded scenes. The noticeable difference with [SMD08b], [SDc], and few directional existing works e.g., [139, 140] are stated below:

- Tracking in [SMD08b] based on spatial information only, whereas the method proposed in [SDc] based on both spatial and temporal information.

- In [SMD08b] region covariance has been used as target descriptor and integral images are used for fast covariance computation. Target region with smallest dissimilarity is selected as the matching region and track that region in the next frame. In [SDc] the region of motion history blob on the original frame has been used as a target descriptor and phase correlation techniques are used to find similarity measure. The highest similarity measure of phase correlation techniques is considered as the best match of the target in the next frame.

- A key advantage behind the approach of [SDc] is that it is not necessary to search the possible target region everywhere on the candidate frame except for the candidate regions. Consequently, the searching process becomes overpowering rapid.

- Approaches of [139, 140] work well with five people scenes whereas approach of [SDc] can track multiple targets irrespective of sparse crowd, medium density crowd, and dense crowd scenes.

- Experimental results reported that the proposed framework of [SDc] is good for individual target tracking. The average precision and accuracy rates are also satisfactory for the applications of computer vision. Nevertheless, a deficiency of this approach is that if the target will be confused or misdirected once, then it will be laborious to win back the real target.

- The framework of [SDc] performs better in sparse and medium crowded scenes as compared to high crowded scenes as the rate of ambiguous appearance resulting from high dense is very high.

Considering the excellence and disadvantage of the framework of [SDc], it is yet highly accurate with respect to the effects of mutations in noise and lighting, which assures high-quality performance on fades, targets moving in and out of the shade, and flashes of light.

Future work would primarily focus to overcome the shortcoming of the framework, i.e., if the target is confused or misdirected once, then it is hard to get back the real target. Future work would also make an adaptation of the approach for tracking individual targets in very high density crowded scenes containing thousands or millions of people, e.g., track persons from the pilgrims circling around Kabba in Mecca. The potential problems of persons tracking in such extremely challenging scene include the small number of pixels on targets, ambiguous appearance resulting from dense, etc.

Chapter 6

Summary and Future Work

Contents

6.1 Summary of the Contributions

Detecting human behaviors efficiently in vast amounts surveillance video, both retrospectively and in realtime, is fundamental technology for a variety of higher-level applications of critical importance to public safety and security. Video surveillance systems have proven to be promising approaches in many security-intensive applications. The focus is to identify real challenges in intelligent surveillance systems and technology and to investigate practical solutions to the core problems of computer vision applications in both theoretical and practical perspectives. The cooperative effort of the computer vision research community, intelligent surveillance systems which process video feeds from real-world scenarios have not yet attained the desirable level of applicability and robustness. This is widely due to the algorithmic assumptions as well as the huge amount of video data analysis.

The objective of this thesis is to accommodate some of the challenges in computer vision, posed by interesting event/behavior detection and individual target tracking in diverse crowded scenes, to a certain degree. We have three contributions in this thesis namely unusual or abnormal event detection, usual or normal event detection, and individual target tracking in crowded scenes.

6.1.1 Unusual Event Detection

For safety and security, abnormal (unusual) event detection is an important task in video surveillance system. But it is a very challenging task as abnormal event is rare and occurs infrequently and very hard to define. Automatic video surveillance is attractive because it promises to replace more costly option of staffing video surveillance monitors with human observers. The scientific challenge is to invent and implement automatic systems to achieve detailed information about the activities and behaviors of people or vehicles observed by sensors (e.g., cameras). A good deal of research works have been carried out in the direction of crowd behavior analysis and detection of abnormal activities. Nevertheless, each work consists of both excellence and disadvantage. We also contributed some same directional works.

Since visual attention allows to focus analysis and processing on some restrained parts of images and frames, it has emerged in recent years as a convincing tool to make robot and computer vision more and more operative in a wide variety of jobs. We started by investigating both static and space-time saliency detection to detect abnormalities in various crowded scenes. We concluded that saliency based models would solely be suitable for sparse crowd scenes (e.g., Fig. 3.2) to detect abnormalities. Consequently, we went further in quest of a good approach and proposed different spatiotemporal information based methods for crowd behavior analysis and detection of abnormal activities in crowd scenes with various densities (e.g., Fig. 1.1).

Spatiotemporal information takes into account motion as an informative feature to detect and segment interesting objects or targets by the help of optical flow computation, block matching or other motion detection methods. Upon analyzing the multifarious spatiotemporal information in miscellaneous ways, we brought forward approaches:

- Covariance Matrix 3.2([SID08a]),
- Normalized Continuous Rank Increase Measure 3.3([SD09a]),
- Mahalanobis Metric 3.4 ([SD09c]),
- Bhattacharyya Metric 3.5([SD10a]),
- Enumerated Entropy 3.6 ([SID08b],[SID10]),
- Shannon Entropy 3.7 ([SDb])

which have been affixed on the existing directional start-of-the-art.

Irrespective of indoor and outdoor video surveillance, region of interest (RoI) makes the video processing faster. Based on applications and type of videos, RoI would extend from few parts of a video frame to the whole frame. In case of applications, e.g., to monitor escalators, linear passages, high-way, etc., video processing region can be fixed by using a mask instead of analyzing the whole video frame. We introduced three types RoI namely *motion heat map* (MHM), *motion map* (MM) or *spatiotemporal region of interest* (ST-RoI), *region of interest image map* (RIIM). Basically, their functions are the same, only difference in construction. The MHM expects long video to produce the hot region on the image indicating the main motion activity, whereas both MM and RIIM expect much less long video to generate a general maneuver of the motion. In general, RoI ameliorates the quality of results and makes the processing time fast a bit more.

6.1.1.1 Covariance Matrix Approach

This approach detects unusual events principally from unidirectional flow of crowd (e.g., escalators). The video frames are labeled *normal* or *abnormal* based on the distance measure between covariance matrices of the distributions of the optical flow vectors computed on consecutive frames. Those flow vectors are the result of tracking a set of features points discovered by the Harris corner detector applied on each frame considering a RoI. The MHM has been constructed to represent RoI. The approach has been tested against a single camera data-set placed in the escalator exits in an airport or so-called *Escalator dataset* [132]. The approach is simple and easy to understand. But it may suffer from initialization problem at the implementation stage.

6.1.1.2 NCRIM Approach

This approach also detects abnormal motion frames from real videos irrespective of both static and dynamic backgrounds. The approach is based on the use of the *spatiotemporal region of interest (ST-RoI) features* obtained from ST-RoI, which is estimated using *motion history image* (MHI). Within ST-RoI, exceptional motion makes the motion vectors (e.g., directions) change significantly as compared to normal motion. The *normalized continuous rank-increase measure* (NCRIM) calculated from the ST-RoI features has been used as the judgement index for determining normal or abnormal motion frame. To demonstrate the interest of the proposed approach, the results based on the detection of abnormal motion frames in real videos obtained from escalator dataset have been presented. The approach is better than that of Covariance matrix due to the fact that it can eliminate few hindrances of Covariance matrix approach, e.g., it can detect the undetected unusual event of Fig. 3.9.

6.1.1.3 Mahalanobis Metric Approach

The approach detects abnormal events mainly in surveillance video systems (e.g., escalators, narrow passages, etc.), based on optical flow analysis of crowd behavior followed by *Mahalanobis* and χ^2 metrics. The video frames are flagged as normal or abnormal based on the statistical classification of the distribution of Mahalanobis distances of the normalized spatiotemporal information of optical flow vectors. Optical flow vectors are computed from the small blocks of the specific region of successive frames namely *region of interest image* (RII), which is discovered by *region of interest image map* (RIIM). The RIIM is obtained from specific treatment of foreground segmentation of moving subjects. The Mahalanobis metric removes several limitations of the Euclidean metric:(i) it automatically accounts for the scaling of the coordinate axes, (ii) it corrects for correlation between the different features, (iii) it can provide curved as well as linear decision boundaries. On the other hand, as breakages it faces the problem e.g., *multicollinearity* or the restriction e.g., the number of samples in the data set has to be larger than the number of variables. Yet, in the proposed approach, both problems have been minimized by dint of 5 variables and tracking about 1500 samples (points of interest) in each frame, respectively.

Each estimated *Mahalanobis distance* ($D_m(i)$) belongs to either *member group* or *non-member group*. Sample with a higher $D_m(i)$ than $\sqrt{3}$ is treated as *non-member group*, otherwise *member group*. The *member group* contains solely the samples of a normal event, whereas *non-member group* contains primarily samples of abnormal events including *outliers*. To make a decision about a video frame whether it goes to usual or unusual, we have only processed the samples of the *non-member group*. The approach has been tested against both *Escalator dataset* [132] and *UMN dataset* [137].

6.1.1.4 Bhattacharyya Metric Approach

This approach lies in the use of the Bhattacharyya distance to measure differences in properties of clusters over time between frames. It estimates sudden changes and abnormal motion variations of a set of interest points detected by Harris detector and tracked by optical flow technique and classified by *K-means*. Estimation of Bhattacharyya distance between classes and thereof the normalized Bhattacharyya distance measure provides the knowledge of the state of abnormality. The Mahalanobis distance is a particular case of the Bhattacharyya distance. Original interpretation of the Bhattacharyya measure has few problems, i.e., it does not impose a metric structure since it violates at least one of the distance metric axioms [67]. The authors in [38] proposed a derivative of the Bhattacharyya measure in the form of $\sqrt{1 - cos\theta}$ which does indeed represent a metric distance between distributions as this distance obeys all of the metric axioms. Instead of using the *proposed measure of* [38], we have considered the Bhattacharyya bound which commonly uses in pattern recognition. The Bhattacharyya distance has been used as a class separability measure for feature selection and is known provide the upper and lower bounds of the Bayes error. On estimating all Bhattacharyya distances among classes, we have estimated their *geometric means* and come together those means to calculate the *log-average* to represent a *single effective distance* (G_β) between two consecutive frames using *Algorithm* 1. The normalized G_β provides the knowledge of the state of abnormal activity in video frames over time. To conduct experiments, we have used the *Escalator dataset* [132] and the *UMN Dataset* [137]. We have concluded that distances between clusters of tracked corners on movers are a reasonable way to characterize abnormal behavior as the distances vary significantly in case of abnormalities.

6.1.1.5 Enumerated Entropy Approach

Entropy estimation is an important problem that arises in statistical pattern recognition, adaptive vector quantization, image registration and indexing, and other areas. The proposed entropy approach makes known abnormal motion frames from real videos based on an defined *Entropy* function. Upon obtaining the spatiotemporal information of each frame, we have analyzed that and got the factors *motion area ratio*, *coefficient of direction variation*, *coefficient of distance variation*, and *direction histogram characteristic*. We have defined the function *Entropy* on the basis of these factors. The approach has been tested on both *Escalator dataset* [132] and *UMN dataset* [137]. Although the approach is suitable to detect a wide variety of abnormalities, it presumes a few limitations, e.g.,:

- First, we have defined the function *Entropy* ourselves, and hence it does not reflect the exact definition of *Entropy* which normally used in *information theory* (so called *Shannon Entropy*). Explicitly, our enumerated *Entropy* is constructed from a single probability rather than from a set of probabilities summing to 1.

- Second, as the mid-level features are extracted based on the result of Harris detector, these features might be sensitive to textures and hence features like direction histogram may be distorted.

6.1.1.6 Shannon Entropy Approach

We have put forward this simple but effective approach (in [SDb]) to detect anomalies in videos based on *Shannon Entropy*, which is estimated on the statistical treatments of the spatiotemporal information of a set of interest points within a region of interest by measuring their *degree of randomness* of both directions and displacements. Entropy is a measure of the disorder/randomness in video frame. On estimating the entropy, we can detect anomalies directly without segmentation or tracking subject singly. It has been showed that degree of randomness of the directions (*circular variance*) changes markedly in abnormal state of affairs and does change only direction variation but does not change with displacement variation of the interest point. Degree of randomness of the displacements has been applied to counterbalance this deficiency. Simple simulations have been exercised to see the characteristics of these crude elements of entropy. Normalized entropy measure provides the knowledge of the state of anomalousness. Experiments have been conducted on various real world video datasets e.g., *Escalator datset* [132], *UMN dataset* [137], *Web dataset* as operated by Mehran et al. [131], etc. Both simulation and experimental results have reported that entropy measures of the frames over time is an outstanding way to characterize aberrations in videos. Results have also reported that the proposed method performs something to a greater degree to distinguish abnormal sequences. The results are likely a bit superior to the work of Mehran et al. [131] in the sense that there is no reported false positives on the proposed method. Table 3.3 provides the quantitative results of a comparison with Mehran et al.′s [131] results for the same four sample videos.

6.1.1.7 Favoring and Disfavoring factors inside Approaches

We introduced six different approaches to detect abnormalities in crowd videos. They are based on the analysis of the spatiotemporal information, hence there are some unique global pros and cons reflect on them. They have been tested on surveillance videos obtained by single camera.

Their key excellences include: (i) simple and easy to understand, (ii) do not need explicit learning process and training data, (iii) omnidirectional, (iv) do not impose limitation on the number of movers in the videos, (v) reduce processing time by deeming region of interest on each video frame, (vi) detect abnormalities directly without segmentation or tracking subject individually, (vii) do not necessitate low level change detection algorithms.

On the other hand, their main shortcomings include: (i) expect a predefined threshold to make a decision, (ii) do not effectively handle occlusion due to the assumption of optical flow method, (iii) do not localize the abnormalities on the video frames.

6.1.2 Usual Event Detection

Detection of usual events in the surveillance video, e.g., TRECVID2008 [43], is an extremely difficult task. The serious challenges include, but are not limited to: (i) a wide variety in the appearance of event types with different view angles, (ii) divergent degrees of imperfect occlusion, (iii) complicated interactions among people, (iv) unilluminated area on the video frame, (v) varying target sizes and poses, (vi) massive population flow, (v) miscellaneous scales, (vi) light reflection, (vii) fluctuation, etc.

In this direction of researches, we lined up our efforts and successfully annexed the following contributions ([SD09b]):

- First, we keyed out a new method which generates automatically *pseudo Euclidian distance* (PED) from the trigonometrically treatment of the *motion history blob* (MHB).
- Second, we proposed a methodology based on PED for the *video event detection* (VED).

To extract image features, optical flow estimation would be a superior grade for the crowd scene but it is too sensitive to the small noise due to the broadness of the camera view. If there are many people in the videos, the existence of small motion noise will be extremely negative and unreliable. Thus, estimating the movement of objects by optical flow is difficult. Deeming this fact we rest with confidence on *motion history image* (MHI), *motion history blob* (MHB), and trigonometric treatments of MHB which generate PED to extract efficient image features. The MHI is a representation of the history of pixel-wise changes, yet it remains a computationally inexpensive method for analysis of object motions and effectively only previous frame needs to be stored. Given a point with its direction of motion where the point coincides the center of a circle. *How far the point can virtually travel inside the circle with that direction?* That virtual distance is called *pseudo Euclidian distance* (PED). Circle is an exceptional ellipse in which the two foci are coincident at the center of the ellipse. The global motion orientation of a *motion history blob* inside of an ellipse has been figured at Fig. 4.2. Fig. 4.3 shows a simple example of PED. The PED, would be potentially used in wide variety of computer vision applications, remains an important contribution on this thesis. There are several state-of-the-art algorithms for tracking *object of interest* (OoI) based on e.g., particle filtering [11], hybrid strategy [31], etc. Since occlusions happen frequently in limited camera scope, particle filtering may archive a commendable performance. But particle filtering is a time-consuming process, especially when the object tracked is large. It is difficult to complete the test on evaluation data within the limited time. Henceforth, we have taken up PED for MHB or OoI tracking with the aim of different kinds of VED. To show the interest of the usage of PED, we proposed a PED based methodology for various video events detection, e.g., *PersonRuns*, *ObjectPut*, *OpposingFlow*, *PeopleMeet*, *Embrace*, and *PeopleSplitUp*. The results based on the detection of several events at *TRECVID2008* in real videos have been demonstrated. Some results show the robustness of the methodology, while the rests give evidence the dimension of the difficulty of the problem at hand. The *TRECVID2008* surveillance event detection task is a big challenge to test the applicability of such methodologies in a real world setting. Big challenges include highly clutter, massive population flow, heavy occlusion, reflection, shadow, fluctuation, varying target sizes, low video quality, etc. Problem also includes the MHB tracking, which heavily depends on direction and position of the *blob*. Hence, *MHB tracker* is inefficient where there needs more information than only direction and position, e.g., if the blob of a person occludes long time or moves long time with occlusion or does not move an elongated period of time, etc. Yet, we hold firmly the view that we have achieved much valuable insights and experience to practical problems and future PED based evaluation of more effective VED methodologies have the potential to come across with better results.

6.1.3 Individual Target Tracking

Individual target tracking is a challenging research topic in computer vision. Obstacles in tracking individual targets can grow due to quick target motion, changing appearance patterns of both the target and the scene, nonrigid target structures, dynamic illumination, inter-target and target-to-scene occlusions, multi-target confusion, etc.

We also employed many of our efforts in this direction of researches and contributed as stated below.

- *First*, we have studied a possible extended method ([SMD08b]), which was originally proposed by [166]. The method is based on spatial information. It follows the detection of a target in a video and the target is to be tracked in the following frames using the *region covariance matric* method as introduced by [166]. The method can work in some extent as a single covariance matrix extracted from a region of interest can match the region in some else views and poses.

- *Second*, we have proposed an approach ([SDc]) based on *temporal-spatial* information suitable for tacking individual targets in parse crowd, medium density crowd, and dense crowd. This approach differs from the *MHB tracking method* (PedVed [SD09b]) by two key directions:

  - We have proposed how to extract the target (or the region of interest over frame in time) and candidate (or the region of possible target over next frame in time) regions from the silhouetted structures of more recently moving pixels of the object of interest by combining two techniques, namely MHI [22] and *Hu's moments* [80]. The MHI, which uses temporal history of position or motion, helps to create an MHB or *silhouetted region of motion component* (SRMC) while Hu's moments find the *center of mass* of the SRMC (*Hu center*). Extraction of target/candidate is similar to the MHB tracking method (PedVed [SD09b]), except after getting the *center of mass* from each SRMC, we consider the original video frame to locate that center. A key advantage behind of this hybrid technique is that it is not necessary to search the possible target region everywhere on the candidate frame except for the candidate regions. Accordingly, searching process grows overpoweringly rapid.
  - We have introduced individual target tracking techniques using distinct maximum peaks, obtained by phase-correlation techniques, from the resulting peaks of target regions and the candidate frame's candidate regions. When two target and/or candidate regions are similar, their phase-correlation function gives a distinct sharp peak (see Fig. 5.3 (a) and (b)). Conversely, the peak of two dissimilar target and/or candidate regions drops significantly

(see Fig. 5.3 (c)). The motivation of the usage of phase-correlation techniques is the fact that unlike many spatial-domain algorithms, the phase-correlation means is resilient to noise, occlusions, and other defects typical of medical or satellite images. There would be more than one candidates which will have almost the same peak heights and cause ambiguity and difficulty in tracking process. To solve the problem, we propose tracking techniques based on the *highest geometric mean*. In the evaluation method, if a *ground truth ellipse* overlaps with the system output rectangle in any frame, we define it to be a *true positive* or *correct detection*. If there is no overlapping system output within the ground truth ellipse, we define it as a *false negative* or *miss detection*. If there is no ground truth but there is a system output, then it is a *false positive* or *false alarm*. If there is no ground truth ellipse as well as there is no system output rectangle, then we define it as a *true negative* or *correct rejection*. Normally, true negative occurs when the target is motionless or the state of being occluded or out of the video frame.

The video sequences of the *PETS 2009 Benchmark data* have been considered for performance evaluation of the approach. The noticeable differences among [SMD08b], [SDc], and [139, 140] are:

- Tracking in [SMD08b] exclusively based on spatial information, whereas the method proposed in [SDc] based on both spatial and temporal information.

- In [SMD08b] region covariance has been used as target descriptor and the integral images are processed for fast covariance computation. Target region with smallest dissimilarity is selected as the matching region and track that region in the next frame. In [SDc] the *silhouetted region of motion component* on the original frame has been used as a target descriptor and phase correlation techniques are used to find similarity measure. The highest similarity measure of phase correlation techniques is considered as the best match of the target in the next frame.

- A key superiority behind the approach of [SDc] is that it is not necessary to search the possible target region everywhere on the candidate frame except for the candidate regions. Consequently, the searching process becomes overpowering rapid.

- The approaches of [139, 140] work well with five people scenes; whereas the approach of [SDc] can work to track multiple targets irrespective of sparse crowd, medium density crowd, and dense crowd scenes.

- Experimental results reported that the proposed framework of [SDc] is good for individual target tracking. The average precision and accuracy rates are also satisfactory for the applications of computer vision. In spite of that, a deficiency of this approach is that if the target will be confused or misdirected once, then it will be laborious to win back the real target.

Experimental results reported that the proposed approach [SDc] is suitable for individual target tracking in diverse crowd scenes. Nevertheless, an imperfection of this approach is that if the target will be confused or misdirected once, then it will be challenging to get back the real target. Deeming the favorable and the unfavorable factors of the approach, it is still highly accurate and its sensitivity to the effects of mutations in noise and lighting, which assures high-quality performance on fades, targets moving in and out of the shade, and flashes of light.

6.2 Conclusion

In this thesis, we have developed algorithms which accommodate some of the challenges encountered in videos of crowded environments to a certain degree. We have developed six approaches namely, Covariance Matrix, NCRIM, Mahalanobis Metric, Bhattacharyya Metric, Enumerated Entropy, and Shannon Entropy. Approaches have been adopted by first performing a global-level motion analysis within each frame's region of interest that provides the knowledge of crowd's multi-modal behaviors in the form of complex spatiotemporal structures. These structures are then employed in the detection of unusual surveillance events within crowds. To conduct experiments, we have heavily relied on three thought-provoking datasets so-called *Escalator dataset* [132], *UMN dataset* [137], and *Web dataset* [131]. The results reflect some unique global excellences and breakages of the approaches. After analyzing motion in an else view, we have keyed out a metric called *pseudo Euclidian distance*. To show its usage, a methodology based on it has been employed in the detection various usual surveillance events from the *TRECVID2008*. Some results report the robustness of the methodology, while the rests give evidence the dimension of the difficulty of the problem at hand. Big challenges include, but are not limited to, massive population flow, heavy occlusion, reflection, shadow, fluctuation, varying target sizes, etc. Yet, we have obtained much useful insights and experience to the practical problems. Furthermore, the thesis explores a single target tracking algorithm within miscellaneous crowded scenes. Video sequences from the *PETS2009 Benchmark data* have been utilized to evaluate its performance. Viewing its pros and cons, the algorithm is still highly accurate and its sensitivity to the effects of diversifications in noise and lighting, which ascertains high-quality performance on disappearances, targets moving in and out of the shadow, and flashes of light.

6.3 Future Directions

6.3.1 Automatic Estimation of Threshold

Automatic threshold estimation is still one of the major challenges in computer vision. All the methods introduced in this thesis use some sort of threshold anyway. All of the thresholds have been defined statically. Future work would make inquiry how to estimate threshold automatically.

6.3.2 Occlusion Handling

Occlusion handling is another major challenge in computer vision. All the approaches in Chapter 3 based on the optical flow method and the analysis of the spatiotemporal information obtained by it. In optical flow technique occlusion handling is not taken into account, because the occluded pixels violate a major assumption of optical flow technique that each pixel goes somewhere. Consequently, occlusion handling has been looked over all of the approaches. Future work would investigate how to minimize the occlusion problem.

6.3.3 Multi-camera Involvement

Advances in sensing technologies as well as the increasing availability of computational power and efficient bandwidth usage methods are favoring the emergence of applications based on distributed systems combining multiple cameras and other sensing modalities. Multiple cameras can provide different viewpoints of a *region of interest*. Since all experiments have been conducted on videos of single fixed camera, it would be interesting to test the approaches with moving *single camera datasets* or *multi-camera datasets*. Future work would take into account the dedication of multiple cameras so that videos, like escalators, could be conclusively broken down into its essential features properly in all parts (e.g., commencement, halfway point, and outlet) of an elongated escalator to proclaim the eccentric event if there will exist any. Consequently, the engagement of multiple cameras would help to analyze many *region of interests* which would be occluded by a single camera.

englishfrenchb

Chapitre 7

Résumé substantiel en français

Contents

7.1 Récapitulatif des contributions

Détecter de manière efficace les comportements humains à partir d'un grand nombre de séquences de vidéosurveillance - tant sur le plan rétrospectif qu'en temps réel - est une technologie fondamentale pour beaucoup d'applications de haut niveau, dont l'importance se révèle cruciale en matière de sûreté et de sécurité publique. Pour de nombreuses applications de sécurité intensive, les systèmes de vidéosurveillance ont été soumis à des approches prometteuses. L'objectif consiste à identifier les vraies problématiques liées aux systèmes et aux technologies de surveillance, et de trouver des solutions concrètes pour résoudre des questions essentielles relatives aux applications de vision assistée par ordinateur selon des perspectives à la fois théoriques et pratiques.

Cette thèse a pour objectif de minimiser ou de surmonter certains problèmes liés à la vision assistée par ordinateur générés par la détection d'événements/comportements intéressants et le suivi d'individus-cibles dans diverses scènes de foule. Cette thèse s'appuie sur les trois contributions suivantes : la détection d'événements inhabituels ou anormaux, la détection d'événements habituels ou normaux et le suivi d'individus-cibles dans des scènes de foule.

7.1.1 Détection d'événements inhabituels

Dans les domaines de la sûreté et de la sécurité, la détection d'événements anormaux (inhabituels) est une tâche cruciale pour les systèmes de vidéosurveillance. Mais c'est également une mission particulièrement ambitieuse puisque ce type d'événement se produit de manière irrégulière. Il est donc difficile d'en donner une définition. La vidéosurveillance automatique est séduisante car elle a pour but de remplacer des observateurs humains par des systèmes intelligents de vidéosurveillance moins coûteux. Le défi scientifique consiste donc à inventer et appliquer des systèmes automatiques permettant de restituer des informations détaillées sur les activités et les comportements d'individus ou de véhicules observés grâce à des capteurs (par exemple des caméras). La plupart des travaux de recherche se sont orientés vers l'analyse des comportements d'une foule et la détection d'activités anormales. Cependant, chaque étude comporte des avantages et des inconvénients. Les travaux auxquels nous avons contribués s'orientent également dans cette direction.

Puisque l'attention visuelle permet d'orienter l'analyse et le traitement vers certaines parties d'une image, un outil convaincant est apparu ces dernières années pour accroître la performance de la robotisation et de la vision assistée par ordinateur dans de nombreux domaines. Nous avons commencé par examiner la saillance/visibilité statique et spatio-temporelle pour détecter des événements anormaux dans une scène de foule. Nous en sommes arrivés à la conclusion que, pour détecter des événements anormaux, les modèles de saillance/visibilité seraient uniquement adaptés aux scènes dans lesquelles la foule est clairsemée (e.g., Fig. 3.2). Nous avons ainsi recherché une approche convenable et proposé plusieurs méthodes basées sur les informations spatio-temporelles pour analyser les comportements d'une foule et détecter des événements anormaux dans des scènes où la foule est plus ou moins dense (e.g., Fig. 1.1).

Les informations spatio-temporelles prennent en compte les données de mouvement pour détecter et segmenter les cibles ou objets d'intérêt grâce au calcul du flux optique, à la correspondance des blocs ou aux autres méthodes de détection du mouvement. Pour analyser des informations spatio-temporelles complexes, nous proposons les approches suivantes :

- La matrice de covariance (3.2 [SID08a]),
- La MNACR (Mesure Normalisée de l'Augmentation Continue du Rang) (3.3 [SD09a]),
- La distance de Mahalanobis (3.4 [SD09c]),
- La distance de Bhattacharyya (3.5 [SD10a]),
- L'entropie énumérée (3.6 [SID08b, SID10]),
- L'entropie de Shannon (3.7 [SDb])

Ces approches ont été annexées aux états de l'art directionnels existants.

Quel que soit l'environnement de vidéosurveillance (en intérieur ou en extérieur), la région d'intérêt (RdI) permet un traitement plus rapide de la vidéo. Selon les applications et les types de vidéos, la RdI peut s'étendre d'une petite partie de l'image à son ensemble. Dans certains cas pratiques (pour contrôler, par exemple, des escalators, des couloirs linéaires, une autoroute etc.), la région de traitement de la vidéo peut être définie en utilisant un masque plutôt qu'en analysant l'ensemble de l'image. Nous avons introduit les trois types de RdI suivants : le MHM (*Motion Heat Map*), le MM (*Motion Map*), et le RIIM (*Region of Interest Image Map*). En fait, si leurs fonctions respectives sont identiques, la seule différence concerne leur implémentation. Le MHM requiert une longue vidéo pour pouvoir produire une région chaude sur l'image indiquant la principale zone d'activité, tandis que le MM et le RIIM nécessitent une séquence vidéo beaucoup plus courte pour générer une manœuvre générale de mouvement. Généralement, les RdI améliorent la qualité des résultats et accélèrent la durée de traitement.

7.1.1.1 La matrice de covariance

Cette approche détecte les événements inhabituels découlant principalement du flot unidirectionnel de la foule (par exemple, des escalators). Les images vidéo sont étiquetées comme *normales* ou *anormales* selon la mesure de distance entre les matrices de covariance relatives à la distribution des vecteurs de flux optique calculés sur plusieurs images consécutives. Ces vecteurs de flux sont le résultat du suivi d'un ensemble de points caractéristiques déterminés par le détecteur de Harris appliqués sur chaque image selon une RdI. Le MHM a été conçu pour représenter les RdI. La Fig. 3.5 est un simple organigramme représentant la structure proposée. Cette approche a été testée à partir d'un ensemble de données émises par une caméra unique placée à la sortie des escalators dans un aéroport (appelé *ensemble de données Escalator* [132]).

L'approche est simple et facile à comprendre. Un problème d'initialisation peut être ressenti pendant l'étape de mise en application.

7.1.1.2 La MNACR

Cette approche permet de détecter également les images anormales à partir de vidéos dont l'arrière-plan est indifféremment statique ou dynamique. Elle s'appuie sur l'utilisation de caractéristiques obtenues à partir d'une *région d'intérêt spatio-temporelle* (RdI-ST), estimée grâce au MHI (*Motion History Image*). Dans les RdI-ST et contrairement à un mouvement normal, un mouvement exceptionnel va modifier les vecteurs de mouvement (la direction, par exemple) de manière significative. La *mesure normalisée de l'augmentation continue du rang* (MNACR) calculée à partir des caractéristiques RdI-ST a été utilisée en tant qu'indice pour déterminer la normalité ou l'anormalité d'une image. Les résultats liés à la détection d'images anormales dans des séquences vidéo réelles, obtenues à partir des données *Escalator* ([132]), ont été présentés pour démontrer l'intérêt de l'approche proposée.

Cette approche, qui permet de détecter des événements anormaux non détectés par la matrice de covariance (e.g., voir Fig. 3.9), est donc plus performante.

7.1.1.3 La distance de Mahalanobis

Comme les approches précédentes, celle-ci permet de déceler des événements anormaux principalement dans les systèmes de vidéosurveillance (par exemple des escalators, des couloirs étroits etc.) en s'appuyant sur l'analyse du flux optique correspondant au comportement de la foule selon les calculs de *Mahalanobis* et χ^2. Les images vidéo sont signalées comme *normales* ou *anormales* selon la classification statistique donnée par la distribution des distances de Mahalanobis, qui portent sur les informations spatio-temporelles normalisées des vecteurs de flux optique. Ces vecteurs sont calculés à partir de petits blocs présents dans une région spécifique formée par des images successives, à savoir le RII (*Region of Interest Image*), mis en évidence par le RIIM (*Region of Interest Image Map*). On obtient le RIIM grâce à un traitement spécial issu de la segmentation au premier plan des sujets en mouvement.

La *distance de Mahalanobis* est une *mesure* qui satisfait les conditions de distance suivantes : (i) la non-négativité, (ii) l'identité des indiscernables, (iii) la symétrie ou la commutativité, et (iv) l'inégalité triangulaire. Cette distance s'appuie sur les corrélations entre les variables à partir desquelles plusieurs modèles peuvent être identifiés et analysés. Cette méthode est utile pour déterminer la similarité d'un ensemble d'échantillons inconnu par rapport à un ensemble connu. Cette mesure diffère de la distance euclidienne puisqu'elle prend en compte les corrélations entre l'ensemble des données et que l'échelle est invariante. En effet, l'échelle des mesures n'est pas prise en compte. De nombreuses statistiques de test sont approximativement distribuées, comme χ^2. La distance au carré de Mahalanobis est distribuée de la même manière qu'une distribution χ^2 avec un degré de liberté égal au nombre de variables indépendantes de l'analyse. Cependant, la distribution χ^2 ne comporte qu'un seul paramètre appelé *degré de liberté*. La courbe de distribution χ^2 a une forme oblique générée par de très faibles degrés de liberté, et elle change radicalement lorsque les degrés de liberté augmentent. Enfin, lorsque les degrés de liberté sont élevés, la courbe de distribution χ^2 semble normale. Comme pour toutes les autres courbes de distribution continue, la surface totale sous une courbe de distribution χ^2 est égale à 1.0. La règle des trois sigmas indique que dans le cas d'une distribution normale, environ 68%, 95% et 97,7% des valeurs comportent des déviations de moyenne standard d'une valeur de 1, 2 et 3. De toute évidence, presque toutes les valeurs atteignent le niveau 3. Par conséquent, les échantillons dont la distance au carré de Mahalanobis est supérieure à 3 ont une probabilité inférieure à 0,01. Ces échantillons sont classés comme appartenant à un groupe non-membre. Les échantillons dont la distance au carré de Mahalanobis est inférieure à 3 sont classés dans un groupe membre. La détermination du seuil dépend de l'application et des types d'échantillons. Dans l'approche que nous proposons, chaque distance de Mahalanobis estimée ($D_m(i)$) appartient soit à un *groupe membre* soit à un *groupe non-membre*. L'échantillon dont la $D_m(i)$ est supérieure à $\sqrt{3}$ est considéré comme appartenant à un *groupe non-membre*. Sinon, il appartient à un *groupe membre*. Un *groupe membre* comporte exclusivement des échantillons relatifs à un événement normal, tandis qu'un *groupe non-membre* contient essentiellement des échantillons relatifs à un événement anormal, notamment les aberrations (*outliers*). Pour s'assurer qu'une image vidéo est normale ou anormale, nous traitons simplement les échantillons du groupe non-membre. L'utilisation de la distance de Mahalanobis fait disparaître les limitations liées à la distance euclidienne, par exemple :

- Elle indique systématiquement l'échelle de l'axe des coordonnées ;
- Elle améliore la corrélation entre les différentes caractéristiques ;
- Elle produit des limites décisionnelles à la fois courbées et linéaires.

Néanmoins, ces avantages ne sont pas sans conséquence. Le calcul de la *matrice de corrélation* peut soulever quelques problèmes. Lorsque les données étudiées sont mesurées sur un grand nombre de variables, elles peuvent contrôler beaucoup d'informations redondantes ou corrélées. C'est cette *multicolinéarité* des données qui aboutit à une matrice de corrélations unique ne pouvant être inversée. Par ailleurs, le nombre d'échantillons présents dans l'ensemble de données doit être supérieur au nombre de variables. Cependant, dans notre approche, ces deux problèmes ont été minimisés grâce, respectivement, à 5 variables et au suivi de 1500 échantillons (*points d'intérêt*) dans chaque image. Enfin, cette approche a été testée à partir de l'ensemble des données *Escalator* ([132]) et *UMN* ([137]).

7.1.1.4 La distance de Bhattacharyya

L'un des problèmes majeurs en vision assistée par ordinateur consiste à calculer la différence entre les distributions des caractéristiques, notamment de couleurs et de textures [150]. Notre étude s'intéresse principalement à la distance de Bhattacharyya et à ses dérivés. La statistique χ^2 est utilisée pour apporter une mesure de similarité entre deux distributions ou histogrammes [110]. Cette mesure estime la statistique χ^2. En transformant toutes les variances pour les rendre constantes, la mesure de Batthacharyya élude le problème de singularité de la statistique χ^2 au moment de comparer les classes vides (*empty bins*) des histogrammes [2]. Après avoir comparé la distance de Bhattacharyya et la divergence de Kullback-Leibler, l'auteur de [95] a remarqué que, pour certains points, la distance de Bhattacharyya aboutit à de meilleurs résultats tandis que pour d'autres les résultats sont équivalents. De nombreuses mesures (Bhattacharyya, euclidienne, Kullback-Leibler, Fisher) ont été examinées pour la différentiation d'image, et il en a été conclu que la distance de Battacharyya est le différentiateur le plus efficace [18]. Les mesures de différentiation, qui s'appuient sur des estimations empiriques de la distribution des caractéristiques, ont été développées pour la classification [138], la récupération d'images [145, 151], la segmentation non-surveillée [77], la détection des contours [153], le suivi d'objets [38], etc. Des études de référence préliminaires ont confirmé que les mesures de différentiation basées sur la distribution présentent des résultats excellents dans les domaines suivants : récupération d'images [145], segmentation non-surveillée des textures [77], conjonction avec un classificateur *k-plus proche voisin* (*k–nearest-neighbor*), et reconnaissance d'objets basée sur la couleur et la texture [163, 138].

Dans notre approche, nous estimons les changements soudains et les variations anormales de mouvement au sein d'un ensemble de points d'intérêt identifiés par le détecteur de Harris, suivis par la technique du flux optique et le classement par l'algorithme *K-means*. La distance de Mahalanobis est un cas particulier de la distance de Bhattacharyya, dont l'interprétation initiale suscite quelques problèmes. Ainsi, elle n'impose aucune structure métrique car elle transgresse au moins un des axiomes de mesure de distance [67]. Les auteurs de [38] ont proposé une méthode dérivée de la mesure de Bhattacharyya sous la forme $\sqrt{1-cos\theta}$ (avec θ l'angle formé par les deux vecteurs de positons) qui représente en effet la distance entre les distributions, puisque cette distance obéit à tous les axiomes métriques. Au lieu de recourir à la mesure proposée par [38], nous prenons en compte la borne de Bhattacharyya communément utilisée pour la reconnaissance de modèle. La distance de Bhattacharyya a été utilisée comme mesure de séparabilité de classe dans la sélection des caractéristiques, et est connue pour indiquer les bornes supérieures et inférieures de l'erreur de Bayes. Pour calculer toutes les distances de Bhattacharyya, on procède au calcul des *moyennes géométriques* (*geometric means*) des distances de Bhattacharyya dans les classes et l'on regroupe ces moyennes pour calculer le *log-average* et représenter la distance effective unique (G_β) entre deux images consécutives en utilisant l'algorithme 1. La distance normalisée G_β fournit les informations relatives à l'état d'une activité anormale dans des images vidéo à travers le temps. Pour mener nos expérimentations, nous avons utilisé l'ensemble de données *Escalator* ([132]) et *UMN* ([137]). Nous en avons déduit que les distances entre les amas des coins suivis dans les zones en mouvement (*movers*) représentent une méthode convenable permettant de définir un comportement anormal puisque les distances varient de manière significative en cas d'anomalie.

7.1.1.5 L'entropie énumérée

Cette approche permet de déceler des images anormales dans des vidéos en temps réel. Elle se déroule en plusieurs étapes :

1. Extraction du MHM (*Motion Heat Map*). La carte de chaleur représente les intensités de mouvement. Les zones chaudes correspondent par exemple aux mouvements de grande intensité, tandis que les zones froides représentent les mouvements de faible intensité, etc.
2. Extraction des points d'intérêt de Harris dans les régions chaudes de la scène. Dans les cas les plus simples, cette étape est appliquée dans des zones bien délimitées. On considère un *MHM binaire* (*binary MHM*), blanc (mouvement) et noir (absence de mouvement). Les points d'intérêt

sont appliqués dans les régions blanches, et les tâches sont extraites.

3. Calcul des flux optiques sur les points d'intérêt dont les limites sont définies par les régions chaudes de la scène.
4. Calcul des caractéristiques de niveau intermédiaire comme la densité, le coefficient de variation directionnelle et l'histogramme directionnel.
5. A partir du calcul des caractéristiques de niveau intermédiaire tel qu'indiqué dans l'étape précédente, définition des caractéristiques de haut niveau (comme l'entropie) définissant la normalité ou l'anormalité des événements et renvoyant plusieurs types d'anormalités.

Les étapes 1 à 4 sont génériques et ne dépendent pas d'un domaine d'application particulier. Elles concernent l'extraction des caractéristiques de bas niveau. La cinquième étape dépend du domaine d'application et requiert un processus d'apprentissage spécifique. L'organigramme de nos travaux est indiqué en Fig. 3.24 et s'articule autour d'une structure comportant les trois niveaux de caractéristiques suivants :

- Bas niveau (*Low-level*) : il s'applique aux calculs directement extraits du signal (données visuelles) comme les points d'intérêt, les régions d'intérêt, les tâches (*blobs*), les contours, les stries (*ridges*), le flux optique, etc. Nous utilisons un mélange gaussien pour détecter les premiers plans dans les zones où la foule est clairsemée (faible densité), et les flux optiques dans les zones où la foule est importante (forte densité).
- Niveau intermédiaire (*Mid-level*) : il concerne les caractéristiques générées après un processus d'apprentissage directement à partir des caractéristiques de bas niveau, et contribue à rehausser les caractéristiques de haut niveau (sémantiques) comme la densité de la foule (proportion de tâches dans la scène), la trajectoire, la vitesse, la direction, l'accélération etc. Les caractéristiques de niveau intermédiaire sont calculées à partir des caractéristiques de bas niveau (par exemple les régions d'intérêt, les points d'intérêt etc.) et sont classées en structures.
- Haut niveau (*High-level*) : il s'applique aux caractéristiques comportant plus de sémantiques qu'au niveau intermédiaire et en nombre suffisant pour prendre des décisions. Nous sommes ici en présence d'événements normaux/anormaux. Nous avons développé une fonction appelée *entropie* qui utilise les caractéristiques de niveau intermédiaire telles que le MAR (*Motion Area Ratio*), le coefficient de variation directionnelle, le coefficient de variation de la distance, et les caractéristiques de l'histogramme directionnel. Pour savoir si une image (*frame*) est normale ou anormale, nous observons son entropie minimale.

Cette approche a été testée à partir des ensembles de données *Escalator* et *UMN*. Bien qu'elle permette de détecter un grand nombre d'anomalies, elle est sujette aux restrictions suivantes :

- En premier lieu, nous avons nous-mêmes défini la fonction d'entropie, qui par conséquent ne reflète pas la définition normalement utilisée en théorie de l'information (aussi appelée *entropie de Shannon*). De manière plus explicite, notre entropie énumérée est conçue à partir d'une unique probabilité et non d'un ensemble de probabilités dont la somme est égale à 1.
- De plus, puisque les caractéristiques de niveau intermédiaire sont extraites selon les résultats de la détection des coins de Harris, ces caractéristiques peuvent être sensibles à la texture. Des caractéristiques comme celle de l'histogramme directionnel pourraient être déformées dans une telle situation.

7.1.1.6 L'entropie de Shannon

Estimer l'entropie est un problème crucial qui se pose lors de la reconnaissance de modèle statistique, de la quantification vectorielle, de l'indexation et de l'enregistrement d'images, etc. L'estimation non-paramétrique de l'entropie de Shannon a suscité l'intérêt dans de nombreux domaines comme la statistique non-paramétrique, la reconnaissance de modèle, l'identification de modèle, l'enregistrement d'image etc. [3, 168, 49, 90, 74, 13, 171].

Nous proposons une approche simple mais efficace (dans [SDb]) permettant de détecter des anomalies dans des vidéos en utilisant l'entropie de Shannon. Le calcul du *degré du caractère aléatoire* (*degree of randomness*) permet d'estimer l'entropie de Shannon lors du traitement statistique des informations spatio-temporelles relatives à un ensemble de points d'intérêt dans une région d'intérêt. L'entropie est une mesure calculant le caractère désordonné/aléatoire d'une image vidéo. Il a été démontré que dans les situations anormales, le degré du caractère aléatoire des directions (variance circulaire) change de manière notable et que seule la variation de la direction change, ce qui n'est pas le cas lorsqu'il y a variation du déplacement du point d'intérêt. Le degré du caractère aléatoire des déplacements a été appliqué afin de compenser cette faiblesse. Des simulations simples ont été effectuées pour reconnaître les caractéristiques de ces éléments rudimentaires relatifs à l'entropie. Le calcul de l'entropie normalisée fournit des informations sur l'état d'anormalité. La Fig. 3.27 expose l'approche proposée.

Notre premier objectif consiste à présenter une méthode holistique, indépendante de la segmentation ou du suivi d'un individu, pour détecter les anomalies en calculant le degré de désordre/chaos contenu dans des vidéos. Nos travaux et ceux de [8, 7, 83] indiquent qu'en cas d'urgence, le modèle de flux optique comporte suffisamment de perturbations. De même, dans les travaux de [86], les auteurs ont utilisé quelques termes s'approchant des nôtres tels que *vitesse* et *accélération*. D'une certaine manière, notre approche pourrait être considérée comme une prolongation de ces travaux. Toutefois, nous avons pris une direction différente en utilisant deux mesures statistiques appelées *degré du caractère aléatoire des directions* (variance circulaire) et *degré du caractère aléatoire des déplacements* (coefficient de variation des déplacements) des points d'intérêt. Ces éléments sont à la fois essentiels et rudimentaires pour définir notre calcul de l'entropie, c'est-à-dire le calcul du caractère désordonné et aléatoire d'une image vidéo. Plus le désordre/chaos est présent dans une image, plus il y a d'entropie, c'est-à-dire que les images anormales ont une entropie plus élevées que les images normales. Nous avons démontré que la variance circulaire - l'un des deux principes essentiels de l'entropie - change de manière significative dans des circonstances anormales, et qu'elle change non pas lorsque le vecteur de longueur du point d'intérêt varie mais uniquement lorsqu'il y a variation directionnelle du point d'intérêt. Pour compenser ce défaut lié à la variance circulaire, nous avons utilisé une quantité normalisée et adimensionnelle appelée *degré du caractère aléatoire des déplacements* - un autre principe essentiel de l'entropie - qui consiste en un calcul statistique du rapport entre *déviation standard* et *moyenne*. De plus, nous avons précisé qu'une anomalie concerne également les déplacements. Par conséquent, le degré du caractère aléatoire des déplacements - un facteur suffisant dans le calcul de l'entropie - joue un rôle important dans la détection de certains types d'aberrations dans les vidéos. Lors de l'estimation de l'entropie, on peut directement détecter des anomalies sans procéder à la segmentation ou au suivi d'un individu. Par ailleurs, ce système fonctionne tout autant avec des scènes comportant de hautes densités de mouvement qu'avec des scènes de faible densité. Il y a également d'autres avantages : (i) il détecte tous les événements d'une vidéo comportant des variations d'entropie importantes par comparaison aux précédents événements ; (ii) il autorise tous les flux directionnels de mouvement sans restriction de nombre ; (iii) il ne nécessite pas de lourd processus ni de données d'apprentissage mais un seuil doit être défini au préalable.

Nous avons mené nos expériences en conditions réelles à partir de divers ensembles de données *Escalator*, *UMN*, *Web* ([131]), etc. Les résultats des simulations et des expériences indiquent que le calcul de l'entropie des images à travers le temps est une manière efficace de déterminer les aberrations présentes dans une vidéo. Les résultats montrent que la méthode proposée fonctionne à un niveau avancé pour détecter des anomalies dans une séquence, et qu'elle est un peu plus performante que celle présentée dans les travaux de Mehran et al. [131], puisqu'en effet aucun faux positif n'a été reporté dans la méthode proposée. Le tableau 3.3 présente les résultats quantitatifs obtenus après comparaison avec les résultats de Mehran et al. [131] à partir de quatre échantillons vidéo identiques.

7.1.1.7 Avantages et inconvénients de ces approches

Nous avons présenté six approches différentes permettant de détecter les anomalies présentes dans des vidéos impliquant des scènes de foule. Elles s'appuient toutes sur l'analyse des informations spatio-temporelles, ce qui implique des avantages et inconvénients particuliers sur le plan général. Toutes ces méthodes ont été testées à partir de séquences de vidéosurveillance produites par une seule caméra.

Avantages - Les principaux atouts des systèmes proposés sont les suivants :

- Pour la plupart, ces méthodes sont simples, facilement compréhensibles et applicables ;
- Elles ne requièrent pas de processus ni de données d'apprentissage explicites ;
- Elles supportent tous les flux directionnels de mouvement sans restriction de nombre dans les vidéos ;
- La durée de traitement est réduite grâce à la prise en compte d'une région d'intérêt ;
- Les anomalies sont directement détectées sans recourir à la segmentation ou au suivi individuel.

Inconvénients - Les trois principaux inconvénients des approches proposées sont les suivants :

- Pour qu'une décision soit prise, ces approches nécessitent qu'un seuil soit préalablement défini ;
- Elles sont basées sur le concept de flux optique. Celui-ci ne prend pas en compte la gestion des occlusions, incompatibles avec l'un des principes de base du flux optique. Par conséquent, la gestion des occlusions est ignorée.
- Elles détectent les aberrations mais ne les localisent pas dans la vidéo.

7.1.2 Détection d'événements habituels

Dans des environnements très fréquentés comme les aéroports, les centres commerciaux, les gares, les parkings, les centres-villes etc., il est très fréquent que les objets se confondent et s'obstruent les uns les autres. Par conséquent, les méthodes conventionnelles de soustraction de l'arrière-plan ne fonctionnent pas de manière très efficace. De nombreux algorithmes de détection d'image, s'appuyant sur des cascades de transfert [170, 114] et de reconnaissance [41, 157, 20] ont montré des résultats très prometteurs en matière de détection de piétons dans un environnement réel très fréquenté comportant des occlusions. Pour détecter les piétons, un histogramme de gradients a été utilisé dans [41], tandis que les auteurs de [157, 20] ont eu recours à un modèle inspiré de la biologie pour reconnaître différentes catégories, dont les piétons. Cependant, la plupart de ces algorithmes de détection de piétons est particulièrement lente pour être utilisée par des applications fonctionnant en temps réel. Ainsi, les auteurs de [184] ont remarqué que les algorithmes complexes permettant de détecter les piétons, ont besoin d'environ 0,5 secondes pour reconnaître une image de dimension 128x64 dans [41], de deux secondes par image dans [157] et de 80 secondes par image dans [20]. Un algorithme de détection et de suivi d'une cible, s'appuyant sur les calculs réalisés par un capteur audio stéréo et de vision cyclopique, a été présenté dans [192]. Pour détecter les événements de surveillance issus de TRECVID2008 [43], de nombreux algorithmes ont été proposés, basés notamment sur les éléments suivants : la détection des changements [188], l'analyse de la trajectoire [109], la connaissance de la trajectoire et du domaine [55], les cubes vidéo spatio-temporels [53], la méthode de Haar pour détecter les piétons et la concordance d'histogrammes [184], les concepts de flux optique [61, 97, 58], etc. Dans les travaux présentés dans [109], l'événement *rencontre de personnes* (*PeopleMeet*) a été essentiellement détecté grâce à l'analyse de la trajectoire des piétons. Les personnes ont été détectées et pistées dans la scène grâce à la méthode décrite dans [81]. Pour obtenir des trajectoires piétons fiables dans le cadre de la détection d'un événement où des personnes se rencontrent, les auteurs ont proposé une méthode d'association hiérarchique basée sur la détection, méthode qui permet de suivre de manière efficace plusieurs piétons dans des conditions difficiles. Leur méthode a permis de déterminer les trajectoires en associant progressivement les réponses envoyées par le détecteur de piétons présenté dans [176]. La combinaison de la trajectoire et des sous-systèmes de connaissance du domaine est présentée dans [55]. Un sous-système *trajectoire* applique la technique de détection humaine et de suivi pour générer la trajectoire. Trois niveaux de caractéristiques sont utilisés pour la détection : *PersonsRuns*, *PeopleMeet*, *PeopleSplitUp* et *Embrace*. Le sous-système *domaine* élabore des modèles spécifiques pour *PeopleMeet*, *OpposingFlow* et *ElevatorNoEntry*, selon la connaissance du domaine. Néanmoins, le grand nombre de possibilités liées à un événement observé sous différents angles, à différentes échelles, à différents degrés d'occlusion partielle etc., complique le travail des détecteurs d'événement. Par conséquent, il est nécessaire d'en améliorer l'efficacité en procédant à des recherches plus poussées.

C'est dans cette direction que nous avons orienté nos efforts et ajouté avec succès les contributions suivantes (4 [SD09b]) :

- Nous avons d'abord adapté une nouvelle méthode permettant de générer automatiquement la distance pseudo-euclidienne (*Pseudo Euclidean Distance* - PED) à partir du traitement trigonométrique relatif au ciblage de l'historique des blobs dans l'image (*Motion History Blob* - MHB).
- Puis nous avons proposé une méthodologie basée sur le PED pour la détection d'un événement vidéo (*Video Event Detection* - VED).

Pour procéder à l'extraction des caractéristiques d'une image, l'estimation du flux optique pourrait être un outil performant dans le cas de scènes de foule. Cependant, il est trop sensible au bruit, même faible, car l'angle de vue de la caméra est trop large. Lorsqu'une vidéo implique beaucoup de monde, la présence d'un faible bruit dans une image sera extrêmement préjudiciable et peu fiable. Par conséquent, il est difficile d'estimer le mouvement des objets par le biais du flux optique. Ceci dit, nous nous appuyons avec confiance sur le MHI (*Motion History Image*), le MHB (*Motion History Blob*) et le traitement trigonométrique des MHB qui génère des PED pour extraire efficacement les caractéristiques d'une image. Le MHI est une représentation de l'historique des changements rencontrés par les pixels. Cette méthode de calcul est peu coûteuse pour analyser les mouvements d'un objet, et il est vrai que seules les images précédentes doivent être mémorisées. Prenons un point et la direction de son mouvement. Ce point coïncide avec le centre d'un cercle. Quelle distance ce point peut-il virtuellement parcourir à l'intérieur du cercle dans cette même direction ? La distance virtuelle est appelée *distance pseudo-euclidienne* (Pseudo Eclidian Distance - PED). Le cercle a la forme d'une ellipse exceptionnelle dans laquelle deux foyers coïncident avec le centre de cette ellipse. L'orientation générale de l'image d'un MHB dans une ellipse est indiquée dans la Fig. 4.2.

Un exemple basique de calcul du PED est indiqué dans la Fig. 4.3. Le PED, qui pourrait être utilisé dans un grand nombre d'applications de vision assistée par ordinateur, occupe une part importante de cette thèse. Il existe plusieurs algorithmes performants permettant de suivre des *objets d'intérêt* (OdI) à partir, par exemple, du filtrage de particules (*particle filtering*) [11], d'une stratégie hybride [31], etc. Dans la mesure où des occlusions se produisent de manière fréquente lorsque le champ de la caméra est restreint, le filtrage de particules pourrait donner des résultats remarquables. Mais le filtrage est un processus qui requiert beaucoup de temps, particulièrement lorsque les dimensions de l'objet à suivre sont grandes. Il est difficile de tester des données d'évaluation sur un temps limité. Par conséquent, nous avons opté pour le suivi PED plutôt que pour le MHB ou le PdI afin d'obtenir plusieurs types de VED. Afin de démontrer l'intérêt du PED, nous avons proposé une méthodologie pour détecter plusieurs types d'événements dans une vidéo, tels que PersonRuns, ObjectPut, OpposingFlow, PeopleMeet, Embrace et PeopleSplitUp. Les résultats relatifs à la détection de plusieurs événements dans des vidéos réelles issues de TRECVID2008 ont été démontrés. Certains d'entre eux montrent la robustesse de cette méthode, d'autres la difficulté à résoudre ce problème. La détection d'un événement vidéo issu de TRECVID2008 [43] est particulièrement complexe pour tester l'applicabilité de telles méthodologies en conditions réelles. Cette complexité découle des éléments suivants : un encombrement important, un flux massif de personnes, une occlusion très élevée, la réflexion, les ombres, les fluctuations, le changement de taille des cibles, la mauvaise qualité de la vidéo etc. D'autres problèmes portent également sur le suivi MHB, fortement dépendant de la direction et de la position des blobs. Au final, le pisteur MHB est inefficace dans les cas où la direction et la position ne sont pas des informations suffisantes (par exemple, lorsque qu'un blob représentant une personne est soumise à une occlusion pendant une longue durée, qu'il se déplace longtemps avec l'occlusion, ou encore qu'il ne bouge pas pendant un certain temps, etc.).

Néanmoins, nous sommes persuadés que nos idées et nos expériences ont été bénéfiques pour résoudre des problèmes pratiques relatifs à l'évaluation future des PED en faveur de méthodologies VED plus efficaces et performantes.

7.1.3 Suivi d'une cible individuelle

Le suivi d'une cible individuelle est un sujet de recherche complexe dans le domaine de la vision assistée par ordinateur. En effet, les obstacles rencontrés sont toujours plus nombreux lorsqu'il s'agit de suivre des cibles individuelles. Diverses raisons peuvent intervenir à savoir le déplacement rapide de la cible, des changements intervenants dans les modèles d'apparence tant de la cible que de la scène, la structure non-rigide de la cible, l'éclairage dynamique, les occlusions entre les cibles ou entre la scène et la cible, la confusion entre plusieurs cibles etc. Une bonne sélection des caractéristiques joue un rôle fondamental dans le suivi. Cette étape est en effet étroitement liée à la représentation de la cible. Une cible peut être représentée par une forme géométrique primitive comme un rectangle, une ellipse etc. Un certain nombre d'algorithmes ont été utilisés pour le suivi de cible. Ces algorithmes diffèrent d'abord dans le sens où ils ont recours aux caractéristiques d'une image et au modèle de mouvement, ainsi qu'à l'apparence et à la forme de la cible [185]. Lors du suivi de silhouette, par exemple, la silhouette est pistée à la fois par le biais de la concordance de formes [96] et par l'évolution des contours [17]. Le suivi d'un corps en conditions réelles et le système d'animation d'un humanoïde sont illustrés dans [36]. Dans le suivi du kernel, la cible peut être suivie en calculant le mouvement du kernel dans des images consécutives [6, 149]. Le kernel peut être de forme elliptique et associé à un histogramme [38] suivi d'une procédure *mean-shift* permettant de localiser la cible ou un modèle rectangulaire associé à une matrice de covariance [139] donnant lieu par la suite à une recherche de force brute pour localiser la cible. Les auteurs de [166] ont présenté le concept de matrice de région de covariance pour détecter un objet et localiser la cible. La caractéristique de concordance de ce concept consiste à effectuer une recherche simple du plus proche voisin sous la mesure de distance, une tâche accomplie rapidement en utilisant des images intégrales. Les matrices de région de covariance peuvent être utilisées pour détecter une cible dans une vidéo, cette cible pouvant être suivie dans les images suivantes grâce à l'approche que ces auteurs ont proposée. En s'inspirant de ce concept, les auteurs de [139, 140] ont proposé d'autres approches pour détecter, classer et suivre plusieurs cibles. Les cibles sont représentées par des matrices de région de covariance, et des filtres particulaires effectuent le suivi. Leurs approches sont intéressantes mais ne peuvent pas fonctionner pour suivre des cibles séparément lorsque la foule se disperse, ou que la densité de la foule est moyenne ou dense. En effet, les résultats indiquent seulement des scènes comportant 5 personnes. Par conséquent, il est nécessaire de développer un algorithme capable de prendre en considération ces types de scènes pour ensuite suivre des cibles individuelles.

Nous avons également orienté nos efforts dans cette direction et apporté notre contribution de la manière suivante (5 [SMD08b, SDc]) :

En premier lieu, nous avons étudié la possibilité de développer notre méthode ([SMD08b]), initialement proposée par [166]. Cette méthode s'appuie sur les informations d'ordre spatial. Elle consiste à pister la détection d'une cible dans une vidéo, la cible étant suivie dans les images suivantes grâce à l'utilisation de la matrice de région de covariance telle que présentée dans [166]. Dans une certaine mesure, une telle méthode peut fonctionner en tant que simple matrice de covariance extraite d'une région d'intérêt, région qui peut concorder avec d'autres vues et d'autres poses.

Par ailleurs, nous avons proposé une approche ([SDc]) basée sur les informations spatio-temporelles adaptées au suivi de cibles individuelles dans une foule dispérsée, de densité moyenne ou de haute densité. Cette approche diffère de la méthode de suivi MHB (PiedVed [SD09b]) de par les deux directions-clés suivantes :

- Nous avons proposé une méthode pour extraire une cible (ou une région d'intérêt sur une image dans le temps) ou des régions candidates (régions avec des cibles possibles sur l'image suivante) à partir de structures indiquant les contours des pixels ayant les plus récemment été soumis à un mouvement dans un objet d'intérêt. Cette méthode combine deux techniques, à savoir le MHI [22] et le Mouvement de Hu (*Hu's moments*) [80]. Le MHI a recours à l'historique temporel d'une position ou d'une image, et permet de créer un MHB ou un SRMC (Silhouetted Region of Motion Component), tandis que l'approche de Hu consiste à trouver le centre de masse du SRMC. L'extraction de la cible/du candidat est identique à la méthode de suivi MHB (PedVed [SD09b]) à ceci près qu'après avoir obtenu le centre de masse de chaque SRMC, nous prenons en considération l'image vidéo originale pour localiser le centre. L'un des principaux avantages de cette technique hybride est qu'il est inutile de rechercher la région-cible dans toute l'image candidate, sauf dans les cas de région candidate. Par conséquent, le processus de recherche devient beaucoup plus rapide.

- Nous avons présenté des techniques pour pister des cibles individuelles en utilisant les pics maximum distincts obtenus via des techniques de corrélation de phases à partir des pics résultants des régions-cibles et des régions des images candidates. Lorsque deux cibles et/ou régions candidates sont identiques, la fonction de corrélation de phases fait apparaître un brusque pic distinct (voir Fig. 5.3 (a) et (b)). Inversement, le pic correspondant à deux cibles et/ou régions candidates diminue considérablement lorsque celles-ci sont différentes (voir Fig. 5.3 (c)). L'utilisation de techniques de corrélation de phases est motivée par le fait que contrairement à nombre d'algorithmes *espace-domaine*, la corrélation de phases résiste au bruit, aux occlusions et à d'autres défauts typiques des images satellites ou médicales. Sachant que plus d'un candidat rencontrera des hauteurs de pics identiques, créant ainsi des ambiguïtés et des difficultés dans le processus de suivi, nous proposons des techniques basées sur la plus haute moyenne géométrique (*highest geometric mean*) pour résoudre ce problème.

Notre approche comporte les étapes algorihtmiques suivantes :

- Estimation du premier plan pour obtenir le SRMC ;
- Segmentation du SRMC pour obtenir une séquence SRMC ;
- Estimation du centre de masse, de la gravité ou du centroïde pour obtenir le centre de chaque SRMC ;
- Estimation des cibles ou des régions candidates par le biais des coordonnées du centroïde ;
- Application des techniques de corrélation de phases pour obtenir les plus grandes hauteurs de pics des cibles et/ou des candidats ;
- Traitement des hauteurs de pics les plus grandes via l'algorithme qui suit la cible grâce à la plus haute moyenne géométrique. Si la plus haute moyenne géométrique est supérieure à la limite dynamique connue, alors le suivi est normalement effectué. Sinon, la cible est immobile, en état d'occlusion ou en dehors de l'image.

Selon la méthode d'évaluation, si une ellipse de réalité terrain chevauche le rectangle correspondant au résultat du système, nous la qualifions de *vrai positif* (*true positive*) ou *détection correcte*. S'il n'y a pas de chevauchement, nous la qualifions de *faux négatif* (*false negative*) ou *détection manquée* (*miss detection*) . S'il n'y a pas d'ellipse de réalité terrain mais qu'il y a un rectangle correspondant au résultat du système, il s'agit alors de *faux positif* (*false positive*) ou *fausse alarme* (*false alarm*). S'il n'y a ni ellipse ni rectangle, on parle de *vrai négatif* (*true negative*) ou *rejet correct*. Normalement, un vrai négatif se produit lorsque la cible est immobile, en état d'occlusion ou en dehors de l'image. Les séquences vidéo issues de la base PETS 2009 Benchmark [42] ont été prises en compte pour procéder à l'évaluation de cette approche. Les différences les plus manifestes parmi [SMD08b], [SDc], et [139, 140] sont les suivantes :

- Dans [SMD08b], le suivi est effectué à partir d'informations d'ordre spatial uniquement, tandis que dans [SDc] il se fait à partir d'informations spatio-temporelles.
- Dans [SMD08b], la région de covariance a été utilisée en tant que descripteur de cible, et des images intégrales sont utilisées pour un calcul rapide de la covariance. La région-cible ayant la dissemblance la plus faible est sélectionnée comme la région de concordance, puis cette région est pistée dans l'image suivante. Dans la région [SDc] correspondant à l'historique de l'image, le blob fait office de descripteur de cible, et des techniques de corrélation de phases sont utilisées pour trouver une mesure de similarité. La mesure ayant la similarité la plus élevée après application des techniques de corrélation de phases est considérée comme la meilleure correspondance pour la cible dans l'image suivante.
- L'un des principaux avantages de l'approche [SDc] est qu'il est inutile de rechercher la région cible possible dans toute l'image candidate, à l'exception des régions candidates. Le processus de recherche est ainsi considérablement accéléré.
- L'approche proposée dans [139, 140] fonctionne bien lorsque la scène est limitée à 5 personnes. En revanche, l'approche présentée dans [SDc] permet de suivre plusieurs cibles quelle que soit la densité de la foule (dispersement, moyennement dense, très dense).
- Les résultats expérimentaux ont montré que le cadre proposé dans [SDc] est adapté au suivi de cibles individuelles. Les degrés moyens de précision et d'exactitude sont également satisfaisants pour des applications de vision assistée par ordinateur. Cependant, cette approche souffre de la faiblesse suivante : lorsque la cible est floue ou mal orientée, il est difficile de la retrouver.

Les résultats expérimentaux ont montré que le cadre proposé dans [SDc] est adapté au suivi de cibles individuelles. Si l'on considère les avantages et les inconvénients de cette approche, elle fait toujours preuve d'une grande exactitude et bénéficie d'une bonne sensibilité aux changements de bruit, de luminosité (permettant ainsi d'obtenir de très bons résultats lorsque la lumière faiblit), aux cibles entrant et sortant des zones d'ombres, et aux lueurs soudaines.

7.2 Orientations futures

7.2.1 Estimation automatique du seuil

L'estimation automatique du seuil est encore l'un des principaux défis à relever en matière de vision assistée par ordinateur. Toutes les méthodes présentées dans cette thèse ont recours, d'une manière ou d'une autre, au seuil. Tous les seuils ont été définis manuellement. Des travaux complémentaires devraient s'orienter vers une méthode permettant d'estimer ce seuil automatiquement.

7.2.2 Gestion des occlusions

La gestion des occlusions est un autre défi. Toutes les approches présentées dans le chapitre 3 s'appuient sur la méthode du flux optique et sur l'analyse des informations spatio-temporelles obtenues. La méthode du flux optique ne prend pas en compte la gestion des occlusions car les pixels soumis à l'occlusion sont incompatibles avec le principe de base selon lequel chaque pixel a une position précise. Par conséquent, toutes ces approches ont ignoré la gestion des occlusions. Des travaux complémentaires devraient s'orienter vers une méthode permettant de minimiser les problèmes d'occlusion.

7.2.3 Utilisation de plusieurs caméras

Les progrès accomplis dans le domaine des technologies de perception, ainsi que la disponibilité des méthodes relatives à la puissance informatique et à l'efficacité de la bande passante favorisent l'émergence d'applications basées sur des systèmes de distribution combinant plusieurs caméras et d'autres modes de perception. La présence de plusieurs caméras permet d'obtenir des angles de vue différents pour une région d'intérêt. Dans la mesure où toutes les expériences ont été conduites via l'installation d'une seule caméra, il serait intéressant de tester ces approches à partir d'un ensemble de données issues d'une seule caméra mobile ou de plusieurs caméras. Ainsi, dans une scène d'escalators par exemple, les caractéristiques essentielles de chaque partie de l'escalator (l'entrée, le milieu, la sortie) seraient décomposées de manière concluante afin de déterminer l'événement anormal qui se produit le cas échéant. Par conséquent, l'implication de plusieurs caméras permettrait d'analyser de nombreuses régions d'intérêt alors qu'actuellement, l'utilisation d'une seule caméra engendre des occlusions.

Publications

[ASID08] Mahmoudi Sidi Ahmed, Md. Haidar Sharif, Nacim Ihaddadene, and Chabane Djeraba. Detection of abnormal motions in video. In *Proceedings of the International Workshop on Multimodal Interactions Analysis of Users in a Controlled Environment (MIAUCE), October 20-22, Chania, Crete, Greece*, pages 1–4, 2008.

[ISD08] Nacim Ihaddadene, Md. Haidar Sharif, and Chabane Djeraba. Crowd behaviour monitoring. In *Proceedings of the ACM International Conference on Multimedia (ACM MM), October 26-31, Vancouver, British Columbia, Canada*, pages 1013–1014, 2008.

[SABD10] Md. Haidar Sharif, Husam Alustwani, Ioan Marius Bilasco, and Chabane Djeraba. Détection des mouvements anormaux dans des vidéos. In *Proceedings of the Extraction et Gestion des Connaissances (EGC), January 26-29, Hammamet, Tunisia*, pages 699–700, 2010.

[SBSH08] Md. H. Sharif, A. Basermann, C. Seidel, and A. Hunger. High-performance computing of $1/\sqrt{x_i}$ and $exp(\pm x_i)$ for a vector of inputs x_i on Alpha and IA-64 CPUs. *Journal of Systems Architecture - Embedded Systems Design* [1], 54(7) :638–650, 2008.

[SDa] Md. Haidar Sharif and Chabane Djeraba. Bhattacharyya metric for crowd behavior supervision. *Will be submitted in July* (Graphical Models : Journal special issue selected papers at CompIMAGE2010).

[SDb] Md. Haidar Sharif and Chabane Djeraba. An entropy approach for abnormal activities detection in videos. *Under Review* (CVIU-10-142) : Computer Vision and Image Understanding.

[SDc] Md. Haidar Sharif and Chabane Djeraba. Individual target tracking in crowded scenes. *Under Review* (TIST-2010-04-0089) : ACM Transactions on Intelligent Multimedia Systems and Technology.

[SD09a] Md. Haidar Sharif and Chabane Djeraba. Exceptional motion frames detection by means of spatiotemporal region of interest features. In *Proceedings of the International Conference on Image Processing (ICIP), 7-10 November, Cairo, Egypt*, pages 981–984, 2009.

[1] The work is beyond this Dissertation and it has been included here as there is no special curriculum vitae added.

[SD09b] Md. Haidar Sharif and Chabane Djeraba. PedVed : Pseudo euclidian distances for video events detection. In *Proceedings of the International Symposium on Visual Computing (ISVC), Part II, November 30 - December 2, Las Vegas, NV, USA*, volume 5876 of *Lecture Notes in Computer Science*, pages 674–685. Springer, 2009.

[SD09c] Md. Haidar Sharif and Chabane Djeraba. A simple method for eccentric event espial using Mahalanobis metric. In *Proceedings of the Iberoamerican Conference on Pattern Recognition (CIARP), November 15-18, Guadalajara, Jalisco, Mexico*, volume 5856 of *Lecture Notes in Computer Science*, pages 417–424. Springer, 2009.

[SD10a] Md. Haidar Sharif and Chabane Djeraba. Crowd behavior surveillance using Bhattacharyya distance metric. In *Proceedings of the International Symposium on Computational Modeling of Objects Presented in Images : Fundamentals, Methods, and Applications (CompIMAGE), May 5-7, Buffalo, NY, USA*, volume 6026 of *Lecture Notes in Computer Science*, pages 311–323. Springer, 2010.

[SD10b] Md. Haidar Sharif and Chabane Djeraba. Target tracking in crowded scenes. In *Proceedings of the International Symposium on Computing in Science & Engineering (ISCSE), June 3-5, Tusan Beach Resort, Kusadasi, Turkey*, pages 1–6, 2010.

[SID08a] Md. Haidar Sharif, Nacim Ihaddadene, and Chabane Djeraba. Covariance matrices for crowd behaviour monitoring on the escalator exits. In *Proceedings of the International Symposium on Visual Computing (ISVC), Part II, December 1-3, Las Vegas, NV, USA*, volume 5359 of *Lecture Notes in Computer Science*, pages 470–481. Springer, 2008.

[SID08b] Md. Haidar Sharif, Nacim Ihaddadene, and Chabane Djeraba. Crowd behaviour monitoring on the escalator exits. In *Proceedings of the International Conference on Computer and Information Technology (ICCIT), December 25-27, Khulna, Bangladesh*, pages 194–200, 2008.

[SID10] Md. Haidar Sharif, Nacim Ihaddadene, and Chabane Djeraba. Finding and indexing of eccentric events in video emanates. *Journal of Multimedia*, 5(1) :22–35, 2010.

[SMD08a] Md. Haidar Sharif, Jean Martinet, and Chabane Djeraba. Motion saliency applied to video surveillance. In 2nd Edition B. Furth, editor, *Encyclopedia of Multimedia*, pages 442–444. Springer-Verlag, Oct 2008. ISBN 978-0-387-74724-8.

[SMD08b] Md. Haidar Sharif, Jean Martinet, and Chabane Djeraba. Object tracking in video using covariance matrices. In 2nd Edition B. Furth, editor, *Encyclopedia of Multimedia*, pages 677–680. Springer-Verlag, Oct 2008. ISBN 978-0-387-74724-8.

Bibliographies

[1] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3) :555–560, 2008.

[2] F. Aherne, N. Thacker, and P. Rockett. The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 34(4) :363–368, 1998.

[3] I. A. Ahmad and P. E. Lin. A nonparametric estimation of the entropy for absolutely continuous distributions. *IEEE Transactions on Information Theory*, 22(3) :372–375, 1976.

[4] L. E. Aik and Z. Zainuddin. Curve analysis for real-time crowd estimation system. *European Journal of Scientific Research*, 38(3) :441–453, 2009.

[5] S. Ali and M. Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. pages 1–6, 2007.

[6] S. Ali and M. Shah. Floor fields for tracking in high density crowd scenes. In *European Conference on Computer Vision (ECCV)*, pages 1–14, 2008.

[7] E. L. Andrade, S. Blunsden, and R. B. Fisher. Hidden markov models for optical flow analysis in crowds. In *International Conference on Pattern Recognition (ICPR)*, pages 460–463, 2006.

[8] E. L. Andrade, S. Blunsden, and R. B. Fisher. Modelling crowd scenes for event detection. In *International Conference on Pattern Recognition (ICPR)*, pages 175–178, 2006.

[9] P. Antonakaki, D. Kosmopoulos, and S. J. Perantonis. Detecting abnormal human behaviour using multiple cameras. *Signal Processing*, 89 :1723–1738, 2009.

[10] G. Antonini and J. P. Thiran. Counting pedestrians in video sequences using trajectory clustering. *Transactions on Circuits and Systems for Video Technology*, 16(8) :1008–1020, 2006.

[11] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2) :174–188, 2002.

[12] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1) :43–77, 1994.

[13] J. Beirlant, E. J. Dudewicz, L. Gyorfi, and E. C. Meulen. Nonparametric entropy estimation : an overview. *International Journal of the Mathematical Statistics Sciences*, 6(1) :17–39, 1997.

[14] Y. Benabbas, N. Ihaddadene, and C. Djeraba. Global analysis of motion vectors for event detection in crowd scenes. In *Workshop of IEEE Computer Society Conference on Computer Vision*, pages 109–116, 2009.

[15] Y. Benabbas, A. Lablack, N. Ihaddadene, and C. Djeraba. Action recognition using direction models of motion. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2010.

[16] Y. Benezeth, P. Jodoin, V. Saligrama, and C. Rosenberger. Abnormal events detection based on spatio-temporal co-occurences. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2458–2465, 2009.

[17] M. Bertalmio, G. Sapiro, and G. Randall. Morphing active contours. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7) :733–737, 2000.

[18] A. Bhalerao and N. Rajpoot. Selecting discriminant subbands for texture classification. In *British Machine Vision Conference (BMVC)*, 2003.

[19] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35 :99–109, 1943.

[20] S. Bileschi and L. Wolf. Image representations beyond histograms of gradients : The role of gestalt descriptors. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[21] M. J. Black and A. D. Jepson. Eigentracking : Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1) :63–84, 1998.

[22] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3) :257–267, 2001.

[23] B. A. Boghossian and S. A. Velastin. Motion-based machine vision techniques for the management of large crowds. In *International Conference on Electronics, Circuits and Systems (ICECS)*, pages 961–964, 1999.

[24] O. Boiman and M. Irani. Detecting irregularities in images and in video. *International Journal of Computer Vision*, 74(1) :17–31, 2007.

[25] S. Bouchafa, D. Aubert, L. Beheim, and A. Sadji. Automatic counterflow detection in subway corridors by image processing. *Journal of Intelligent Transportation Systems*, 6(2) :97–123, 2001.

[26] J. Y. Bouguet. Pyramidal implementation of the lucas kanade feature traker. In *A part of OpenCV Documentation, Intel Corporation, Microprocessor Research Labs*, 2000.

[27] G. Brostow and R. Cipolla. Unsupervised bayesian detetcion of independent motion in crowds. In *Computer Vision and Pattern Recognition (CVPR)*, pages 594–601, 2006.

[28] R. Brunelli and T. Poggio. Face recognition : Features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10) :1042–1052, 1993.

[29] A. Burton and J. Radford. *Thinking in Perspective : Critical Essays in the Study of Thought Processes*. Routledge. ISBN 0416858406, 1978.

[30] Y. Cai, N. D. Freitas, and J. J. Little. Robust visual tracking of multiple targets. In *European Conference on Computer Vision (ECCV)*, pages 107–118, 2006.

[31] A. Cavallaro, O. Steiger, and T. Ebrahimi. Tracking video objects in cluttered background. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(4) :575–584, 2005.

[32] M. T. Chan, A. Hoogs, J. Schmiederer, and M. Petersen. Detecting rare events in video using semantic primitives with hmm. In *International Conference on Pattern Recognition (ICPR)*, pages 150–154, 2004.

[33] C. F. Chi, T. C. Chang, and C. L. Tsou. In-depth investigation of escalator riding accidents in heavy capacity MRT stations. *Accident Analysis & prevention*, 38(4) :662–670, 2006.

[34] S. Y. Cho, T. W. S. Chow, and C. T. Leung. A neural-based crowd estimation by hybrid global learning algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 29(4) :535–541, 1999.

[35] T. Coianiz, M. Boninsegna, and B. Caprile. A fuzzy classifier for visual crowding estimates. In *International Conference on Neural Networks (ICNN)*, pages 1174–1178, 1996.

[36] C. Colombo, A. D. Bimbo, and A. Valli. A real-time full body tracking and humanoid animation system. *Parallel Computing*, 34(12) :718–726, 2008.

[37] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition (CVPR)*, page 142–149, 2000.

[38] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5) :564–577, 2003.

[39] US Consumer Product Safety Commission. *CPSC Document #5111 : Escalator safety*. US Consumer Product Safety Commission, Washington, DC, 2003.

[40] F. Cravino, M. Dellucca, and A. Tesei. Dekf system for crowding estimation by a multiple-model approach. *Electronics Letters*, 30(5) :390–391, 1994.

[41] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.

[42] Benchmark data of PETS. *PETS 2009 Benchmark Data*. available from http ://www.cvg.rdg.ac.uk/PETS2009/a.html, 2009.

[43] Benchmark data of TRECVID. *Surveillance Event Detection Pilot*. available from http ://www-nlpir.nist.gov/projects/trecvid/, 2008.

[44] A. C. Davies, J. H. Yin, and S. A. Velastin. Crowd monitoring using image processing. *Electronics and Communication Engineering Journal*, 7(1) :37–47, 1995.

[45] J. Davis and G. Bradski. Real-time motion template gradients using intel cvlib. In *IEEE ICCV Workshop on Framerate Vision*, 1999.

[46] J. W. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *Computer Vision and Pattern Recognition (CVPR)*, page 928–934, 1997.

[47] H. Dee and D. Hogg. Detecting inexplicable behaviour. In *British Machine Vision Conference (BMVC)*, pages 477–486, 2004.

[48] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VSPETS)*, pages 65–72, 2005.

[49] D. L. Donoho. One-sided inference about functionals of a density. *Annals of Statistics*, 16(4) :1390–1420, 1988.

[50] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2nd Edition, 2001.

[51] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *International Conference on Computer Vision (ICCV)*, pages 726–733, 2003.

[52] E. B. Ermis, V. Saligrama, P. M. Jodoin, and J. Konrad. Motion segmentation and abnormal behavior detection via behavior clustering. In *International Conference on Image Processing (ICIP)*, pages 769–772, 2008.

[53] A. Hauptmann et al. Informedia @ trecvid2008 : Exploring new frontiers. In *CMU at TRECVID*, 2008.

[54] B. Scholkopf et al. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7) :1443–1471, 2001.

[55] J. Guo et al. Trecvid 2008 event detection by MCG-ICT-CAS. In *MCG-ICT-CAS at TRECVID*, 2008.

[56] L. Gorelick et al. Shape representation and classification using the poisson equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12) :1991–2005, 2006.

[57] M. Betk et al. Tracking large variable numbers of objects in clutter. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[58] O. B. Orhan et al. University of Central Florida at trecvid 2008 : Content based copy detection and surveillance event detection. In *UCF at TRECVID*, 2008.

[59] Q. C. Pham et al. Real-time posture analysis in a crowd using thermal imaging. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[60] R. Hamid et al. Unsupervised activity discovery and characterization from event-streams. In *Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 251–258, 2005.

[61] S. Hao et al. Tokyo Tech at trecvid. In *Tokyo Institute of Technology at TRECVID*, 2008.

[62] W. Hu et al. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9) :1450 –1464, 2006.

[63] X. Li et al. Robust visual tracking based on incremental tensor subspace learning. In *International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[64] X. Li et al. Visual tracking via incremental log-euclidean riemannian subspace learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[65] W. Foerstner and B. Moonen. *A metric for covariance matrices*. Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University, 1999.

[66] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1) :119–139, 1997.

[67] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Inc., 2nd Edition, 1990.

[68] M. Bierlaire G. Antonini, S. V. Martinez and J. P. Thiran. Behavioral priors for detection and tracking of pedestrians in video sequences. *International Journal of Computer Vision*, 69(2) :159–180, 2006.

[69] D. Gao and N. Vasconcelos. Discriminant saliency for visual recognition from cluttered scenes. In *Neural Information Processing Systems (NIPS)*, 2004.

[70] G. Gennari and G. D. Hager. Probabilistic data association methods in visual tracking of groups. In *Computer Vision and Pattern Recognition (CVPR)*, pages 876–881, 2004.

[71] G. Golub and C. V. Loan. *Matrix Computations*. 3rd edition, Johns Hopkins University Press, 1996.

[72] L. Gorelick, M. Blank, M. Irani, and R. Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12) :2247–2253, 2007.

[73] E. W. Grafarend. *Genauigkeitsmasse geodaetischer Netze*. DGK, Bayerische Akademie der Wissenshaften, Muenchen, http ://www.bayenishcer.uni.de, 1972.

[74] P. Hall and S. C. Morton. On the estimation of entropy. *Annals of the Institute of Statistical Mathematics*, 45(1) :69–88, 1993.

[75] R. Hammond and R. Mohr. Mixtire densities for video objects recognition. In *International Conference on Pattern Recognition (ICPR)*, pages 71–75, 2000.

[76] C. Harris and M.J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–152, 1988.

[77] T. Hofmann, J. Puzicha, and J. Buhmann. Unsupervised textured image segmentation in a deterministic annealing framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8) :803–818, 1998.

[78] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *International Conference on Computer Vision (ICCV)*, pages 84–93, 2001.

[79] B. K. P. Horn. Robot vision. In *MIT Press, Cambridge*, 1986.

[80] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on information Theory*, IT-8(2) :179–187, 1962.

[81] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *European Conference on Computer Vision (ECCV)*, pages 788–801, 2008.

[82] Y. Huang and I. Essa. Tracking multiple objects through occlusions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1051–1058, 2005.

[83] N. Ihaddadene and C. Djeraba. Real-time crowd motion analysis. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008.

[84] L. Itti and C. Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10-12) :1489–1506, 2000.

[85] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11) :1254–1259, 1998.

[86] I. Ivanov, F. Dufaux, T. M. Ha, and T. Ebrahimi. Towards generic detection of unusual events in video surveillance. In *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 61–66, 2009.

[87] H. Wang J. C. Niebles and L. F. Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3) :299–318, 2008.

[88] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4) :620–630, 1957.

[89] F. Jiang, Y. Wu, and A. K. Katsaggelos. Abnormal event detection from surveillance video by dynamic hierarchical clustering. In *International Conference on Image Processing (ICIP)*, pages 145–148, 2007.

[90] H. Joe. On the estimation of entropy and other functionals of a multivariate density. *Annals of the Institute of Statistical Mathematics*, 41(4) :683–697, 1989.

[91] Neil Johnson and David Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14 :583–592, 1996.

[92] I. N. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. In *International Conference on Pattern Recognition (ICPR)*, volume 2, pages 716–719, 2004.

[93] T. Kadir. *Scale, Saliency and Scene Description.* PhD Thesis, University of Oxford, 2002.

[94] T. Kadir and M. Brady. Scale, saliency and image description. *Vision Research*, 45(2) :83–105, 2001.

[95] T. Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, 15(1) :52–60, 1967.

[96] J. Kang, I. Cohen, and G. Medioni. Object reacquisition using geometric invariant appearance model. In *International Conference on Pattern Recognition (ICPR)*, pages 759–762, 2004.

[97] Y. Kawai, M. Takahashi, M. Sano, and M. Fujii. NHK STRL at trecvid 2008 : High-level feature extraction and surveillance event detection. In *NHK STRL at TRECVID*, 2008.

[98] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *International Conference on Computer Vision (ICCV)*, pages 166–173, 2005.

[99] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.

[100] Z. Khan, T. R. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision (ECCV)*, pages 279–290, 2004.

[101] J. Kim and K. Grauman. Observe locally, infer globally : a space-time mrf for detecting abnormal activities with incremental updates. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2928, 2009.

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

[103] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. American Mathematical Society. MR0620955. ISBN 0-8218-5001-6, 1980.

[104] U. Knauer, T. Dammeier, and B. Meffert. The structure of road traffic scenes as revealed by unsupervised analysis of the time averaged optical flow. In *17th International Conference on the Applications of Computer Science and Mathematics in Architecture and Civil Engineering*, 2006.

[105] T. Kobayashi and N. Otsu. Action and simultaneous multiple-person indentification using cubic higher-order local auto-correlation. In *International Conference on Pattern Recognition (ICPR)*, pages 741–744, 2004.

[106] C. Koch and S. Ullman. Shifts in selective visual attention : towards the underlying neural circuitry. *Human Neurobiology*, 4(4) :219–227, 1985.

[107] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. In *International Conference on Cybernetics and Society*, pages 163–165, 1975.

[108] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(3) :107–123, 2005.

[109] S. C. Lee, C. Huang, and R. Nevatia. Definition, detection, and evaluation of meeting events in airport surveillance videos. In *USC at TRECVID*, 2008.

[110] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1) :29–44, 2001.

[111] K. Li, M. Chen, and T. Kanade. Cell population tracking and lineage construction with spatiotemporal context. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 295–302, 2007.

[112] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian. Foreground object detection from videos containing complex background. In *ACM International Conference on Multimedia (ACM MM)*, pages 2–10, 2003.

[113] G. Liang, K. K. Lee, and Y. Xu. Multi-resolution crowd density estimation based on texture analysis and learning from demonstration. *International Journal of Information Acquisition*, 4(1) :1–14, 2007.

[114] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *International Conference on Image Processing (ICIP)*, pages 900–903, 2007.

[115] E. Limpert, W. A. Stahel, and M. Abbt. Log-normal distributions across the sciences : Keys and clues. *BioScience*, 51(5) :341–352, 2001.

[116] S. F. Lin, J. Y. Chen, and H. X. Chao. Estimation of number of people in crowded scenes using perspective transformation. *IEEE Transactions on Systems, Man, and Cybernetics*, 31(6) :645–654, 2001.

[117] W. Lin and Y. Liu. Tracking dynamic near-regular textures under occlusion and rapid movements. In *European Conference on Computer Vision (ECCV)*, pages 44–55, 2006.

[118] W. Lin and Y. Liu. A lattice-based mrf model for dynamic near-regular texture tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5) :777–792, 2007.

[119] T. Lindeberg, A. Akbarzadeh, and I. Laptev. Galilean-diagonalized spatio-temporal interest operators. In *International Conference on Pattern Recognition (ICPR)*, page 57–62, 2004.

[120] S. P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2) :129–136, 1982.

[121] B. P. L. Lo and S. A. Velastin. Automatic congestion detection system for underground platforms. In *International Symposium on Intelligent Multimedia, Video and Speech Processing (ISIMP)*, pages 158–161, 2001.

[122] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2) :91–110, 2004.

[123] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[124] Y. Ma and M. Li. Detection for abnormal event based on trajectory analysis and fsvm. In *International Conference on Intelligent Computing (ICIC)*, pages 1112–1120, 2007.

[125] J. MacCormick and A. Blake. A probabilistic exclusion principle for tracking multiple objects. *International Journal of Computer Vision*, 39(1) :57–71, 2000.

[126] P. C. Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1) :49–55, 1936.

[127] D. Makris and T. J. Ellis. Path detection in video surveillance. *Image and Vision Computing Journal*, 20 :895–903, 2002.

[128] A. N. Marana, L. F. Costa, R. A. Lotufo, and S. A. Velastin. On the efficacy of texture analysis for crowd monitoring. In *International Symposium on Computer Graphics (SIBGRAPI)*, pages 354–361, 1998.

[129] A. N. Marana, L. F. Costa, R. A. Lotufo, and S. A. Velastin. Estimating crowd density with minkowski fractal dimension. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3521–3524, 1999.

[130] R. Maree, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *Computer Vision and Pattern Recognition (CVPR)*, page 34–40, 2005.

[131] R. Mehran, A. Oyama, and M. Shah. Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition (CVPR)*, pages 935–942, 2009.

[132] Project MIAUCE. *This publicly unavailable dataset belongs to the MIAUCE project, EU Research Programme (IST-2005-5-033715).* http ://www.miauce.org/, 2008.

[133] H. P. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. In *Technical Report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University* & *doct. diss., Stanf. University*, 1980.

[134] T. Nanri and N. Otsu. Unsupervised abnormality detection in video surveillance. In *IAPR Conference on Machine Vision Applications (IAPR MVA)*, pages 574–577, 2005.

[135] V. Navalpakkam and L. Itti. Modeling the influence of task on attention. *Vision Research*, 45(2) :205–231, 2005.

[136] V. Navalpakkam and L. Itti. An integrated model of top-down and bottom-up attention for optimal object detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2049–2056, 2006.

[137] University of Minnesota. *Unusual crowd activity dataset of University of Minnesota.* available from http ://mha.cs.umn.edu/movies/crowdactivity-all.avi, 2009.

[138] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based feature distributions. *Pattern Recognition*, 29(1) :51–59, 1996.

[139] H. Palaio and J. Batista. Multi-object tracking using an adaptive transition model particle filter with region covariance data association. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4, 2008.

[140] H. Palaio and J. Batista. A kernel particle filter multi-object tracking using gabor-based region covariance matrices. In *International Conference on Image Processing (ICIP)*, pages 4085–4088, 2009.

[141] N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1034–1040, 2001.

[142] D. Parkhurst, K. Law, and E. Niebur. Modeling the role of salience in the allocation of overt visual attention. *Vision Research*, 42(1) :107–123, 2002.

[143] F. Porikli. Integral histogram : A fast way to extract histograms in cartesian spaces. In *Computer Vision and Pattern Recognition (CVPR)*, pages 829–836, 2005.

[144] F. Porikli and T. Kocak. Robust license plate detection using covariance descriptor in a neural network framework. In *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2006.

[145] J. Puzicha, T. Hofmann, and J. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *Computer Vision and Pattern Recognition (CVPR)*, pages 267–272, 1997.

[146] V. Rabaud and S. Belongie. Counting crowded moving objects. In *Computer Vision and Pattern Recognition (CVPR)*, pages 705–711, 2006.

[147] D. Ramanan and D. A. Forsyth. Automatic annotation of everyday movements. In *Neural Information Processing Systems (NIPS)*, 2003.

[148] D. Ramanan, D. A. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1) :65–81, 2007.

[149] M. Rodriguez, S. Ali, and T. Kanade. Tracking in unstructured crowded scenes. In *International Conference on Computer Vision (ICCV)*, 2009.

[150] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Computer Vision and Image Understanding*, 84(1) :25–43, 2001.

[151] Y. Rubner, C. Tomasi, and L. J. Guibas. A metric for distributions with applications to image databases. In *International Conference on Computer Vision (ICCV)*, pages 59–66, 1998.

[152] C. D. T. Runge. Üeber empirische funktionen und die interpolation zwischen äqui-distanten ordinaten. *Zeitschrift für Mathematik und Physik*, 46 :224–243, 1901.

[153] M. Ruzon and C. Tomasi. Color edge detection with the compass operator. In *Computer Vision and Pattern Recognition (CVPR)*, pages 160–166, 1999.

[154] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2) :151–172, 2000.

[155] A. J. Schofield, P. A. Mehta, and T. J. Stonham. A system for counting people in video images using neural networks to identify the background scene. *Pattern Recognition*, 29(8) :1421–1428, 1996.

[156] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions : A local svm approach. In *International Conference on Pattern Recognition (ICPR)*, pages 32–36, 2004.

[157] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Computer Vision and Pattern Recognition (CVPR)*, pages 994–1000, 2005.

[158] L. Shan and M.C. Lee. Fast visual tracking using motion saliency in video. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1073–1076, 2007.

[159] E. Shechtman and M. Irani. Space-time behavior based correlation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 405–412, 2005.

[160] J. Shi and C. Tomasi. Good features to track. In *Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, 1994.

[161] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition (CVPR)*, pages 23–25, 1999.

[162] C. Stauffer and W. L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :747–757, 2000.

[163] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1) :11–32, 1991.

[164] A. Torralba. Modeling global scene factors in attention. *Journal of the Optical Society of America*, 20(7) :1407–1418, 2003.

[165] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition (CVPR)*, pages 586–591, 1991.

[166] O. Tuzel, F. Porikli, and P. Meer. Region covariance : A fast descriptor for detection and classification. In *European Conference on Computer Vision (ECCV)*, page 589–600, 2006.

[167] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[168] O. Vasicek. A test for normality based on sample entropy. *Journal of the Royal Statistical Society : Series B (Methodological)*, 38(1) :54–59, 1976.

[169] A. Verri and T.A. Poggio. Against quantitative optical flow. In *International Conference on Computer Vision (ICCV)*, pages 171–180, 1987.

[170] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition (CVPR)*, page 511–518, 2001.

[171] P. A. Viola and W. M. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2) :137–154, 1997.

[172] D. Walther. *Interactions of Visual Attention and Object Recognition : Computational Modeling, Algorithms, and Psychophysics*. PhD Thesis, Cal. Inst. of Tech., California, 2006.

[173] D. H. Warren and E. R. Strelow. *Electronic Spatial Sensing for the Blind : Contributions from Perception*. Springer. ISBN 9024726891, 1985.

[174] W. Weibull. A statistical distribution function of wide applicability. *Transactions of the ASME : Journal of Applied Mechanics*, 18(3) :293–297, 1951.

[175] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2) :249–257, 2006.

[176] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2) :247–266, 2007.

[177] T. Xiang and S. Gong. Video behavior profiling and abnormality detection without manual labeling. In *International Conference on Computer Vision (ICCV)*, pages 1238–1245, 2005.

[178] T. Xiang and S. Gong. Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5) :893–908, 2008.

[179] L. Xiaohua, S. Lansun, and L. Huanqin. Estimation of crowd density based on wavelet and support vector machine. *Transactions of the Institute of Measurement and Control*, 28(3) :299–308, 2006.

[180] X. Xue and al. Fudan University at trecvid 2008. In *Surveillance Event Detection Pilot. http ://www-nlpir.nist.gov/projects/trecvid/*, 2008.

[181] M. Sheikh Y. Sheikh and M. Shah. Exploring the space of a human action. In *International Conference on Computer Vision (ICCV)*, pages 144–149, 2005.

[182] T. Yu Y. Wu and G. Hua. Tracking appearances with occlusions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 789–795, 2003.

[183] M. Yang, J. Yuan, and Y. Wu. Spatial selection for attentional visual tracking. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[184] P. Yarlagadda, M. Demirkus, K. Garg, and S. Guler. Intuvision event detection system for trecvid 2008. In *Intuvision at TRECVID*, 2008.

[185] A. Yilmaz, O. Javed, and M. Shah. Object tracking : A survey. *ACM Journal of Computing Surveys*, 38(4) :1–45, 2006.

[186] A. Yilmaz, X. Li, and M. Shah. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11) :1531–1536, 2004.

[187] A. Yilmaz and M. Shah. Actions sketch : a novel action representation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 984–989, 2005.

[188] K. Yokoi, H. Nakai, and T. Sato. Surveillance event detection task. In *Toshiba at TRECVID*, 2008.

[189] D. Zhang, D. G. Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *Computer Vision and Pattern Recognition (CVPR)*, page 611–618, 2005.

[190] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *Computer Vision and Pattern Recognition (CVPR)*, pages 459–466, 2003.

[191] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Computer Vision and Pattern Recognition (CVPR)*, pages 819–826, 2004.

[192] H. Zhou, M. Taj, and A. Cavallaro. Target detection and tracking with heterogeneous sensors. *IEEE Journal on Selected Topics in Signal Processing*, 2(4) :503–513, 2008.

[193] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *International Conference on Pattern Recognition (ICPR)*, pages 28–31, 2004.