Contextual Analysis of Videos Capturing Multiple Moving Targets

by

MYO THIDA

Faculty of Science, Engineering and Computing
School of Computing and Information Systems
Kingston University, London

A dissertation submitted in partial fulfilment of the requirements
of Kingston University
for the degree of Doctor of Philosophy.

May 2013

Director of Study: Professor Paolo Remagnino

Author:
MYO THIDA

Supervisors:
Dr. Eng How-lung - Dr. Dorothy N. Monekosso - Professor Paolo Remagnino (Directory of Study)

Abstract

Over the last two decades, computer vision researchers have been working to improve the accuracy and robustness of algorithms for the context analysis of videos capturing single or multiple moving targets. However, devising algorithms that can work in uncontrolled environments with variable and unfavourable lighting conditions is still a major challenge. This thesis aims to develop robust methodologies to analyse scenes with multiple moving targets captured by a stationary camera.

First, a new particle swarm optimisation algorithm is proposed to incorporate social interaction among targets. A set of interactive swarms is employed to track multiple pedestrians in a crowd. The proposed method improves the standard particle swarm optimisation algorithm with a dynamic social model that enhances the interaction among swarms. In addition, constraints provided by temporal continuity and strength of person detections are incorporated in the tracking process. This allows the particle swarm optimisation algorithm to track multiple moving targets in a complex scene.

Second, a novel method is proposed to detect global unusual events and accurately localise abnormal regions in the monitored scene. The idea is to exploit temporal coherence between video frames and use the manifold learning algorithm, in particular Laplacian Eigenmaps, to discover different crowd activities from a video. The proposed method provides an advantage of visualising and identifying different crowd events in a low dimensional space and detect abnormality. Then, this method is further extended to detect localised abnormality where the behaviour of an indi-

vidual deviates from the rest of the crowd. In this approach, the visual contexts of multiple local patches are studied to model the regular behaviour of a crowded scene. This local probabilistic model allows to detect abnormal behaviour in both local and global context and localise the regions where abnormal behaviour occurs.

The performance of the proposed algorithms is validated using standard data-sets and surveillance videos captured in uncontrolled environments.

Declaration

I hereby declare that the work presented in this thesis describes my own work and it has not been submitted, either in the same or different form, to any other university for an award. Some parts of the work presented in this thesis have been published in the following articles:

Book

M. Thida, H.-L. Eng, D. N. Monekosso, P. Remagnino, Video understanding/behavior analysis, in: Synthesis Lectures on Image, Video, and Multimedia Processing, July 2013, to be published.

Awards

IES (Institution of Engineers Singapore) Prestigious Engineering Achievement Award 2008.

IWA (International Water Association) Honour Award for Applied Research Category 2012.

M. Thida and H.-L. Eng, "Learning group behaviour: Detecting water toxicity by biological monitoring," in International Summer School on Pattern Recognition, vol. Springer: **Best Poster Award**, 2011.

Chapter 2

- **M. Thida**, Yoke Leng Yong, Pau Climent-Pérez, H.-L. Eng, and P. Remagnino, A Literature Review on Video Analytic of Crowded Scenes, in "Intelligent Multimedia Surveillance: Current Trends and Research", 2013 Editors. Pradeep Atrey, Mohan Kankanhalli and Andrea Cavallaro. (Accepted)

Chapter 3

- **M. Thida**, P. Remagnino, H.-L. Eng, A particle swarm optimisation approach for multi-objects tracking in crowded scene, in proceedings of IEEE International Workshop on Visual Surveillance, 2009, pp. 1209 – 1215.
- **M. Thida**, H.-L. Eng, D. N. Monekosso, P. Remagnino, A particle swarm optimisation algorithm with interactive swarms for tracking multiple targets, Applied Soft Computing, vol. 13, pp. 3106-3117, 2013

Chapter 4

- **M. Thida**, H.-L. Eng, and P. Remagnino, " Laplacian Eigenmap with Temporal Constraints for Local Abnormality Detection in Crowded Scenes," in IEEE Transactions on Systems, Man, And Cybernetics Part B, vol. 99, pp. 1–10, February 2013.
- **M. Thida**, H.-L. Eng, D. N. Monekosso, and P. Remagnino, "Learning video manifolds for content analysis of crowded scenes," Information Processing Society of Japan (IPSJ) Transactions on Computer Vision and Applications, vol. 4, pp. 71–77, May 2012.
- **M. Thida**, H.-L. Eng, and P. Remagnino, "Laplacian Eigenmap with Temporal Constraints for Local Abnormality Detection in Crowded Scenes," in Lecture Notes in Computer Science: Computer Vision, vol. 6492/2011. Springer, pp. 439–449, November 2010.
- H. Lu, H.-L. Eng, **M. Thida**, and K. Plataniotis, "Visualisation and clustering of crowd video content in MPCA subspace," in ACM Conference on Information and Knowledge Management, pp. 1777–1780 October 2010.

Appendix B

- **M. Thida** and H.-L. Eng, "Learning group behaviour : Detecting water toxicity by biological monitoring," in International Summer School on Pattern Recognition, vol. Springer: Best Poster Award, 2011.
- **M. Thida**, H.-L. Eng, and Bong Fong. Chew, "Automatic analysis of fish behaviour and abnormality detection," in In proceedings of IAPR Conference on Machine Vision Applications, May 2009, pp. 278–282.
- Bong Fong. Chew, H.-L. Eng, and **M. Thida**, "Vision-based real-time monitoring on the behaviour of fish school," in In proceedings of IAPR Conference on Machine Vision Applications, May 2009, pp. 90–94 .

To my parents and siblings,
for their encouragement and love.

Acknowledgement

First and foremost, I would like to express my deep appreciation and sincere gratitude to my director of study - Prof Paolo Remagnino for his useful advices, patient discussions and moral support. Special thanks go to Dr Dorothy N. Monekosso for her kind help and great suggestions.

I would like to express my deep and sincere thanks to my local supervisor, Dr. Eng How-Lung from Institute for Infocomm Research (I2R, Singapore) for giving me confidence and support to embark on my PhD programme. This work would not be possible without his valuable advices, guidance and encouragement during the course of this research.

I also would like to thank to my PhD examiners: Prof. Andrea Cavallaro and Dr Jean-Christophe Nebel for their advice and feedbacks on the thesis. It is my pleasure to extend my warmest thanks to my friends, ex-colleagues and colleagues from Institute for Infocomm Research (I2R, Singapore) for their continued support during all these years. I would like to give special thanks to my best friends, Ms Chew Boon Fong and Ms You Yilun for their understanding, encouraging and sharing my stress throughout the hard time.

Last but not least, I wish to thank my parents and siblings, to whom this thesis is dedicated, for years of unconditional love and support. Especially, I would like to thank my little sister, Su Myat Mon, for helping me to maintain a relatively balanced life despite the pressures of doing a PhD.

Table of Contents

List of Tables

List of Figures

Acronyms and Abbreviations

The following acronyms and abbreviations are used in this work:

DCM	Discrete Choice Model
MCMC	Markov chain Monte Carlo
PTZ	Pan-Tilt-Zoom
TREC	Text REtrieval Conference
TRECVID	TREC Video Retrieval Evaluation
PETS	Performance Evaluation of Tracking and Surveillance
PSO	Particle Swarm Optimisation
HOG	Histogram of Oriented Gradients
CAVIAR	Context Aware Vision using Image-based Active Recognition
CLEAR	CLassification of Events, Activities and Relationships
MOTP	Multiple Object Tracking Precision
MOTA	Multiple Object Tracking Accuracy
PCA	Principal Component Analysis
MDS	Multi-Dimensional Scaling
ISOMAP	ISOmetric feature Mapping
LLE	Locally Linear Embedding

LE	Laplacian Eigenmaps
ST-LE	Spatio-Temporal Laplacian Eigenmaps
BIC	Bayesian information criterion
ROC	Receiver Operating Characteristic
GMM	Gaussian Mixture Model
EER	Equal Error Rate
TPR	True Positive Rate
FPR	False Positive Rate

"The most beautiful experience we can have is the mysterious. It is the fundamental emotion which stands at the cradle of true art and true science."

Albert Einstein

1

Introduction

In the last decade, a surveillance system has become an integral part of security and law enforcement in today's society. A vast number of surveillance systems are being installed everywhere, ranging from residential areas to public spaces such as airports and shopping malls. As these surveil-

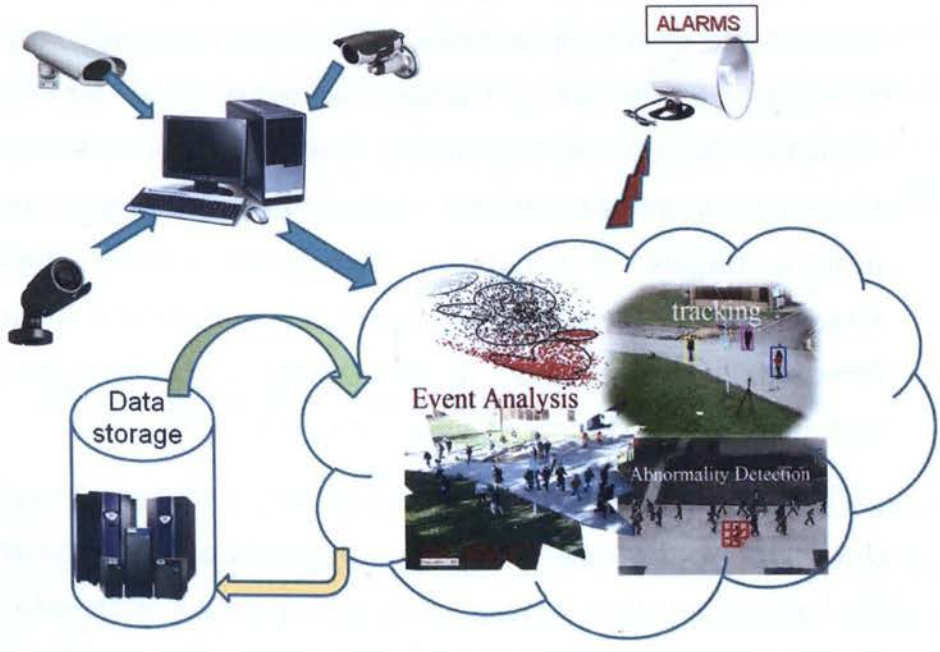


Figure 1.1: An illustration of a process of automated video context analysis.

lance systems collect a huge amount of video data everyday, it is important to automate the process of video context analysis (Figure 1.1). Automating

surveillance tasks such as intruder detection, people tracking, detection of abandoned luggage and abnormal behaviour is a desirable and interesting problem to solve.

Computer vision researchers have worked to improve the accuracy and robustness of algorithms to automatically analyse the context of a video. However, there are still many unsolved problems in analysing video context for non-ideal conditions such as a cluttered and unknown environment, for surveillance of indoor and outdoor scenes. In addition, most research has focused on modelling and detecting anomalies of isolated or independent individual behaviours. However, examining individual behaviour of a target in isolation is insufficient to describe potential abnormal behaviours involving multiple targets in a complex scene.

On the other hand, with the increased deployment of surveillance systems in real-world applications and various scenarios ranging from home security to public safety, the demand for the video context analysis has changed. The interest has shifted from the understanding of actions performed by individuals to the analysis of a complex behaviour involving multiple targets. For instance, surveillance systems deployed in public spaces, such as airports or shopping malls, monitor a scene involving over hundreds of people. Thus, more intelligent algorithms are required to detect anomalous events involving multiple targets.

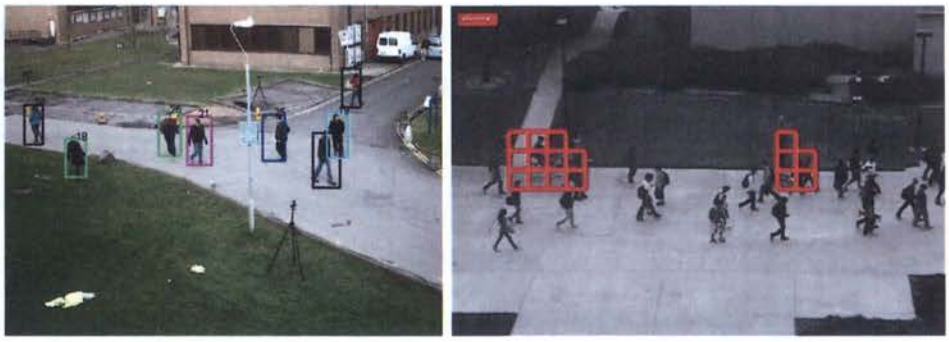
The complexity of the problem and the challenges described above motivated me to devise advanced computer vision algorithms to analyse behaviours of multiple targets. Firstly, this thesis addresses the problem of tracking multiple targets in a complex scene. This problem arises in a variety of different contexts. For instance, at locations such as train stations, airports, shopping malls, security officials are interested to track some people in the crowd, to keep an eye on their activities. Several computer

vision techniques have been developed to track individual targets. However, these techniques can not be directly applied in crowded scenes due to complex interactions among targets. Therefore, it is important to have a mechanism to handle social interactions among targets, while developing a method to track multiple targets in a complex scene.

Next, novel methods are developed to analyse and detect abnormal events in a scene involving multiple targets. In a scene where multiple targets enter and leave at the same time, tracking multiple targets is very difficult, if not feasible. In addition, as the number of targets increases in the scene, interactions between targets are unavoidable and the number of pixels belonging to a target decreases. Hence, an individual-based tracking approach becomes infeasible in this environment. A coarse level analysis of the scene where the group is considered as a single entity is a better solution for understanding a crowded scene. However, considering a group as a single entity may miss localised abnormality happening in a monitored scene. Therefore, it is important to develop different computer vision algorithms to address different challenges encountered in different scenarios.

1.1 Aims and Objectives

The goal of this thesis is, as shown in Figure 1.2, to devise computer vision algorithms capable of interpreting the context of videos involving multiple targets. Specifically, this work focusses on two particular tasks in the research of video context analysis. The first task is to develop techniques for tracking multiple moving targets. The second task is to detect and localise abnormal regions in crowded scenes.



(a) (b)

Figure 1.2: The objectives of the work in this thesis: a) To track individual targets and b) to detect abnormal behaviour in a crowded scene.

1.2 Challenges

Addressing the objectives listed above is a difficult task, particularly in uncontrolled environments. Some challenges encountered are listed below:

Frequent Interactions

In a crowded scene, interactions among targets are unavoidable and the occlusion is often persistent. In addition, as the number of targets increases in the scene, the number of pixels belonging to a target decreases and the appearance information becomes ambiguous. This makes the

Figure 1.3: Some examples of multiple targets tracking in a crowded scene. Frequent interaction among targets and occlusions increases the difficulty of the tracking task.

tracking task more challenging and it is difficult to persistently track the targets throughout the scene. Figure 1.3 shows some example scenes studied in this thesis.

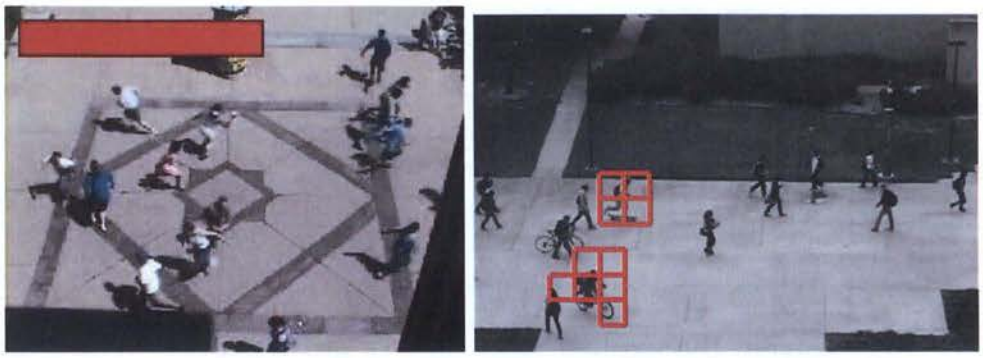
Modelling Behaviours

While the modelling of actions performed by a single target is yet to be fully solved, the explicit modelling of group/crowd behaviours faces even more challenges as there is a large quantity of possible behaviours. Unpredictable interactions among individuals pose significant challenges for approaches based on detection and tracking of individual targets. Some activities analysed in this thesis are shown in Figure 1.4.

Figure 1.4: Examples of different crowd behaviours studied in this thesis.

Localising Abnormal Regions

The next challenge is to detect and localise abnormal behaviours arisen due to an unexpected action of an individual in a crowd. When there is no scene layout, the motion of each target is random and each spatial location can support more than one behaviour. Hence, using global frame features, localised abnormal behaviours will be averaged among all the other actions taking place and hence be difficult to detect. Examples of global and local abnormal events are shown in Figure 1.5.



(a) (b)

Figure 1.5: Examples of crowded scenes and abnormal events: (a) global abnormal event: sudden movement of people in a crowd (b) local abnormal events: unexpected actions of an individual in a crowd.

1.3 Nomenclature

Many terms that are employed to describe phenomenon related to research problem in this thesis are defined loosely in the literature. To avoid confusion, definitions and explanation of these terms are provided in this section:

- The term *crowded/complex scene* is used to refer to a scene that contains more than ten people with frequent interactions.
- The term *abnormal behaviour* is used to refer to a region of the scene or the whole scene where the behaviour of the crowd is different from its learnt patterns.
- The term *localised abnormal region* is used to refer to a region of the scene where the behaviour of an individual is different from the rest in the scene.
- The term *context* refers to the contextual knowledge or motion information in the monitored scene.
- The term *precision* and *MOTP-multiple object tracking precision* are

used interchangeably, referring to a quantity that measures the average distance between the centroid positions of targets returned by the tracker and the centroid given in the ground-truth.

- The term *accuracy* and *tracking accuracy* are used to refer to a metric for measuring the accuracy of a tracker with regard to False Negatives (undetected targets) and False Positives (detected boxes that do not overlay any ground truth area) and identity switches (the number of switches in the tracker output ID for a tracked target.).

1.4 Contributions

The major contributions of this thesis to the contextual analysis of videos are as follows:

1. **Contribution to Multi-target Tracking**

   A new tracking method is developed to track multiple moving targets in a crowded scene. Specifically, this method is designed to work even when there are frequent interactions among targets and number of targets are unknown and varying over time. The main contribution to the state of the arts is introducing an idea of multiple interactive swarms to the standard particle swarm optimisation (PSO) algorithm to track multiple pedestrians in a crowd. The contribution constitutes incorporating constraints provided by the social behaviour (motion information among pedestrians), temporal continuity of target tracks and the strength of person detection.

   This work has been published in the conference proceeding of IEEE visual surveillance workshop **[ThidaVS09]** and the applied soft computing journal **[ThidaASC12]**.

2. **Contribution to Abnormality Detection**

   Two novel manifold learning-based algorithms are developed for the detection of anomalies in a crowded scene. The major contribution of the first approach is the application of manifold learning algorithms for abnormality detection in a crowded scene. Specifically, this approach exploits temporal coherence between video frames and uses the manifold learning algorithm, in particular the Laplacian Eigenmap, to discover different crowd activities from a video. This method provides a compact, yet informative representation for visualising and identifying different crowd events in a low dimensional space.

   The second approach contributes to the state of the arts by employing a manifold learning algorithm to detect and localise abnormal regions in a crowded scene. This approach captures the spatial and temporal variations of local motions using a graph-based embedding method. This approach provides a tool not only to detect abnormal crowd activities but also to localise the regions which show abnormal behaviour.

   The results on abnormality detection were published in the conference proceedings **[ThidaACCV10]**, **[ThidaACM10]** and a scientific journal **[ThidaCVA12]**. The algorithm for local abnormality detection is published in an IEEE journal **[ThidaSMC12]**.

1.5 Organisation of the Thesis

Figure 1.6 illustrates an overview of the thesis. This chapter has introduced the problem, the research questions and the contributions to the field.

Chapter 2 reviews the state-of-the-art literature on automated video

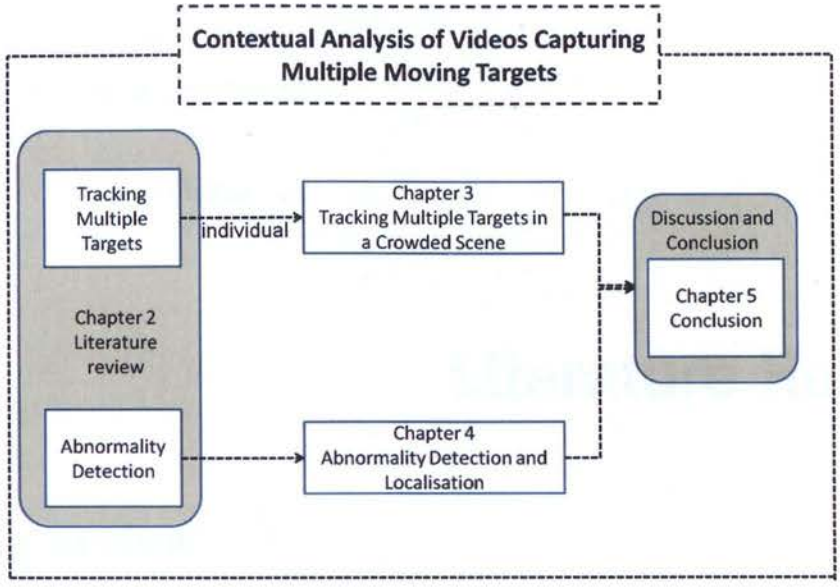


Figure 1.6: A graphical illustration of the thesis structure.

analysis of a crowded scene. The literature review mainly focusses on specific techniques for tracking individuals in a crowd and understanding crowd behaviour.

Chapter 3 presents a particle swarm optimisation framework for tracking individual targets in a crowded scene. This chapter discusses how the social interaction of targets can be integrated into a particle swarm optimisation framework to track multiple targets with heavy social interactions and frequent occlusions.

Chapter 4 presents a spatio-temporal manifold embedding framework to detect abnormality in a crowded scene. This chapter presents a way to discover an embedded space which captures dynamics and behaviours of the crowd and steps involved in detecting and localising abnormality in a crowded scene.

Chapter 5 provides conclusions and suggests a number of research directions to be pursued for future work.

"A man who reviews the old so as to find out the new is qualified to teach others."

Confucius

2

Literature Review

2.1 Overview

Automated video content analysis has been an active research area in the field of computer vision in the last few years. This strong interest is driven by the increased demand for public safety at places such as airports, train stations, malls, and stadiums, etc. In such scenes, many algorithms which consider an individual in isolation (i.e. individual object segmentation and tracking) often face difficult situations such as complex dynamics and severe occlusions in the scene. For this reason, in recent years, many computer vision algorithms are being explored to address the problems of analysing a video involving multiple targets.

This chapter presents a review and systematic comparison of the state-of-the-art methods in the domain of video context analysis. Particularly, this review focusses on specific techniques for tracking multiple targets and understanding crowd behaviour.

2.2 Tracking Multiple Targets

Tracking is one of the highly researched areas in the field of computer vision. The complexity of tracking algorithms depends on the context and environment in which the tracking is performed. In the context of crowd video analysis, the problem of tracking individuals within a crowd introduces additional complexity due to the interactions and occlusions between people in the crowd. A number of tracking methods has been proposed to overcome the challenges encountered in a crowded scene. In this section, some popular human tracking methods in the context of crowd video analysis are discussed. The reader is referred to the survey by Yilmaz *et al.* [15] for a comprehensive review of various trackers.

A number of tracking methods [43, 44, 70, 112, 163, 173] have been proposed in the past two decades. Most of the existing tracking methods can be seen as a dynamic optimisation process, which search the best match of the target descriptors in subsequent frames. Mean shift [43], which is one of the kernel-based tracking algorithms, has been proved an efficient tool to handle partial occlusions. It is an iterative process for searching local maxima of a similarity measure between the kernel density functions (for instance, colour histograms) of the target model and a candidate region. This method is prone to fail if tracked targets are moving fast or when occlusions exist. In [120], a covariance-based tracker is proposed to perform an exhaustive search of the model descriptor in the whole image. The advantage of the covariance-based tracker is its ability to combine spatial and statistical properties of tracked targets. However, its exhaustive search has problems when heavy occlusion and clutter background occur. The particle filter [24, 34, 70] has been widely used in object tracking, due to its ability to handle cluttered background. This method formulates the

object tracking as finding the maximum of the posterior distribution of the state space using a large number of weighted particles.

2.2.1 Tracking Multiple Targets using Particle Filter

The particle filter-based tracking framework has been extended in a series of papers [12, 39, 57, 80, 113] for tracking multiple targets. For example, Okuma *et al.* [113] extend a particle framework by incorporating a cascaded Adaboost algorithm for the detection and tracking of multiple hockey players in a video. The Adaboost algorithm is used to generate detection hypotheses of hockey players. Once the detection hypotheses are available, each hockey player is modelled with an individual particle filter that forms a component of a mixture particle filter. Similarly, Ali and Dailey [12] combine an AdaBoost cascade classifier-based head detection algorithm and the particle filtering method for tracking multiple persons in high density crowds. The performance is further improved by a confirmation-by-classification method to estimate confidence in a tracked trajectory.

Table 2.1: Summarisation of Particle filter-based tracking methods

Works/Papers	Additional Information or Method	Multi-view/-target
[91]	Contour information	No
[121, 139, 156]	SIFT, Harris-SIFT	No
[142]	Histogram of Oriented Gradients (HOG)	No
[25]	Mean Shift / Joint Probabilities	No
[67, 166]	None: Changes in the transition model	No
[110]	None: Particles are fused among views	Multi-view
[145, 171]	Particle filter for blob tracking	Multi-target
[99]	None: GCs[a] used to recover contours	No
[12, 113]	AdaBoost, Cascaded AdaBoost	Multi-target
[80]	MRF[b], MCMC[c]	Multi-target
[39]	NN Data Association, Mean-shift	Multi-target

[a] Graph Cuts.

[b] Markov Random Field.

[c] Markov Chain Monte Carlo.

Table 2.1 presents a list of particle filter-based methods for tracking multiple targets. Both single and multiple view methods are presented, as well as single and multiple target ones. Despite its success in some applications, the particle filter is less efficient in a high dimensional space as the number of particles required increases exponentially with the dimensionality of the studied state space.

2.2.2 Tracking Multiple Targets using Additional Cues

In recent years, many research work has demonstrated that to employ high level cues for tracking multiple targets in a complex scene. These high level cues can be contextual information such as motion information, scene structure or the social interactions among the people in the crowd. An overview of tracking algorithms that incorporate different high level contextual information is illustrated in Figure 2.1.

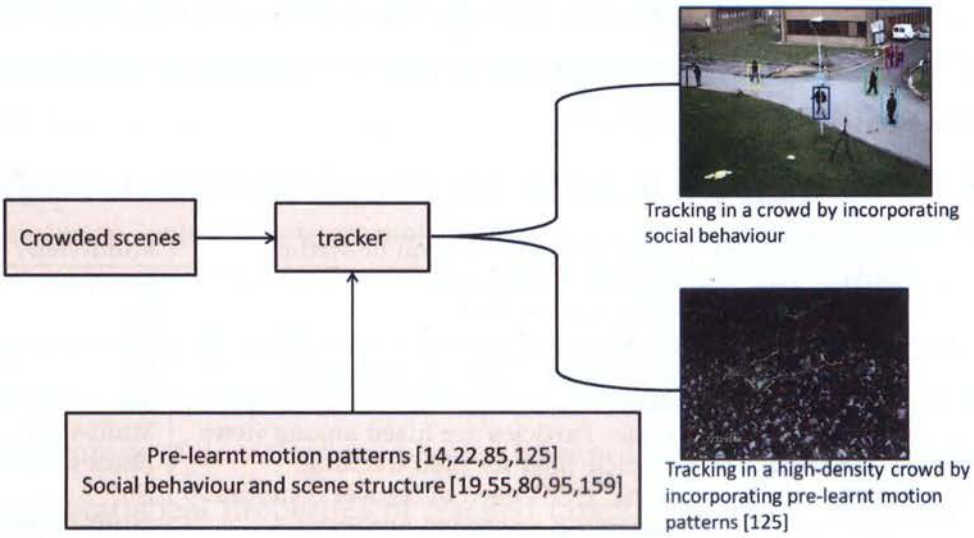


Figure 2.1: An overview of different tracking algorithms that incorporate high-level contextual information.

2.2.2.1 Pre-learnt Motion Patterns

Antonini *et al.* [22] use a discrete choice model (DCM) as motion priors to predict human motion patterns and then, fuse this model in a human tracker for improved performance. Similarly, Ali *et al.* [14] propose to exploit contextual (motion) information for tracking multiple people in a structured crowded scene. Assuming that all participants of the crowd are moving in one direction, Ali *et al.* learn the direction of motion as a prior information based on floor fields. The authors have demonstrated that a higher-level constraint greatly increases the performance of the tracker. However, floor fields can be learned only when the scene has one dominant motion. As a result, the method proposed in [14] cannot be applied for crowded scenes where the motion of a crowd appears to be random with different participants moving in different directions over time. Some examples of unstructured crowded scenes include crowds at exhibitions, sporting events and railway stations. This shortcoming is addressed by Mikel *et al.* [125] where the authors employ a correlated topic model for modelling random motions in an unstructured crowded scene. Similarly, L. Kratz and K. Nishino [85] employ the normal motion pattern to predict tracking individuals in a crowd scene where the normal motion pattern is learnt based on local motion at fixed-size cells.

2.2.2.2 Social Interactions

Another interesting direction of tracking multiple targets is to integrate social interaction of targets in the tracking algorithm. This idea is motivated by the behaviour of targets in a crowd. In crowded scenarios, the behaviour of each individual target is influenced by the proximity and behaviour of other targets in the crowd. Several methods [19, 55, 80, 95, 159]

have proposed to integrate the social interactions among targets in the tracking algorithms. This direction has shown promising performance to track multiple targets in crowded scenes. An early example which models the social interaction of targets is *Markov Chain Monte Carlo*-based (MCMC) particle filter [80]. Their method models social interactions of targets using Markov Random Field and adds motion prior in a joint particle filter. The traditional importance sampling step in the particle filter is replaced by a MCMC sampling step. French *et al.* [55] extended the method in [80] by adding social information to compute the velocity of particles. In [159], the authors formulated the tracking problem as a problem of minimising an energy function. The energy function is defined based on both the social information and physical constraint in the environment. Their preliminary results indicate that social information provides an important cue for tracking multiple targets in a complex scene.

2.2.3 Multiple-camera Tracking

Researchers have also explored the use of multiple cameras for tracking people under severe occlusion in a complex environment. Multiple camera tracking methods intend to expand the monitored area and provide complete information about interesting persons by gathering evidences from different camera views. Lee *et al.* [89] propose a multiple people tracking method for wide-area monitoring. An automated calibration method is introduced to find correspondences between distributed cameras. In their method, all camera views are calibrated to a global ground-plane view based on geometric constraints and tracking trajectories from each view. Another example in a similar context can be found in the papers by Khan and Shah [78, 79]. A planar homographic occupancy constraint that combines foreground likelihood information from different views is

proposed for detection and occlusion resolution.

Another use of multiple cameras is to track people in an environment covered by multiple cameras with overlapping views. Mittal and Davis [23] use pairs of stereo cameras and combine evidences gathered from multiple cameras for tracking people in a cluttered scene. Foreground regions from different views are projected back into a $3D$ space so that the endpoints of the matched regions yield $3D$ points belonging to people. Dockastader and Tekalp [51] employ a Bayesian network for fusing $2D$ position information acquired from different views to estimate the $3D$ coordinate position of the interested person. Finally, a layer of Kalman filtering is used to update the position of people. A combination of static and pan-tilt-zoom (PTZ) cameras for multiple camera tracking is introduced in [135]. The static cameras are used to provide a global view of the interested persons when the PTZ cameras are used for face recognition of people.

The brief overview of the research literature indicates that multiple camera tracking methods provide an interesting mechanism to handle severe occlusion and to monitor large areas at public spaces. However, advantages of the multiple cameras come together with additional issues such as camera calibration, matching information across the camera views, automated camera switching and data fusion. These challenges are still yet to be solved. On the other hand, integrating the social interaction among targets in the tracking algorithms has shown promising performance to track individual targets in a crowd.

2.3 Analysis of Crowd Behaviour

Automated video content analysis is highly desirable but an open ended issue with high complexity. Early research focusses on recognition or detection of behaviour of a single target [58, 69]. In these approaches, an activity is detected from an unseen sequence and classified as one of predefined activities learnt during the training stage. Recently, the focus has been shifted from monitoring the behaviour of a single target to understanding the behaviour of multiple targets. Many research works, for instance [161, 174], have been proposed to analyse the dynamics of a large group, for instance, learning motion pattern or pathways to detect abnormal motion.

There are two major approaches for learning motion pattern and detecting abnormal behaviour: micro-observation and macro-observation. The micro-observation approach analyses the crowd motions based on the details of individuals moving in the scene. This approach, in general, requires a precise detection and tracking individuals in a crowd. On the other hand, the macro-observation approach describes the crowd motion from a global aspect using an abstract representation of video frames.

2.3.1 Abnormality Detection using Micro-Observation

Micro-observation approach depends on the analysis of video trajectories of moving entities. This approach, in general, contains the following steps:

1. detection of moving targets in the scene,
2. tracking of detected targets and
3. analysis of trajectories to detect dominant flows and to model typical motion patterns.

Researchers have used different detection and tracking algorithms discussed in aforementioned sections to generate reliable trajectories. Given a set of trajectories, dominant motion directions are detected by clustering these trajectories in space and time. Then, models of motion patterns are built to represent the semantic status of the scene and to detect hazardous or emergency events. A list of popular methods using micro-observation approach is given in Table 2.2.

Paper/Work	Object detection and Tracking	Motion Pattern Extraction
Hu *et al.* [68] Wang *et al.* [155], Basharat *et al.* [27]	Track foreground regions via background subtraction approaches	*Iterative Clustering:* trajectories are clustered hierarchically and each motion pattern is represented with a chain of Gaussian distribution.
Piciarelli *et al.* [117]	Track pedestrians using a combination of Kalman-based and CamShift trackers.	*Support Vector Machine:* Abnormal event is detected using a one-class support vector machine (SVM).
Johnson and Hogg [74]	Track pedestrians using an active shape model-based tracker	*Neural networks:* A model of the distribution of typical trajectories is learnt using neural networks.
Jiang *et al.* [73]	Track foreground regions via background subtraction approaches.	*Hidden Markov Model (HMM) and dynamic hierarchical clustering (DHC):* Trajectories are modelled with a hidden Markov models and a dynamic hierarchical clustering method (DHC) is employed to extract abnormal trajectories.

Table 2.2: A list of popular methods for micro-observation approach.

Hu *et al.* [68] propose a technique in which statistical motion patterns of the scene are learned automatically for anomaly detection. Given a video, trajectories of moving objects are first extracted by clustering foreground pixels using a fast fuzzy k-means algorithm. Using spatial features, these

trajectories are initially clustered into various categories. Each of these clusters is then further grouped into sub-categories using temporal features. The clustered trajectories are employed to model statistical motion patterns of the scene where each motion pattern is represented with a chain of Gaussian distributions. Based on learned motion patterns, statistical methods are used for abnormality detection and path predictions.

Similarly, Wang *et al.* [155] propose a trajectory similarity measure to cluster trajectories so that each cluster represents a significant activity in the scene. Trajectories are first clustered into vehicles and pedestrians, and then further grouped using spatial and velocity distributions. The statistical models for different paths of the scene are learned using probability distributions. Then, the semantic scene models are used for the detection of anomalous activity in the scene. Basharat *et al.* [27] propose to model the motion patterns of a scene with a Gaussian mixture model of speed and size features from trajectories. The learnt model is later used to detect abnormal events and improve object detection algorithm.

Piciarelli *et al.* [117] employ a one-class support vector machine (SVM) method to cluster trajectories based on geometric features. Each trajectory is represented by $2-D$ coordinates and sub-sampled to obtain a fixed-dimension feature vector. The presence of outlier in the data-set is detected by using a one-class SVM. The use of neural networks for modelling motion patterns from trajectories is proposed by Johnson and Hogg [74]. They use an active shape model-based tracker to obtain trajectories of pedestrians. Trajectories are employed to model the distribution of typical motion patterns using neural networks. The learnt motion patterns are used for abnormal event detection and track prediction. Jiang *et al.* [73] propose to model trajectories using a hidden Markov models (HMM). These trajectories are then grouped using a dynamic hierarchical clustering method

to extract unusual trajectories from normal ones. The use of HMM for modelling the common event behaviours can also be found in [124].

However, the performance of micro observation-based methods heavily depends on the ability of tracking algorithms. As discussed in [15], the tracking itself is a complex problem and it is hard to obtain the reliable tracking results in a crowded scene. To address this limitation, researchers have proposed to use holistic properties of crowded scenes where long-term tracks of moving targets are not available or not reliable.

2.3.2 Abnormality Detection using Macro-Observation

To learn the typical motion patterns in a crowded scene, macro observation-based methods utilise holistic properties of the scene such as motions in local spatio-temporal cuboid or instantaneous motion.

2.3.2.1 Optical Flow Feature

Optical flow, which is a dense field of instantaneous velocities computed between two consecutive frames, is a commonly used feature. Given a video, the first step is to segment the input video into smaller video clips and to compute pixel-wise optical flow between consecutive frames of each clip using the techniques in [26, 63, 96]. The extracted flow vectors may contain noise and redundant information. In order to reduce the computational cost and remove noise, researchers utilise unsupervised (Andrade *et al.* [16, 17] and Yang *et al.* [161]) or supervised (Hu and Shah [65, 66]) dimensional reduction techniques. The next step is to find the representative motion patterns of the scene by merging flow vectors from all video frames. This can be in the form of sink seeking process, interaction force modelling and clustering.

Sink Seeking Process

In the sink seeking process, a grid of particles is overlaid on the first frame of the video clip and advected using a numerical scheme. The path taken by a particle to its final position is called a sink path and thus, the process of finding sinks (exits) and sink paths is called a sink seeking process. Hu and Shah [65, 66] carry out sink seeking process for each particle and thus generate one sink path per particle. These sinks and sink paths are later clustered to extract the dominant motion paths of the scene using an iterative clustering algorithm. On the other hand, Ali and Shah [13] generate a static floor field where each particle holds a value that represents the minimum distance to the nearest sink form its current location. The dominant motion paths learnt by the sink seeking process can be used to detect abnormality or improve the tracking algorithm [14].

Interaction Force Modelling

The typical behaviour of a crowd can also be modelled using interaction forces of people in the scene. For example, Mehran *et al.* [104] employ the optical flow vectors to model pedestrian motion dynamics using a social force model. Social force models [60] have been used in many studies in computer graphic fields for creating animations of the crowd [32]. In this model, the motions of pedestrians are modelled with two forces: a personal desire force and an interaction force. The interaction force is defined as an attractive and repulsive force between pedestrians. In [104], an interaction force between pedestrians is estimated based on optical flow computed over a grid of particles. The normal pattern of this force is later used to model the dynamics of a crowded scene and detect abnormal behaviours in crowds.

Optical Flow Clustering

Another approach is to cluster optical flow vectors in a low dimensional space. For instance, Andrade *et al.* [16, 17] models the principal components of the optical flow vectors in each video clip using Hidden Markov Models. Then, video segments which have similar motion pattern are grouped together using the spectral clustering method. The resulting clustered video segments are modelled using a chain of HMMs to represent the typical motion pattern of the scene. The emergency events in the monitored scene are detected by finding deviations from the obtained model.

2.3.2.2 Other Features

In addition to optical flow information, other features such as spatial temporal interest points [108] or spatio-temporal gradient [84, 100] are used to model the regular movement of a crowd. In [84], the coupled HMM is trained based on the distribution of spatio-temporal motions to detect localised abnormalities in densely crowded scenes. Mahadevan *et al.* [102] combine motion information and appearance features to represent the local properties of a scene. The normality of a scene is learned using a mixture of dynamic textures. Then, temporal and spatial abnormalities are separately detected by finding deviations from the normal pattern. Their method has been proved to achieve the better performance than state-of-the-art methods at high computational cost. To address this limitation, Reddy *et al.* [123] propose a simpler method using a set of similar features including shape, size and texture extracted from foreground pixels. The computational cost is reduced by removing background noise and considering each feature type individually. Compared to [102], the method proposed by Reddy *et al.* [123] achieves considerably better results.

2.3.3 Event Detection

The current research on detection of pre-defined events in a scene is still limited. Most of the existing work focusses on a single target where the event of interest is defined in terms of individual targets [33, 38, 58, 61, 69, 75, 76, 107, 133, 152, 158, 164] or a small group of targets [36, 62, 128–130].

2.3.3.1 Detecting Individuals or Multi-agents Events

The launch of Text REtrieval Conference (TREC) video retrieval evaluation (TRECVid) [140] in 2008 motivated vision researchers to work on detection of video events in a crowded scene. The TRECVid provided a standard benchmark for detection of individuals or multi-agents events in an airport surveillance video. Some examples of events provided by [140] are individual events such as *CellToEar: People-calling-cellphone, Object-Put: People-dropping-something and Pointing: People-pointing-something* or multi-agents events such as *Opposing Flow:two group of people walking in opposite, People meeting: two or more people coming towards each other.* Many algorithms [90, 136, 144, 174] have been proposed to detect such events. Their algorithms differ from each other based on 1) the model they used to represent the events or the actions and 2) the classification techniques used to detect different events. For instance, Zhu *et al.* [174] use the spatio-temporal information of low-level features, *e.g.* image gradient and optical flow fields for detection of individual events while Lee *et al.* [90] detect *Meeting Event* based on analysis of video trajectories. The aforementioned methods focus on the detection of individual or multi-agents events in a crowded scene where the people involved in the event is limited to less than ten.

2.3.3.2 Detecting Crowd Events

Recently, Performance Evaluation of Tracking and Surveillance (PETS) [9] launched a new data-set of crowd events performed by 40 people. These events involve walking, running, dispersion, grouping, etc. A number of papers have been published for detecting these crowd events using flow vectors [29, 40] and histogram of oriented gradients [56]. However, the results reported in [29, 40, 56] indicate that the detection accuracy still needs an improvement. Research in this area is still in its infancy with respect to the understanding and recognising the actions of individuals or groups of people. In addition, the manual-based annotation of crowd events becomes challenging as the crowd density increases, due to the huge quality of possible events and context dependency.

2.3.4 Graph-based and Manifold Learning Algorithms

Graph-based methods [35, 50] and manifold learning algorithms [118, 147, 149, 150] have been used for the analysis of video sequences. In [50], Ding *et al* detect repetitive sequential activities using a temporal graph where vertices correspond to pre-defined primitive events. Brendel *et al.* [35] advance the prior work by representing a video of a group activity as a spatiotemporal graph where nodes correspond to multi-scale video segments. Graph matching is employed to recognise group activities such as hand-shaking, kicking, punching, and pushing, etc. However, this method focusses on the modelling of group activities where people involved in the event is limited to small groups with two or three people.

On the other hands, Gaussian process related algorithms such as Gaussian Process Latent Variable Model (GPLVM) [87] and graph spectral methods such as isometric feature mapping (ISOMAP) [146], Local Linear Em-

bedding (LLE) [126] and Laplacian Eigenmaps (LE) [28] have been employed to embed high dimensional video data in a low dimensional space. For instance, in [118], ISOMAP is used to represent a high-dimensional video as a trajectory in the manifold space. Then, different tasks of video content analysis such as visualisation and video event segmentation are performed by analysing the embedded video data. However, these methods ignore the temporal coherence between frames, though this cue provides a useful information about neighbouring structure in video data.

In recent years, several manifold learning methods such as Gaussian Process Dynamical Models (GPDM) [153], back-constrained GPLVM (BC-GPLVM) [88], sptio-temporal Isomap (ST-ISOMAP) [71] and temporal Laplacian Eigenmap (TLE) [98, 149] have been proposed for time series analysis where the temporal ordering of input sequences is considered in the manifold structure. Wang *et al* [154] employ GPDM for modelling temporal data of human motion while Shaobo *et al* [64] use BC-GPLVM for learning a low dimensional space of human motion using training data. The prior dynamic model learnt using BC-GPLVM is later used for tracking and estimating human poses from images captured by multiple cameras in [64]. However, these methods mainly apply manifold learning algorithms for estimating human poses and tracking motions of a single target.

In [149], Laplacian Eigenmap with a temporal constraint is employed for detecting abnormal events from a long video sequence. The authors have proved that a video corresponds to a trajectory in an embedded space and different appearances on manifolds indicate different video events. The abnormal events can be detected using a simple classifier based on the training data in the embedded space. Similarly, [147] employs a spatial temporal Laplacian Eigenmap for analysing videos of crowded scenes. The pair-wise graph was constructed between video frames in the temporal

domain. The results demonstrated that the spatial-temporal Laplacian Eigenmap provides a compact low dimensional space for clustering different crowd events and detecting abnormal events. However, the use of global frame features limits the previous approach to detect localised abnormal activities.

2.3.4.1 Local Spatio-Temporal Modelling

Instead of embedding the global frame features, another approach is to consider the spatial and temporal variations of local motions. The local motions can be obtained either by spatial division of image frame [81, 83, 84, 100, 102, 157, 161] or spatial grouping of optical flow vectors [131]. The first one is to divide an image space into cells of a specific size (e.g., 10×10 in [161]) or cuboids (e.g., $30 \times 30 \times 20$ in [84]). Then, optical flow computed in each cell is quantised into different directions. For instance, Yang *et al.* [161], considered each quantised direction of a given location as a word and cluster these video words into different cluster using a diffusion embedding method. Each node in the graph corresponds to a word and the clusters extracted in the embedded space represent the typical motion patterns of the scenes. Kim and Grauman [81] use a space-time Markov random field (MRF) graph to detect abnormal activities in video. Each node in the graph corresponds to a local region in the video frames where the local motion is modelled using a mixture of probabilistic principle component analysis. Wu *et al.* [157] use Lagrangian framework to extract particle trajectories. These particle trajectories are later used for modelling of regular crowd motion. The deviations of new motion from the learnt model indicates abnormal event.

The second one is to cluster optical flow vectors by spatial grouping as in [131]. Imran *et al.* [131] propose to cluster optical flow vectors in each

video clip into N Gaussian mixture components. Then, these Gaussian components are linked over time using a fully connected graph. The graph connected component analysis is performed to discover different motion patterns. However, their method still faces the problem of having to determine how many components should be in the mixture.

2.4 Summary

This chapter presents a review and comparative study of various topics in the area of video content analysis. The advantages and disadvantages of the state-of-the-art methods related to the work in this thesis have been discussed.

Tracking individuals in a crowd has been addressed in recent years. A major advance is the introduction of high-level crowd motion pattern as a prior into a general framework [14, 125]. However, the problem of tracking still remains as a challenging problem in the area of computer vision. One major challenge for tracking in a crowded scene is inter-object occlusion due to the interactions of participants in a crowd. There remains a gap between the state-of-the-art and robust tracking of people in a crowded scene.

During recent years there has been substantial progress towards understanding crowd behaviour and abnormality detection based on modelling crowd motion pattern. However, these approaches capture general movement of a crowd but do not accurately detect details of individual movements. As a result, the current literature in understanding crowd motion is not ready to capture the motion pattern of an unstructured crowd scene where the motion of the crowd appears to be random [125].

Future research in this area requires localised modelling of crowd motion to capture different behaviours in the unstructured crowded scene. On the other hand, the understanding and modelling of crowd behaviour remains immature despite the considerable advances in human activity analysis. Progress in this area requires further advances in modelling or representation of a crowd event and recognition of these events in a natural environment.

"Research is to see what everybody else has seen, and to think what nobody else has thought."

Albert Szent-Gyorgyi

3

Tracking Multiple Targets using Particle Swarm Optimisation

In this chapter, a population-based particle swarm optimisation (PSO) algorithm is studied. In particular, a study is carried out on the use of the standard PSO algorithm and its variants for tracking targets in surveillance videos. The proposed method extends the standard PSO algorithm to the problem of finding dynamic optima (pedestrians) where these optima interact frequently.

3.1 Introduction

Tracking multiple people and interpreting their behaviour is an important problem that arises in a variety of different contexts [15]. For instance, it is important to be able to track individuals in a crowd for public security (see Figure 3.1). Despite being a highly researched area, there are still a number of challenges to be addressed: heavy occlusions arising from interaction among targets, erratic motion of the targets and lighting condition of the scene.

In this chapter, the problem of multi-target tracking is addressed using

Figure 3.1: Tracking multiple targets in a crowd [148].

the particle swarm optimisation framework. Recently, particle swarm optimisation (PSO) [77] has gained the attentions of many researchers and it has been proved to be effective in finding the optimum of a function in a search space. In contrast to the particle filter [70], where particles move independently, PSO allows particles to interact; each particle is a candidate solution and searches the optimum using both "social interaction" and "cognitive knowledge" [116, 119]. This idea of PSO is inspired by behaviour models of bird flocking where each bird seeks a target (food) in the search space by sharing information with other birds of the swarm. This underlying concept resembles the social interaction of pedestrians in a crowd where the motion of each pedestrian is influenced by both the environmental structure and the movements of other people in the crowd.

However, the standard PSO is generally used to find a single optimum in a static search space. In contrast, the nature of tracking is dynamic where optima change over time. Thus, the standard PSO cannot be directly used to address the problem of tracking multiple targets. In this chapter, the problem of multi-target tracking is formulated as an optimisation problem of finding dynamic optima (pedestrians) where these optima interact fre-

quently. The motion prediction and social interaction is incorporated in the PSO framework such that each swarm finds the best local optimum based on its best knowledge and exchanges information with others. The main contributions of the work presented in this chapter can be summarised as follows:

1. introducing an idea of multiple interactive swarms to the standard PSO to track multiple moving targets;
2. incorporating higher level information such as social behaviour (motion information among pedestrians) in the process of finding optima in a high dimensional space;
3. integrating constraints provided by temporal continuity of target tracks and the strength of person detections.

The rest of the chapter is organised as follows. Section 3.2 describes the related work on multi-target tracking using PSO algorithm. The standard PSO algorithm is introduced in Section 3.3. Section 3.4 explains the proposed method in details. Experimental results are presented and discussed in Section 3.5. Specifically, the proposed method is evaluated for tracking fixed number of targets as well as a varying number of targets in a complex scene with severe occlusions and heavy interactions among targets. Finally, a summary is given in Section 3.6.

3.2 Literature Review on Particle Swarm Optimisation

PSO was first introduced to the problem of target tracking by M. Kölsch and M. Turk [82]. Particles were represented by the positions of *KLT* (Kanade, Lucas, and Tomasi) feature points [137]. The movement of par-

ticles were spatially confined based on the swarm behaviour using two thresholds: the first one to define the maximum distance between feature points and the other one to define the minimum distance between a particle and the swarm. A similar approach can be found in [21] where the object of interest is represented by N pixels. A swarm with N particles is initialised to track the target in an image space. The above approaches define a particle as a point and hence, their search space is limited to a two-dimensional space. A higher dimensional search space is considered in [169], where the target is represented by the centroid, the width and the height of its bounding box. Similarly, Zhang *et al.* [167] proposed a sequential PSO algorithm where temporal information is incorporated into the standard PSO.

In [160], Yang *et al.* incorporated a PSO algorithm into unscented particle filter-based tracking to avoid an impoverishment problem which is a known problem in particle filter-based tracking [70]. They have demonstrated that incorporating PSO improves the performance of the particle filter-based tracking in terms of accuracy and robustness. Recently, other hybrid trackers that incorporate the PSO algorithm into particle filter [172], Kalman filter [122] and mean shift [92], have shown that swarm optimisation improves the performance of the tracker.

Zhang *et al.* [168] proposed a species-based PSO where the global swarm is divided into many species to track multiple targets. These species track targets independently and interact only when the overlapping area between targets is greater than a particular threshold. Hence, their method requires occluded targets to be detected explicitly and a selective appearance updating scheme is used to handle occlusions. In addition, the number of targets is assumed to be fixed and known a priori, which is hard to achieve in real applications. This limits its applicability and may fail in

crowded situations with heavy interactions and frequent occlusions.

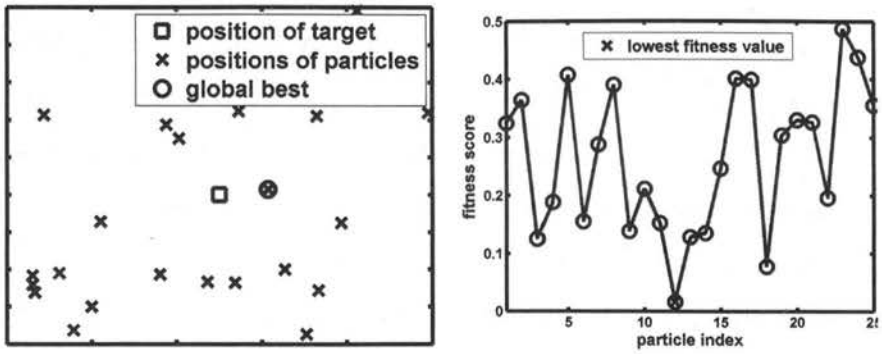
This work tracks multiple targets using a set of **interactive** swarms. In contrast to [168] where each sub-swarm (called as species in [168]) tracks a target independently, the swarms in this method track targets interactively by sharing social information among targets. This method naturally handles the occlusion problem and improves tracking accuracy and precision. The details of the method are explained in Section 3.4.

3.3 Standard Particle Swarm Optimisation

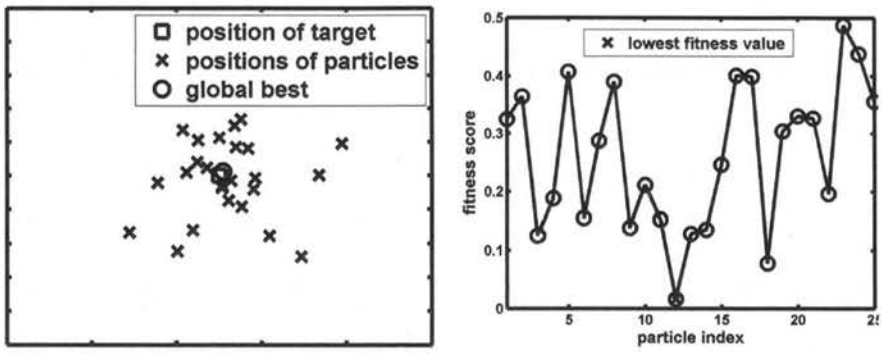
Particle swarm optimisation (PSO) is a population-based optimisation technique in which a set of potential solutions, called particles, $\{\mathbf{x}_i\}_{i=1}^{N}$ iteratively find the optimum solution in a search space. Given a $d-$dimensional search space, each particle, represented as $\mathbf{x}_i = (x_1, x_2, \cdots, x_d)$, evaluates its current position using a fitness function $f(\mathbf{x})$. This fitness function measures the closeness of the current position of the particle to the optimum solution. Mathematically, the objective of PSO can be described as:

$$\mathbf{x}_{opt} = \arg\min_{\mathbf{x}} f(\mathbf{x}) \tag{3.1}$$

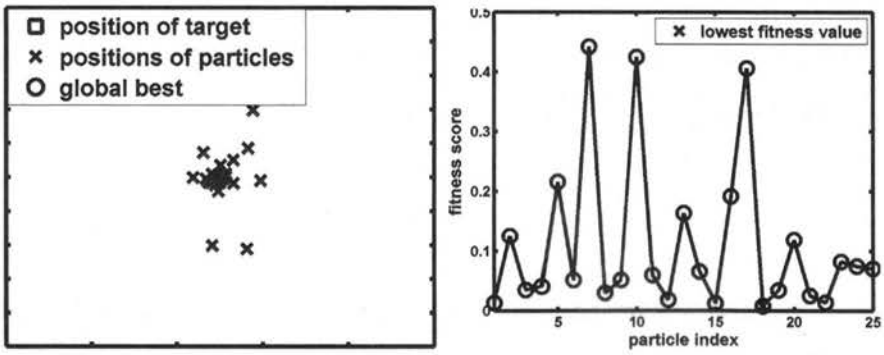
Figure 3.2 illustrates a simulation of the swarm optimisation process of finding a target in a $2D$ search space. In this simulation, the position of the target is fixed and the fitness function is defined as the Euclidean distance between the positions of particles and the position of the target: $f(\mathbf{x}) = \|\mathbf{x}_{target} - \mathbf{x}\|^2$. Figure 3.2(a) shows the distribution of particles at the first iteration $n = 0$. In the first iteration, the positions of particles are randomly initialised and each particle takes its current state as the individual best state and the state which has the smallest fitness value (nearest to the



(a) $n = 0$



(b) $n = 3$



(c) $n = 7$

Figure 3.2: Simulation of particle swarm optimisation in $2D$ search space. Figure(3.2(a)-left) shows the distribution of particles at the first iteration $n = 0$. The fitness values against the states of particles are plotted (right image), while the global best is marked with a small black 'x'. Figure (3.2(b)) and (3.2(c)) show the iteration process of particles in subsequent iterations.

target in this simulation) is selected as the global best. The positions of the particles in the first iteration $n = 0$ and the corresponding fitness values of particles are plotted in Figure 3.2(a). The current positions of particles are marked with a small black 'x' while the global best is marked with a small black circle. The movements of particles in subsequent iterations are shown in Figure 3.2(b) and 3.2(c) respectively.

In subsequent iterations, the movement of each particle depends on two important factors: $\mathbf{x}_i^b$ the best position that the i^{th} candidate has found so far and $\mathbf{x}^g$ the global best position found by the whole swarm (all particles). Based on these two factors, each candidate updates its velocity and position at the $(n+1)^{th}$ iteration as follows.

$$\mathbf{v}_i^{n+1} = \omega \mathbf{v}_i^n + \varphi_1 r_1 (\mathbf{x}_i^b - \mathbf{x}_i^n) + \varphi_2 r_2 (\mathbf{x}^g - \mathbf{x}_i^n) \quad (3.2)$$

$$\mathbf{x}_i^{n+1} = \mathbf{x}_i^n + \mathbf{v}_i^{n+1} \quad (3.3)$$

where ω is the inertia weight, the parameters φ_1 and φ_2 are positive constants, which balance the influence of the individual best and the global best position. The parameters, $r_1, r_2 \in [0, 1]$ are uniformly distributed random numbers and diversify the positions of particles. Over the last decade, many variants of the PSO algorithm have been proposed [105] and some algorithms have addressed dynamic optimisation problems [115, 162]. In the work presented by Clerc and Kennedy [42], a parameter χ, called a constraining factor, is introduced to avoid an unlimited growth of the particles' velocity. Equation (3.2) becomes:

$$\mathbf{v}_i^{n+1} = \chi \left(\mathbf{v}_i^n + \varphi_1 r_1 (\mathbf{x}_i^b - \mathbf{x}_i^n) + \varphi_2 r_2 (\mathbf{x}^g - \mathbf{x}_i^n) \right) \quad (3.4)$$

where $\chi < 1$ is defined as:

$$\chi = \frac{2}{||2 - \varphi - \sqrt{\varphi^2 - 4\varphi}||}, \text{ where } \varphi = (\varphi_1 + \varphi_2) > 4.0 \tag{3.5}$$

This method has been frequently used due to its stability and convergent ability in high-dimensional problems [86, 105]. From equation (3.4), it can be observed that the movement of each particle depends on three components: inertial velocity, cognitive effect and social effect. The first component maintains the direction of the particle during the optimisation process while the second component allows each particle to move based on its own information, i.e., its best known position $\mathbf{x}_i^b$ at the previous iteration. The third component represents the social effect, where a particle in the swarm moves towards the global best position $\mathbf{x}^g$ defined by all the members of the swarm.

The individual best, $\mathbf{x}_i^b$ and the global best, $\mathbf{x}^g$ positions are updated at each iteration, based on the fitness values at the current position $\mathbf{x}_i$. Each particle will update its current position as the best position only if the current position is closer to the target (the fitness value of the current position is smaller than the value evaluated at its previous position); otherwise its previous best position is kept. The global best is the position that has the lowest fitness value among all individual best positions. Mathematically, this can be formulated as follows:

$$\mathbf{x}_i^b = \begin{cases} \mathbf{x}_i^n, & \text{if } f(\mathbf{x}_i^n) < f(\mathbf{x}_i^b); \\ \mathbf{x}_i^b, & \text{otherwise.} \end{cases} \tag{3.6}$$

$$\mathbf{x}^g = \arg\min_{\mathbf{x}_i^b} f(\mathbf{x}_i^b) \tag{3.7}$$

where $f(\mathbf{x}_i^n)$ is the fitness value at the position $\mathbf{x}_i^n$. This process is repeated until a convergence state, as described in the next section, is achieved.

3.3.1 Convergence Criteria

In general, the convergence of a PSO algorithm is defined in terms of either the best positions of all individual particles, or the global best position found by the whole swarm.

Definition 1: $\mathbf{x}_i^{b,n} \approx \mathbf{x}_i^{b,n+1}$ and $\mathbf{x}_i^b \rightarrow \mathbf{x}^g$ for all particles $i \in (1 : N)$. This definition implies that the convergence of the process is achieved when all particles ultimately stop at the global best position. The first condition checks if all particles reach a stable state, while the second condition identifies if $\mathbf{x}^g$ is the best position of each individual particle.

Definition 2: $\mathbf{x}^{g,n} \approx \mathbf{x}^{g,n+1}$ and $f(\mathbf{x}^g) \approx 0$. This implies that the global best position that can be achieved by the optimisation process does not change any more and hence the convergence is achieved. The first condition is to check if the global best position reaches a stable state and the second condition is to identify if the global best position achieved so far is the nearest to the actual position of the target.

3.3.2 Pseudo-code

The standard PSO algorithm can be summarised as follows:

Algorithm 1: Pseudocode of a standard PSO for minimisation problem

```
Randomly generate an initial swarm :{x_i}_{i=1}^N
//Initialisation process
foreach Particle i ∈ 1 → N do
    x_i^b = x_i^n
    compute the fitness value f(x_i^b)
end
x^g = arg max_{x_i^b} f(x_i^b)
//Iteration process
for n = 1 to maximum number of iterations do
    foreach Particle i ∈ 1 → N do
        update velocity using equation( 3.2)
        update position using equation( 3.3)
        if f(x_i^n) < f(x_i^b) then
            x_i^b = x_i^n
        end
        compute the fitness value f(x_i^b)
    end
    x^g = arg max_{x_i^b} f(x_i^b)
    if convergence criteria in Section 3.3.1 are met then
        break;
    end
end
Output: the global best position: x^g
```

3.4 A Modified PSO with Interactive Swarms

This section addresses the problem of tracking multiple interacting targets in a scene. The problem of multi-target tracking is formulated as an optimisation problem of finding dynamic optima (i.e., pedestrians) where these optima interact frequently. Then, the tracking problem is addressed using a modified particle swarm optimisation algorithm. In order to handle the dynamic optimisation problem effectively, three major stages are introduced into the standard PSO framework: 1) a scheme for diversifying

particles and swarms to maintain diversity over time, 2) a novel optimisation process that integrates the concepts of multiple swarms where the PSO updating equation is modified to incorporate temporal continuity information and social interaction among targets, and 3) a swarm initialisation and termination strategy to accommodate targets entering and leaving the scene. The notations adopted are listed in Table 3.1 before elaborating the detailed information of each major stage.

Table 3.1: Notations adopted in this method.

$\mathbf{X}_k(t)$	a swarm corresponds to target k at time t
$\mathbf{x}^b_{i,k}(t)$	individual best for target k at time t
$\mathbf{x}^g_k(t)$	global best of the swarm for target k at time t
$\mathbf{x}^d_k$	state of the target k given by the object detector
K	number of targets (number of swarms)
N	number of particles for each swarm
i	particle index
n	iteration index
k	target index
t	frame(time) index

3.4.1 Particle and Swarm Diversification

`Particle Diversity:` In order to allow a swarm to track a dynamic optimum (moving target), it is important to maintain particles diversity within the swarm over time. In this method, a swarm $\mathbf{X}_k = \{\mathbf{x}_{i,k}\}_{i=1}^N$ is initialised for every new target entering the scene. Each swarm has N particles where each member $\mathbf{x}_{i,k} = (x_c, y_c, w, h)$ is a potential best state of the pedestrian represented by its centroid location, (x_c, y_c) and the width, w and height, h of the bounding box. These particles are sampled from a Gaussian distribution at the beginning of the PSO iteration at time t as follows:

$$\{\mathbf{x}_{i,k}\} \sim N(\mathbf{x}^{pred}_k(t), \Sigma), i = \{1 : N\} \tag{3.8}$$

where the covariance Σ is a diagonal matrix and its entries are given by $\mathbf{v}_k^{pred}(t)$. The predicted position $\mathbf{x}_k^{pred}(t)$ of the target k at time t is given by:

$$\mathbf{x}_k^{pred}(t) = \begin{cases} 0 & \text{when the swarm is first created.} \\ \mathbf{x}_k^g(t-1) + \mathbf{v}_k^{pred}(t) & \text{otherwise.} \end{cases} \tag{3.9}$$

where the predicted velocity for target k is estimated as:

$$\mathbf{v}_k^{pred}(t) = \mathbf{v}_k^{ind}(t) + \mathbf{v}^{soc}(t) \tag{3.10}$$

where $\mathbf{v}_k^{ind}(t)$ refers to individual velocity of target k and $\mathbf{v}^{soc}(t)$ refers to social velocity of the group. Thus, the motion of a target is predicted based on its personal information $\mathbf{v}_k^{ind}(t)$ and the movement of other members of its social group $\mathbf{v}^{soc}(t)$. Here, the individual velocity for target k is estimated by:

$$\mathbf{v}_k^{ind}(t) = \mathbf{x}_k^g(t-1) - \mathbf{x}_k^g(t-2) \tag{3.11}$$

where $\mathbf{x}_k^g(t-1)$ and $\mathbf{x}_k^g(t-2)$ are states of the target k at time $t-1$ and $t-2$ respectively. Then, the social velocity is computed by sharing information among targets which have been moving generally in the same direction as follows:

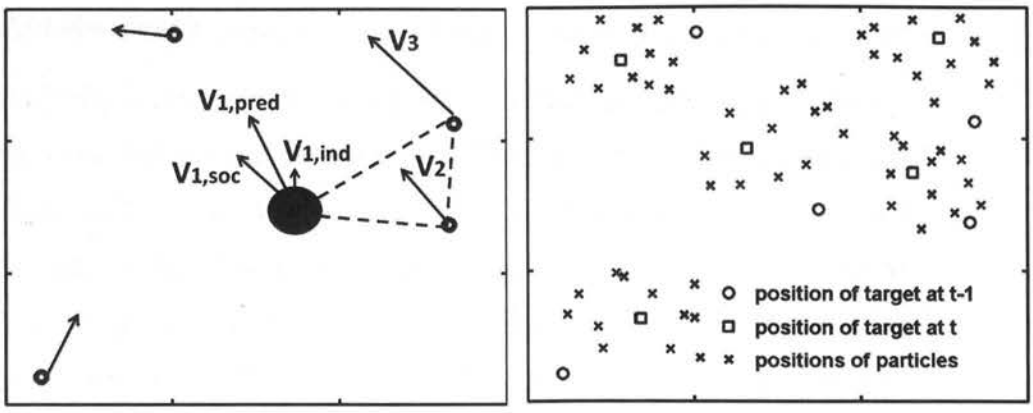
$$\mathbf{v}^{soc}(t) = \frac{1}{K_n} \sum_{j=1}^{K_n} \left(\mathbf{x}_k^g(t-1) - \mathbf{x}_j^g(t-2)\right) \tag{3.12}$$

where K_n is the total number of neighbours of the target k. In this work, two targets are considered as neighbours if they are in close proximity and have similar motion direction and speed for a time-overlap window of Δt frames. More precisely, given a pair of trajectories for target k_1 and k_2 with Δt time-overlap window (in which both targets appear) (in this work, $\Delta t = 10$), the similarity score is computed based on the absolute difference of their position, direction and speed. Two targets are defined as neighbours

when their similarity score is larger than a particular threshold.

It should be highlighted that the motion of a target is predicted based on its personal information as well as the social knowledge among targets (i.e., pedestrians). In this way, the position and motion of an occluded target can be estimated from its social group and particles are distributed at the likely position in the next frame. As a result, this method can recover the target which has been occluded for a period of time. Figure (3.3(a)) shows how the motion of a target is predicted. Dotted lines indicate group membership. Target 1 is being occluded and its motion is estimated from other members of its social group. Figure (3.3(b)) shows the distribution of particles based on its predicted position and motion. The current states of targets are shown with big black circles, while the previous states of targets are shown with lighter black circles. Particles are marked by grey cross symbols.

`Swarm Diversity:` When multiple targets are being tracked, there is a high probability that two targets occlude one another, especially in a crowded scene. This makes two different swarms competing for the same target or to cluster at the same location. To prevent this, the idea of *swarm diversity* is introduced for swarms that are close to each other. The distance between the global best states found by two different swarms are used to decide if two different swarms are competing for the same target or cluster at the same location. When two swarms compete for the same target, the search space of the swarm with the lower fitness value is gradually expanded. As a result, the target can be recovered even after the target has been occluded for a period of time.



(a) $t = t_0$ (b) $t = t_0 + 1, n = 0$

Figure 3.3: Effects of different components of the predicted velocity on initialising particles. Figure(3.3(a) shows how the motion of a target is predicted. Dotted lines indicate group membership. Target 1 is being occluded and its motion is estimated from other members of its social group. Figure 3.3(b) shows the distribution of particles based on its predicted position and motion. The current states of targets are shown in black circles while the previous states of targets are shown in black squares. Particles are marked by black cross symbols.

3.4.2 Swarm Optimisation

In the standard PSO, each particle is a candidate solution and finds the optimum by updating its position based on three components: inertial velocity, cognitive effect and social effect. In this work, a novel PSO updating rule is proposed such that each particle adjusts its speed and position in the search space based on its personal knowledge, a shared information among its own swarm members, and the social activity among swarms. In addition, the detection responses $\mathbf{x}_k^d$ are incorporated in the PSO framework to drive particles to find the new state in the direction given by the pedestrian detector and hence boost the convergence rate.

The proposed velocity and position updating equations for a particle at

time t are defined as follows:

$$\mathbf{v}_{i,k}^{n+1} = \chi[\mathbf{v}_{i,k}^{n} + c_1 r_1(\mathbf{x}_{i,k}^{b} - \mathbf{x}_{i,k}^{n}) + c_2 r_2(\mathbf{x}_{k}^{g} - \mathbf{x}_{i}^{n}) + c_3 r_3(\mathbf{x}_{k}^{d} - \mathbf{x}_{i}^{n})] \quad (3.13)$$

$$\mathbf{x}_{i,k}^{n+1} = \mathbf{x}_{i,k}^{n} + \mathbf{v}_{i,k}^{n+1} \quad (3.14)$$

where $\mathbf{v}_{i,k}^{n}$ and $\mathbf{x}_{i,k}^{n}$ are the velocity and the state of particle i of swarm (target) k at iteration n at time t. Here, the sub-script t is omitted for simplification. The first component $\mathbf{v}_{i,k}$ is the motion prior-based inertial velocity that integrates both individual and social velocity among targets. In contrast to the traditional PSO [77], where the inertial velocity is initialised to zero in the first iteration $n = 0$, this method incorporates the motion prediction based on the individual and the social behaviour of targets as follows:

$$\mathbf{v}_{i,k}^{0}(t) = \begin{cases} 0 & \text{when the swarm is first created } (n = 0) \\ \mathbf{v}_{k}^{pred}(t) & \text{otherwise.} \end{cases} \quad (3.15)$$

where the predicted velocity for the target k is given by equation (3.10). The second component $(\mathbf{x}_{i,k}^{b} - \mathbf{x}_{i,k}^{n})$ corresponds to the cognitive effect where each particle moves to its best known position $\mathbf{x}_{i,k}^{b}$. The third component $(\mathbf{x}_{k}^{g} - \mathbf{x}_{i}^{n})$ is the social effect, where the particle moves towards the global best position $\mathbf{x}_{k}^{g}$ defined by its own swarm.

Compared to the standard PSO [77], this method introduces a new component based on the detection response $\mathbf{x}_{k}^{d}$. This component constrains particles to find the new state in the direction given by a state-of-the-art detector. The details procedures of selecting the detection response will be explained in section 3.4.2.2. The parameters r_1, r_2 and r_3 are random numbers uniformly distributed in $(0, 1]$, generated at every iteration. The parameter $\chi < 1$ confines the velocity of particles within a range and is

defined as: $\chi = 2/||2 - c - \sqrt{c^2 - 4c}||$ where $c = c_1 + (c_2 + c_3)$. The parameters c_1, c_2 and c_3 are positive constants and balance the influence of cognitive, social and detection information respectively. In this method, the value of c_1 is set to $c_1 = (c_2 + c_3) = 2.05$ [105]. This allows each particle of the swarm to use the social knowledge and the detection information collectively but still retains its personal knowledge as independent knowledge. In addition,the parameter $c_3 \in (0, 2.05)$ is set using the normalised matching score between the detection x_k^d and the best state of target k at previous frame $t-1$ such that the influence of the detection information is high only when the selected detection is a good match to the target k.

In the following, the process of selecting the individual best state ($\mathbf{x}_{i,k}^b$), the global best state ($\mathbf{x}_k^g$) and the detection response ($\mathbf{x}_k^d$) for the target k are presented.

3.4.2.1 Identifying Individual and Global Best

The individual best ($\mathbf{x}_{i,k}^b$) and the global best ($\mathbf{x}_k^g$) states of particles are updated at every iteration during the optimisation process by evaluating a fitness (cost) function. In this method, a fitness function is defined based on a localised colour histogram. Given the state of a particle i for target k at time t, the model for the appearance of target k defined by the bounding rectangle $(x_c - \frac{w}{2}, y_c - \frac{h}{2}, x_c + \frac{w}{2}, x_c + \frac{h}{2})$ is described as follow: 1) the target region is first divided into M equal parts (here, M = 9). 2) each part is then represented by a $16H \times 4S \times 4V$ histogram in the HSV colour space. Mathematically, the target model for a given state $\mathbf{x}_i$ is given as $\mathbf{h}(\mathbf{x}_i) = \{\mathbf{h}_m\}_{m=1}^M$ where $\mathbf{h}$ is $16H \times 4S \times 4V$ histogram. Then, the fitness function is defined as:

$$f(\mathbf{x}_i) = \frac{1}{M} \sum_{m=1}^{M} d(\mathbf{h}_m(\mathbf{x}_i), \mathbf{h}_m(\mathbf{x}_k)) \tag{3.16}$$

where $\mathbf{h}_m(\mathbf{x}_i)$ and $\mathbf{h}_m(\mathbf{x}_k)$ are the model and candidate histograms computed at the local part m, M is the total number of parts and $d(\mathbf{h}_m(\mathbf{x}_i), \mathbf{h}_m(\mathbf{x}_k))$ is the distance measure between two histograms. In this work, the quadratic (cross) distance measure [59] is employed to compute the distance between two histograms. The quadratic distance considers the cross-correlation between histogram bins based on the perceptual similarity of the colours represented by bins. The quadratic (cross) distance between histograms $\mathbf{h}_1$ and $\mathbf{h}_2$ is given by

$$d(\mathbf{h}_1, \mathbf{h}_2) = (\mathbf{h}_1 - \mathbf{h}_2)^T \mathbf{A}_{\mathbf{hsv}} (\mathbf{h}_1 - \mathbf{h}_2) \tag{3.17}$$

where $\mathbf{A}_{\mathbf{hsv}} = [a_{ij}]$ is a similarity matrix and a_{ij} gives the similarity between two colours at bins i and j in the HSV colour space [141] and defined as:

$$a_{ij} = 1 - \frac{1}{\sqrt{5}}[(v_i - v_j)^2 + (s_i \cos(h_i) - s_j \cos(h_j))^2 + (s_i \sin(h_i) - s_j \sin(h_j))^2]^{1/2} \tag{3.18}$$

where (h_i, s_i, v_i) and (h_j, s_j, v_j) are two colours at bin i and j.

The next step is to find the individual best and global best state by evaluating the fitness function at different states. A particle updates its current state as the best position (individual best) if the histogram at the current state is more similar to the model (i.e., the fitness value at the current state $\mathbf{x}_i^n$ is lower than the value evaluated at the previous state $\mathbf{x}_i^{n-1}$). Otherwise, the previous best state will be kept. Mathematically,

$$\mathbf{x}_i^b = \begin{cases} \mathbf{x}_i^n, & \text{if } f(\mathbf{x}_i^n) < f(\mathbf{x}_i^b); \\ \mathbf{x}_i^b, & \text{otherwise.} \end{cases} \tag{3.19}$$

Once all particles update their best individual states, the global best among

swarm members is identified as:

$$\mathbf{x}_k^g = \arg\min_{\mathbf{x}_{i,k}^b} f(\mathbf{x}_{i,k}^b) \tag{3.20}$$

where $i = (1, 2, \cdots, N)$ is a member of the swarm for target k.

3.4.2.2 Identifying Detection Response

As explained above, a new component $c_3(\mathbf{x}_k^d - \mathbf{x}_i^n)$ is introduced in equation (3.13) to incorporate a detection response $(\mathbf{x}_k^d)$ in the swarm optimisation process. This term computes the distance between the particle $\mathbf{x}_i^n$ and the associated detection $\mathbf{x}_k^d$ and guides the particles to search the optimum in the region given by an object detector. Unlike the individual best $(\mathbf{x}_{i,k}^b)$ and the global best $(\mathbf{x}_k^g)$ which are updated at every iteration, the state of the detection response $(\mathbf{x}_k^d)$ is fixed during the iteration process. The parameter c_3 tunes the influence of the detection response on the movement of particles. In this method, pedestrian detection for each frame is obtained based on the histograms of oriented gradients (HOG) [48]. Figure 3.4 shows the results of the HOG detector in the monitored scene. Please note that all detection results are not reliable; yielding false positives and missed detections.

Figure 3.4: Sample detection results given by the HOG detector [30].

Given $\{\mathbf{x}_m\}_{m=1}^{K_d}$ detection results at time t by the HOG detector, the next step is to identify a detection response to guide the tracker to a particular target. In order to decide which detection should guide the current tracker, the matching score between detections and the current state of the tracked target k is computed based on the spatial proximity, size and the appearance similarity as follows:

$$A(\mathbf{x}_k, \mathbf{x}_m) = A_s(\mathbf{x}_k, \mathbf{x}_m) \times A_f(\mathbf{x}_k, \mathbf{x}_m) \times A_d(\mathbf{x}_k, \mathbf{x}_m) \tag{3.21}$$

where $x_m \in \{\mathbf{x}_m\}_{m=1}^{K_d}$ is a detection result given by the HOG-based detector [48]. Here, the respective matching score A_s is given by the overlapping area between targets k and the detection m while the score $A_f = \exp\{-d(\mathbf{h}(\mathbf{x}_k), \mathbf{h}(\mathbf{x}_m))\}$ is computed based on the distance between two histograms. Finally, the score A_d is computed using the Euclidean distance between centroid locations of the tracked target k and the detection response m. Next, the detection response, which is the best match to the current state of the tracked target, is identified by finding the maximum matching score:

$$\mathbf{x}_k^{d,t} = \arg\max_{\mathbf{x}_m} A(\mathbf{x}_k, \mathbf{x}_m), m = \{1, 2, \cdots, K_d\} \tag{3.22}$$

where K_d is the number of detections given at time t. The matching score $A(\mathbf{x}_k, \mathbf{x}_m)$ between the selected detection and the tracked target k is normalised and used as c_3, a weighting parameter of detection component in equation (3.13). It can be seen that the matching score or the parameter c_3 will be large only if the selected detection and the current state of the tracked target are highly correlated. This ensures that the output from a detector is integrated in the swarm optimisation only if the selected detection is a good match of the tracked target.

3.4.2.3 Convergence Criteria

As discussed in section 3.3.1, the optimisation process of the standard PSO is terminated when all particles converge at the global best position or the swarm reaches a stable state. However, for tracking application, computation time is an important issue as the objective is to track targets in real or quasi-time. Hence, in this proposed algorithm, the second convergence criterion discussed in section 3.3.1 is modified to stop the optimisation process when a pre-defined fitness value is reached. In short, this method stops the iteration process when one of the following criteria is achieved:

1. $\mathbf{x}_i^{b,n} \approx \mathbf{x}_i^{b,n+1}$ and $\mathbf{x}_i^b \rightarrow \mathbf{x}^g$ for all particles $i \in (1:N)$
2. $\mathbf{x}^{g,n} \approx \mathbf{x}^{g,n+1}$ and $f(\mathbf{x}^g) < TH$
3. the pre-defined maximum number of iterations is reached.

The parameter TH can be defined and updated online by studying the trend of the feature changes of the target k over time. In most experiments presented in this chapter, a good tracking result can be achieved after 5 or 6 iterations.

3.4.3 Swarm Initialisation and Termination

This section describes the swarm initialisation and termination strategy, which accommodates targets entering and leaving the scene.

3.4.3.1 Swarm Initialisation and Learning Target Model

In this method, a new swarm is automatically initialised for each person subsequently detected for T frames. In order to reduce false positive detections, a matching score is computed for each detected target over T.

frames using the equation (3.21). Then, only the associated detections with a matching score higher than the threshold are used to initialise a new swarm. The *HSV* colour histogram of target k $h(x_k)$ at time t is then learnt using the steps discussed in section (3.4.2.1). The length of the observation window T can be determined based on the frame rate of the video and the prior knowledge of the monitoring scene, for instance, T should be set to a low value $(1, 10)$ for a crowded scene where targets enter and leave the scene frequently. Here, the length of preservation window is set to $T = 5$ for the tested video sequences.

3.4.3.2 Updating Target Model

It is important to update the target model $\mathrm{h}(\mathrm{x}_k)$ as the appearance of the target model is expected to have slight variations over time. One solution is to update the target model at every frame (or every T frames) with a new model extracted from the current frame. However, this approach introduces the 'drifting' problem where the target model steadily drifts away from the first model [103]. Another possible solution is to update the target model only when it is observed in isolation, without any occlusion occurring [114, 168].

In this method, the online updating model is proposed to address the appearance variations of targets. Given the previous T appearances (histograms) of the target k, $\{\mathrm{h}(\mathrm{x}_k)(t_1), \mathrm{h}(\mathrm{x}_k)(t_2), \ldots, \mathrm{h}(\mathrm{x}_k)(T)\}$, the minimum appearance change of the target k at time t can be computed as follows:

$$\delta(\mathbf{x}_k(t)) = \min_{t'} d(\mathbf{h}(\mathbf{x}_k(t)), \mathbf{h}(\mathbf{x}_k(\tau))) \tag{3.23}$$

where $\tau = t_1, t_2, \cdots, T$ is a time index for the previous frames and T is a temporal window with a length of $T = 20$ frames. This value gives the smallest

appearance change of target k at time t from the previous observations. Then, the next step is to check if the appearance change of target $\delta(\mathbf{x}_k(t))$ is significantly different from previous appearance changes $\delta(\mathbf{x}_k(\tau))$. The target model should not be updated if there is a significant appearance change in the current frame as this can indicate that the tracker is stuck with the wrong person or the scene condition has suddenly changed. However, if the change is small, the target model must be updated, to accommodate the slight appearance changes.

In this method, the probability density function PDF of the appearance changes of the target, over T previous frames, is estimated using the kernel-based density function as:

$$f(\delta) = \frac{1}{T}\sum_{\tau=1}^{T} K(\delta - \delta_\tau) \tag{3.24}$$

where $K(.)$ is a Gaussian kernel function centred at δ_τ for $\tau = \{1, 2, \cdots, T\}$. Then, the probability score for the appearance change of the target k at time t is computed as:

$$p(\delta_t) = \frac{1}{T}\sum_{\tau=1}^{T} \exp\Big[-(\frac{\delta_t - \delta_\tau}{2\sigma})^2\Big] \tag{3.25}$$

where σ is the standard variation of δ_τ over T frames.[1] The high probability score indicates the slight changes in the appearance target model and hence, the target model $\mathbf{h}(\mathbf{x}_k(t))$ is updated using an adaptive filter as:

$$\mathbf{h}(\mathbf{x}_k(t)) = \alpha\mathbf{h}(\mathbf{x}_k(t)) + (1-\alpha)\mathbf{h}(\mathbf{x}_k(t-1)) \tag{3.26}$$

where $\alpha \in [0, 1]$ is the learning rate. In this work, the learning rate α is

[1]Here, in order to simplify the symbols, $\delta(\mathbf{x}_k(t))$ is represented as δ_t and $\delta(\mathbf{x}_k(\tau))$ is represented as δ_τ.

defined based on the probability score of the appearance change (equation 3.25).

$$\alpha = \begin{cases} p(\delta(\mathbf{x}_k(t))), & p(\delta(\mathbf{x}_k(t))) > th; \\ 1, & \text{otherwise.} \end{cases} \tag{3.27}$$

where th is a threshold that enforces the requirement that the appearance change of the target does not diverge far from the first target model.

3.4.3.3 Swarm Termination

In a scenario where multiple targets enter and leave the monitored scene, it is important to terminate the tracking process when the swarm (tracker) loses its target for a number of subsequent frames. The probability score computed in equation (3.25) indicates the degree of the appearance change of the target at time t from previous frames. The small probability score states that the target has a significant changes from the previous frames or the tracker has lost its target. Based on this observation, a swarm is terminated when the probability score is lower than a particular threshold value for T subsequent frames.

3.4.4 Algorithm Summary

Algorithm 2: Pseudo-code of the proposed algorithm

Input: New image frame at time t and existing trackers: $\{\mathbf{x}_k\}_{k=1}^{K}$
Output: target states at time t $\{\mathbf{x}_k(t)\}_{k=1}^{K}$

Perform HOG detection:$\{\mathbf{x}_m\}_{m=1}^{K_d}$ and
Compute similarity scores (3.21)
for *all targets* **do**
  //Initialisation process
  if *new target* **then**
    Randomly generate a new swarm $\{\mathbf{x}_i\}_{i=1}^{N}$
    Increase total number of trackers K = K + 1
  end
  else
    Find associated detection result: $\mathbf{x}_k^{d,t}$ (3.22)
    Predict target velocity at time t (3.10)
    Re-initialise the positions of particles (3.8)
  end
  foreach *Particle* $i \in 1 \rightarrow N$ **do**
    $\mathbf{x}_{i,k}^{b} = \mathbf{x}_{i,k}^{n}$
    compute the fitness value $f(\mathbf{x}_{i,k}^{b})$
  end
  $\mathbf{x}^{g} = \arg\min_{\mathbf{x}_{i,k}^{b}} f(\mathbf{x}_{i,k}^{b})$
  //Iteration process
  for *n = 1 to maximum number of iterations* **do**
    foreach *Particle* $i \in 1 \rightarrow N$ **do**
      update velocity using equation (3.13)
      update position using equation (3.14)
      if $f(\mathbf{x}_{i,k}^{n}) < f(\mathbf{x}_{i,k}^{b})$ **then**
        $\mathbf{x}_{i,k}^{b} = \mathbf{x}_{i,k}^{n}$
      end
      compute the fitness value $f(\mathbf{x}_{i,k}^{b})$
    end
    $\mathbf{x}^{g} = \arg\min_{\mathbf{x}_{i,k}^{b}} f(\mathbf{x}_{i,k}^{b})$
    if *convergence criteria are met* **then**
      break;
    end
  end
  Output: the global best position: $\mathbf{x}_k(t) = \mathbf{x}_k^{g}$
end

3.5 Experiments

This section presents experimental results in two different contexts. In Section 3.5.1, the performance of the proposed multi-swarm PSO is evaluated against the particle filter and the traditional PSO algorithm. The tracking accuracy and convergence rate of the proposed algorithm is compared against the traditional PSO algorithm. Section 3.5.2 evaluates the task of tracking multiple targets using public surveillance data-sets of crowded scenes with different crowd densities. The proposed method is compared with other state-of-the-art methods in tracking domain. All experiments are performed using Matlab on a platform with a dual-Core $3GHz$ processor and $4GB$ RAM.

3.5.1 Tracking Fixed and Known Number of Targets

The first set of experiments focusses on tracking fixed number of targets where the number of targets are assumed to be fixed and known a priori. In this set of experiments, the detection algorithm is not incorporated. Hence, the results achieved in this set of experiments demonstrate the performance of the proposed multi-swarm algorithm which incorporates interaction among swarms. The objective is to evaluate the improvement that can be obtained in tracking accuracy using the proposed PSO algorithm in the presence of inter-object occlusion. In this set of experiments, the targets are manually initialised in the first frame where the number of particles(N) and the temporal window T is fixed at $N = 15$ and $T = 20$ respectively. The threshold for the learning rate α is set at $th = 0.8$.

3.5.1.1 CAVIAR data-set

The goal of this experiment is to compare the performance of the proposed multi-swarm PSO algorithm and species-based PSO algorithm [168] which is the only PSO-based method evaluated on tracking multiple targets. In order to draw a fair comparison, the same sequences (ThreePastShop2cor.mpg and EnterExitCrossingPaths2cor.mpg from Context Aware Vision using Image-based Active Recognition (CAVIAR) [2]) that were tested in [168] are used in this experiment. The video sequences in this data-set [2] are recorded at 25 frames per second where each video frame is a size of 384×288.

(a) Species (sub-swarm) based PSO [168]. Reported results are extracted from their paper.

(b) Results achieved by the proposed method using HSV colour space and histogram quadratic (cross) distance where the number of particles for each swarm (target) is fixed at $N = 15$. Left to right: Frame 204, 209, 214 and 224.

Figure 3.5: Qualitative comparisons of the proposed method with species (sub-swarm) based PSO [168] on CAVIAR data-set.

Figure 3.5 shows the qualitative results achieved by the proposed method and the species-based PSO where each target is tracked by a sub-swarm. It can be observed that [168] fails to recover the person 'B' after occlusions at frame 211 (Figure 3.5(a)). The authors [168] tackled the problem by incorporating the selective part-based appearance updating model

which updates only the non-occluded parts of the targets. On the other hand, this method, by accounting for social interaction among pedestrians, tracks and recovers targets after occlusion without a need to detect and update the model of non-occluded regions (Figure 3.5(b)). The target model is updated only when the appearance change of the target does not diverge far from the first target model as explained in equation (3.25-3.27).

Figure 3.6: Qualitative Results of the proposed method on CAVIAR data-set [2] using the HSV colour space and histogram quadratic (cross) distance. The number of particles for each swarm is fixed at $N = 15$.

Figure 3.6 shows another example of tracking multiple targets with heavy inter-occlusions. Three persons are successfully tracked even though inter-occlusions occur frequently in frames $450 - 534$. It can be observed

that person 'A' is occluded by person 'B' in frame $453 - 460$ and again occluded by person 'C' who wears the clothes with the similar colour in frame $480 - 510$. However, the results demonstrate that this method, using the proposed swarm diversity scheme and social interaction-based velocity, keeps correct identities throughout the video.

3.5.1.2 Helicopter Sequence

The purpose of this experiment is to validate the proposed approach in a tracking situation with a mobile camera. This method is tested on a Helicopter sequence captured by a mobile camera [8] and let the tracker follows a toy helicopter over 500 frames. Each image has a frame size of 240×320 resolution and shows a remotely controlled toy helicopter with complex dynamics. In this experiment, the number of particles is fixed at $N = 15$ and the target is manually initialised in the first frame.

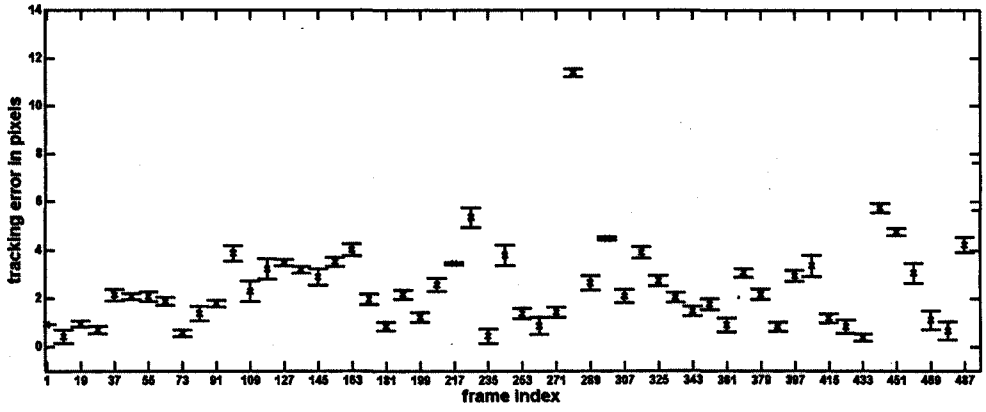


Figure 3.7: Tracking results of the Helicopter sequence using the HSV colour space and the histogram quadratic (cross) distance. The tracking error is shown for every tenth frame and the error is relatively small except at frame 281 where the target is temporarily lost due to the occlusion by the controller.

Figure 3.7 shows the tracking results for the helicopter sequence. The tracking error for every frame is computed as the distance between the

centroid of the target returned by the tracker and the centroid given by the ground truth [8]. The reported results of Figure 3.7 are the mean and standard deviations of 10 runs using the same parameters. It can be observed that the tracking error is relatively small except for frame 281 where the the helicopter is completely occluded by the controller. The target is temporarily lost for frame $281 - 283$, but it can be recovered in frame 284 using the initialisation scheme given in equation (3.15). Some

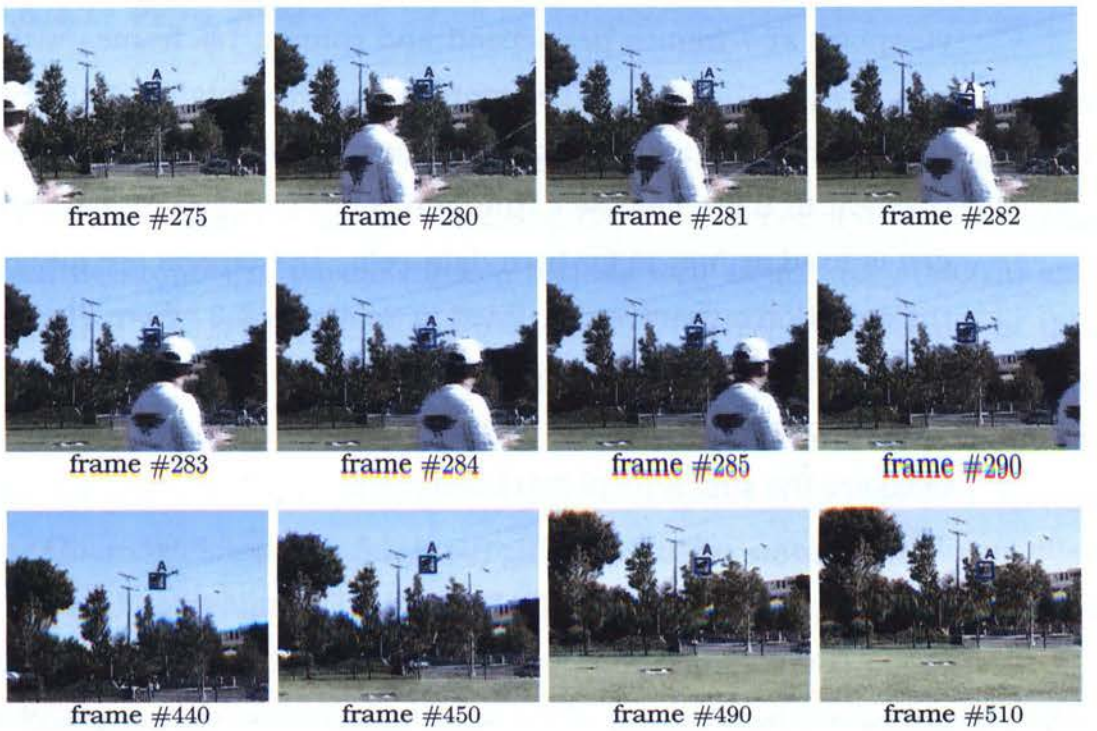


Figure 3.8: Qualitative Results of the proposed method on the helicopter sequence [8] using the HSV colour space and the quadratic (cross) distance. The number of particles is fixed at $N = 15$.

qualitatively results are shown in Figure 3.8. The first two rows show the tracking results under occlusion in frame $275-290$ while the last row shows the results under illumination changes at frames $440 - 450$.

3.5.2 Tracking Unknown and Varying number of Targets

In this set of experiments, the proposed method is evaluated in the context of tracking multiple targets assuming that there is no priori knowledge about the number of targets to be tracked. The objective is to assess the improvement that can be obtained in tracking accuracy. This method is evaluated using two public data-sets: PETS 2009 [9] and the oxford sequence [30]. The PETS video sequence (S2L1) is recorded from an elevated viewpoint at 7 frames per second and contain 795 frames with an image size of 768×576 pixels. The oxford sequence contains 7500 images with a resolution of 1920×1080. Though frame rates, resolutions and densities are different in these data-sets, the number of particles for each swarm (target) is fixed at $N = 15$ for both data-sets. The targets are initialised using the detection output as discussed in Section 3.4.3 where the threshold for the learning rate α is fixed at $th = 0.8$ and the temporal window is set at $T = 10$. In all experiments, the quadratic (cross) distance is employed to measure the similarity of two histograms.

Results are evaluated using the standard Classification of Events, Activities and Relationships (CLEAR) metrics [7]: multiple object tracking precision (MOTP) and multiple object tracking accuracy (MOTA). The MOTP measures the precision of the tracking algorithm on successful detections. Here, MOTP is computed based on the average distance between the centroid positions of tracked targets and the ground truth as in [19]. Given the same detection results, higher MOTP indicates the better precision of a tracking algorithm. The multiple object tracking accuracy (MOTA) measures the tracking accuracy based on false negatives, false positives and identity switches. Hence, this measure combines both the performance of detection and tracking algorithms. In addition, three metrics from [93]: mostly tracked (MT), mostly lost (ML) and partially tracked (PT) are also

used to further evaluate the results. A target is considered as a mostly tracked target if it is being tracked for 80% of the time and mostly track (MT) measures the percentage of successfully tracked targets. Similarly, a target is considered lost if the target is tracked less than 20% of the time and mostly lost (ML) measures the percentage of lost targets. Lower values of ML and PT indicate a good performance of the tracking algorithm.

3.5.2.1 PETS 2009

The ground truth for this sequence is manually annotated by [19] while the detection output is generated by a state-of-the-art HOG detector. Table 3.2 shows the number of mostly tracked (MT), the number of mostly lost (ML) and the number of partial tracked (PT), as well as accuracy (MOTA) and precision (MOTP) for PETS 2009 data-set. The reported results of this method are the averages and standard deviations of 10 runs using the same parameters. For comparisons, the results obtained by the state-of-the-art methods: k-shortest path optimisation-based tracker [31], energy minimisation-based tracking algorithm [19], the extended Kalman filter (EKF) [18] and the occlusion modelling (OM) based tracker [18] are also shown.

Table 3.2: Quantitative comparisons of the proposed method with state-of-the-art methods on PETS 2009 data-set: S2L1 View 1. The results for this method are the average of 10 runs using the same parameters.

Methods	MOTA	MOTP	MT (%)	ML (%)	PT (%)
[31]	79.0%	59.0%	-	-	-
[19]	81.4%	76.1%	82.60%	0.0%	17.40%
Occlusion Model [18]	88.3%	75.7%	86.96%	4.35%	8.70%
EKF [18]	68.0%	76.5%	39.13%	4.35%	56.50%
this method	83.92±4.78%	82.66±1.51%	82.60%	0.0%	17.40%

This method achieves about 80% of tracking precision which is nearly

5% higher than the best reported results. The slightly lower MOTA value (compared to the occlusion model [18]) can be explained by the frequent interactions of targets and the reliability of the resulting detections; i.e., some targets are detected due to persistent false positives occurred by background structures such as signboards and public phone boxes. Figure 3.9 shows some qualitative results on PETS data-set. The first row

frame #35 frame #502 frame #625 frame #661

(a) Results obtained by this method

frame #35 frame #502 frame #625 frame #661

(b) Results obtained by Occlusion Modelling [18]

frame #35 frame #502 frame #625 frame #661

(c) Results obtained by [19]

Figure 3.9: Qualitative Comparisons of the proposed method with state-of-the-art methods on PETS data-set (S2L1). The results for this method is obtained using HSV colour space and the quadratic (cross) distance. The number of particles is fixed at $N = 15$. Results for state-of-the-art methods are extracted from their corresponding papers.

shows the results obtained by this method while the second and third rows show the results from the occlusion modelling (OM) based tracker [18] and

the energy minimisation-based tracking algorithm [19]. Results for [18] and [19] are extracted from their corresponding papers.

3.5.2.2 Oxford sequence

The next experiment evaluates the performance on a town centre sequence [30] recorded at Oxford university. This sequence contains 7500 images with a resolution of 1920×1080. Both the ground truth and the detection outputs for the first 4501 image frames are manually annotated and generated by [30]. In this experiment, this method is compared with two state-of-the-art methods: the stable multi-target tracking [30] and the detector confidence particle filter [34]. The same detection results are used to evaluate the tracking performance of different methods.

Table 3.3: Quantitative comparisons with state-of-the-art methods on the town centre sequence. The results for this method are the average of 10 runs using the same parameters.

Methods	MOTA	MOTP	MT (%)	ML (%)	PT (%)
[30]	79.0%	59.0%	-	-	-
[34]	81.4%	76.1%	-	-	-
this method	82.52±0.16%	80.53±0.13%	76.96%	2.17%	20.87%

Some qualitative results on the town centre sequence [30] are illustrated in Figure 3.10. It can be observed that this method can recover the targets and maintain the same ID after an occlusion (e.g., frame $2128-2169$ on third row). Table 3.3 lists the quantitative results of three different methods. It can be observed that this method achieves about 80% for both MOTA and MOTP and outperforms the state-of-the-art methods.

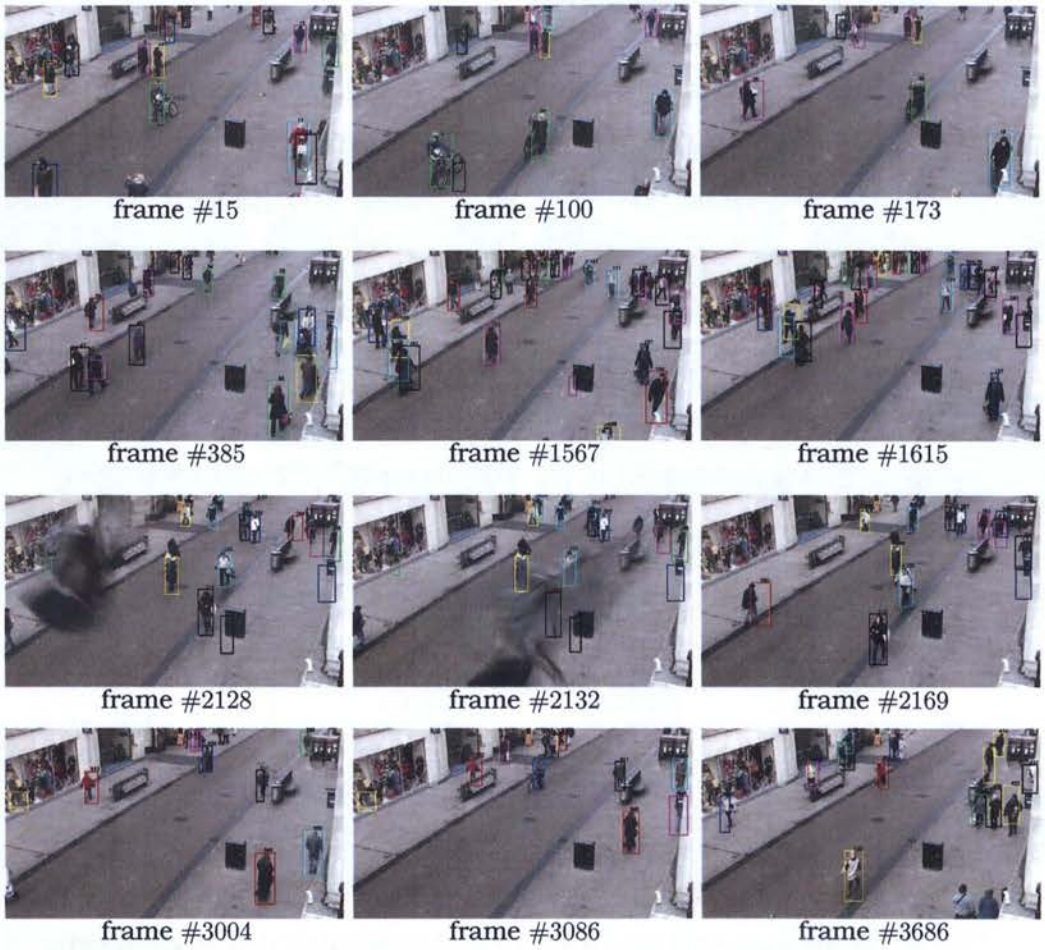


Figure 3.10: Qualitative Results of the proposed method on a town centre sequence from Oxford university obtained using HSV colour space and the quadratic (cross) distance. The number of particles is fixed at $N = 15$.

3.5.3 Performance Evaluation

In this experiment,the performance of this method is evaluated against the number of particles (N) in a swarm. The evaluation is performed using the Helicopter sequence (section 3.5.1.2). The tracking error, the distance between the centroid of the target returned by the tracker and the centroid in the ground-truth, is computed for every frame in the video. The error is then averaged over 5 runs. Similarly, the computational time for each frame is recorded and the time is averaged for the entire video. This process is repeated for different number of particles, $N \in [10, 15, 20, 25, 30, \cdots, 55, 60]$. The results of this evaluation is shown in Figure 3.11.

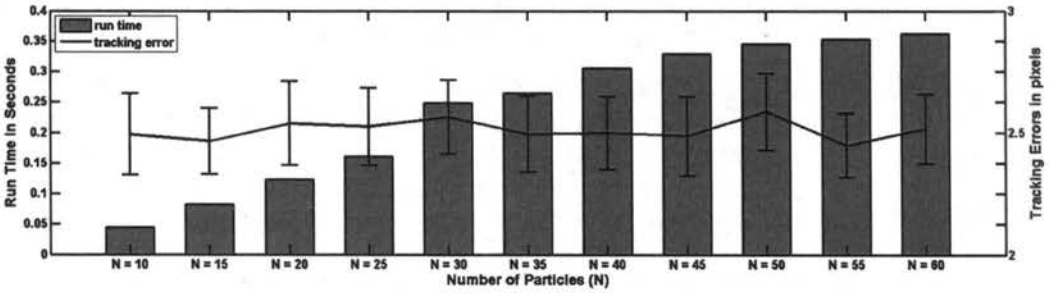


Figure 3.11: Root mean square error and run time against the number of particles on the Helicopter sequence. The average run time (in seconds) for Helicopter sequence is shown on the left axis while the average and standard deviation of tracking error (in pixels) is shown on the right axis.

The means and standard deviations of the tracking errors (shown on the right axis of Figure 3.11) show that the number of particles has only a small influence on the performance for this particular problem. The performance varies only slightly against the number of particles. However, the run-times of the algorithm (shown on the left axis of Figure 3.11) increases as the number of particles increases. It can be observed that this method

takes about 0.08 seconds (about 12 frames per second) while achieving a high tracking precision with an average error of 2.5 pixels for $N = 15$.

Next, the effect of different distance measurement (equation 3.17) on the tracking performance is studied. Evaluations are made using four different distance measurements: normalised Euclidean distance, L1 distance, the quadratic cross distance [59] and Bhattacharyya distance [11]. Figure 3.12 shows the tracking accuracy (MOTA, MOTP and MT) on PETS

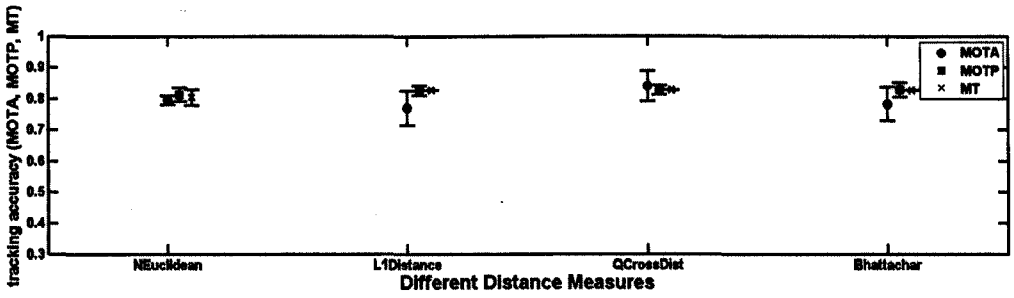


Figure 3.12: Tracking results on PETS 2009$S2L1$ data set using different distance measurements. The averages and standard deviations of MOTA, MOTP and MT of 10 runs are reported. The quadratic cross distance provides the highest MOTA value (83.93% with the standard deviation of 4.78)

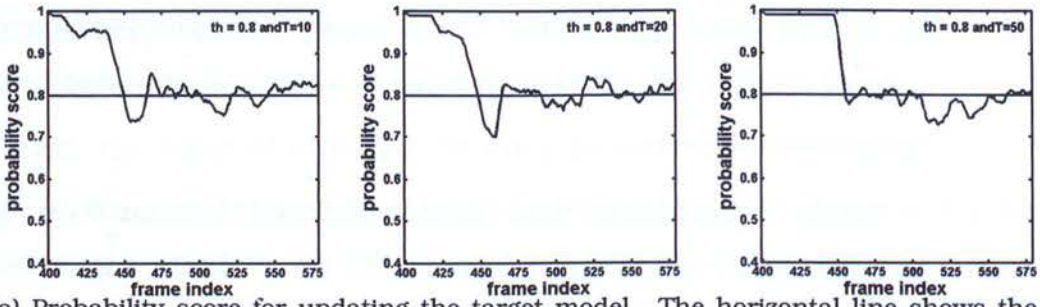
2009$S2L1$ sequence (section 3.5.2.1) using different measurements where the reported results are the average of 10 runs. It is observed that the quadratic cross distance provides the highest MOTA (83.93% with the standard deviation of 4.78) value for PETS 2009 data set where the other measurements obtain about $76 - 80\%$ of MOTA values (79.55%, 76.00% and 78.3% for L1 distance, the quadratic cross distance and Bhattacharyya distance respectively). The MOTA values provided by the normalised Euclidean distance for 10 different runs are peaked around the average and the standard deviations is only about 1.45% where the standard deviations for other measurements are about 5%. The MOTP values is less sensitive to different distance measurements and all four measurements obtain about 82% with

the standard deviations of $1.5 - 2\%$. It can be concluded that this method achieves about 80% MOTA and MOTP values for four different distance measurements.

Finally, the updating model of this method is evaluated using the CAVIAR sequence(section 3.5.1.1). In particular, the evaluation is done on tracking the target 'A' to assess the performance of this method under occlusion. Figure 3.13(a) shows the probability score (equation 3.25) for tracking target 'A' where the length of the temporal window T is set at $T = 10$, $T = 20$ and $T = 50$ respectively. The length of the temporal window T should be long enough to capture the appearance variation of the target model. However, a long temporal window may introduce noises for the probability density function (equation 3.24). For instance, the temporal window $T = 50$ contains the frames $440 - 450$ in which the target 'A' is partially occluded. As a result, the value of the probability scores is high even when the target 'A' is occluded in frames $450 - 463$.

Figure 3.13(b) and 3.13(c) show the tracking results of this method in which the probability density function (equation 3.24) is learnt using the temporal window of $T = 10$ and $T = 20$ respectively. The target model is updated only when the probability score is higher than $th = 0.8$. As can be seen in Figure 3.13(b) and 3.13(c), the target 'A' is tracked successfully under occlusions. However, the tracker fails to track the target 'A' when the probability density function (equation 3.24) is learnt using the temporal window $T = 50$. As discussed earlier, the probability function is no longer representative of the appearance of the target 'A' and hence, the tracker fails to converge to the correct locations as shown in Figure 3.13(d).

Next, the performance is assessed against the selection of the threshold th (equation 3.27). Figure 3.13(e) shows the tracking results using the threshold value of $th = 0.6$ which is lower than the probability scores com-



(a) Probability score for updating the target model. The horizontal line shows the threshold value th.

(b) Tracking Results using $th = 0.8$ and $T = 10$. Left to right: frame #425, frame #455, frame #475 and frame #575

(c) Tracking Results using $th = 0.8$ and $T = 20$. Left to right: frame #425, frame #455, frame #475 and frame #575

(d) Tracking Results using $th = 0.8$ and $T = 50$. Left to right: frame #425, frame #455, frame #475 and frame #575

(e) Tracking Results using $th = 0.6$ and $T = 20$. The target model is updated at every frame. Left to right: frame #425, frame #455, frame #475 and frame #575

Figure 3.13: A qualitative comparison of update strategies using CAVIAR sequence. The first row shows the probability scores for updating the target 'A' using different lengths of temporal window, from left to right: $T = 10$, $T = 20$ and $T = 50$. The second, third and fourth rows show the tracking results using $th = 0.8$ and $T = 10, T = 20$ and $T = 50$ respectively. The last row shows the tracking results using $th = 0.6$ (in other words, the target model is updated at every frame) and $T = 20$.

puted for all frames (please refer to Figure 3.13(a)). Since the threshold value th is very small, the target model is updated at almost every frame, even during the occlusion at frames $440 - 455$. As a result, the tracker fails to keep the correct identity of the target 'A' after the occlusion at frames $475 - 575$.

3.6 Summary

The work presented in this chapter addressed the problem of tracking a variable number of interacting targets in a complex scene. The standard PSO algorithm is extended by introducing an idea of multiple swarms where each swarm tracks an individual target. The proposed method incorporates a number of constraints into the PSO algorithm. Through particles and swarms diversification, motion prediction and interactions among targets are introduced, constraining swarm members to the most likely region in the search space. The output from a pedestrian detector are also incorporated into the velocity-updating equation of the PSO algorithm.

Qualitative and quantitative results for the fixed number of targets are presented in section 3.5.1. Experimental results indicate that the proposed multi-swarm PSO

- is able to track multiple targets in a complex scene with severe occlusion and heavy interactions among targets,
- achieve a better tracking accuracy (above 80% for both MOTA and MOTP) and a faster convergence rate and
- requires fewer particles to adequately track the target over time.

Section 3.5.2 presents results for a varying number of targets in a complex scene. The quantitative comparisons given in Table 3.2 and 3.3 show

that the proposed tracker using a PSO algorithm outperforms the state-of-the-art tracking algorithms. The proposed method is able to correctly detect entering and leaving targets and track targets over time, maintaining a correct, unique identification. Given the same detection outputs, this method achieves a better tracking accuracy and the results indicate that the proposed PSO algorithm can successfully track multiple targets in a complex scene.

"Everybody is a genius. But if you judge a fish by its ability to climb a tree, it will live its whole life believing that it is stupid. "

Albert Einstein

4

Abnormality Detection in Crowded Scenes

The previous chapter presented a particle swarm optimisation framework for tracking individual targets. This method has been proved to be useful for tracking a few individuals in a complex scene and analyse their behaviour. However, as the number of people in the scene increases, the problem of tracking individual targets becomes more challenging. As a result, tracking-based approaches are inadequate for analysing behaviours of a crowd. To address this limitation, in recent years, researchers have proposed to monitor the behaviour of a crowd without identifying the locations and actions of individuals participated in the crowd event.

This chapter presents two novel methods for detecting and localising abnormal regions in a crowded scene.

4.1 Introduction

Abnormality detection refers to the detection of unusual behaviour of individuals, or a group in a crowded scene. For instance, an abnormal event can be a panic or a fight event in a crowd where most people in the scene

suddenly change their behaviour. Several methods [17, 104, 147] have employed frame-based properties to detect the sudden motion of a crowd in the monitored scene. For instance, [17] uses the dense optical flow in the whole frame to learn the regular movement of a crowd where [104] models the normal pattern of an interaction force between pedestrians based on optical flow and the particle advection method. The emergency events such as a sudden fall of a person in the crowd are successfully detected. However, an abnormal event can also arise due to an unexpected action of an individual in a crowd. For example, a running person in a crowd can indicate an abnormal event, if the rest of the crowd is moving at a walking pace. Hence, local motions of the crowd becomes an important cue to detect and localise individual behaviours that deviate from the rest of the crowd's dynamics.

This chapter presents techniques developed for detecting and localising abnormal regions in a crowded scene. The proposed approaches aim not only to detect abnormal activities both in the local and global context but also for an accurate localisation of regions having abnormal or unknown behaviour. Section 4.2 presents the first approach that focusses on the use of manifold learning algorithm for global abnormality detection, where participants in the crowd behave collectively. The idea is to exploit temporal coherence between video frames and use a manifold learning algorithm, for instance Laplacian Eigenmaps [28], to discover different crowd activities from a video. This method provides an advantage of visualising and identifying different crowd events in a low dimensional space and detect abnormality. Section 4.3 presents a novel approach for detecting and localising abnormal activities in crowded scenes. This approach emphasises on local abnormality detection where the behaviour of an individual deviates from the rest of the crowd in a particular time instance.

4.2 Global Abnormality Detection

This section describes a novel approach that analyses video events in crowded scenes. A manifold learning method[1] is proposed to achieve visualisation and modelling of video events in a low dimensional space. This low-dimensional representation preserves the spatio-temporal property of a video as well as the characteristic of the video. Different tasks of video content analysis such as visualisation, video event segmentation and abnormality detection are achieved by analysing these video trajectories based on the Hausdorff distance similarity measure [3].

4.2.1 Frame-based Video Representation

This method begins with an introduction of representing video frames using local motion information. Each frame of a video is represented using a histogram of the optical flow defined as follows. The optical flow vectors are first computed over a m by n cells between two successive frames using the method proposed in [37]. The distribution of optical flow in each cell is then represented using a weighted histogram of B bins, where weight in each bin corresponds to the magnitude of optical flow in one particular direction. Then, a representation of a frame is obtained by concatenating all histograms of cells into a long vector. Mathematically, it is defined as:

$$\mathbf{x} = [\mathbf{h}_k], \quad k = \{1, 2, 3, \ldots, K\} \tag{4.1}$$

where k is an index of each cell, $K = m \times n$ is total number of cells in each frame and $\mathbf{h}_k$ is a weighted histogram of B bins ($B = 4$) for a cell k.

[1]The background theory of different manifold learning algorithms, such as Gaussian process latent variable model (GPLVM), isometric feature mapping (ISOMAP), Local Linear Embedding (LLE)and Laplacian Eigenmaps (LE), are provided in Appendix A.

4.2.2 Spatio-Temporal Laplacian Eigenmaps

The first step of Laplacian Eigenmaps is to compute a weighted neighbourhood graph. In this method, a weighted neighbourhood graph is computed based on spatio-temporal relations where the weight of the edge connecting two video frames, i and j are computed as follows:

$$\omega_{ij} = \omega_s + \omega_t \tag{4.2}$$

where $\omega_s = \exp(-d_s/\sigma_s)$ is the spatial weight computed based on the feature distance. The parameter, $\sigma_s \in [0,1]$ a feature scale parameter that defines the influence of neighbour points and is defined empirically. The weight, $\omega_t = \exp(-d_t/\sigma_t)$ is computed based on temporal information between frames where $d_t \in [0,1]$ is defined as a normalised time difference between frames. That is the value d_t will be zero between two adjacent frames while d_t will be one for two farthest frames. The temporal scale parameter, $\sigma_t \in (0,1)$ decides the influence of the temporal neighbours[2]. In this way, this method preserves spatial weights for all for all pairs of images in a video and provides additional weight for adjacent frames in temporal domain.

The feature distance d_s represents the dissimilarity measure between two frames and it is defined as the weighted sum of distance measures between corresponding cells. Mathematically, it is defined as:

$$d_s(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{K} \alpha_k \times d(\mathbf{h}_k^i, \mathbf{h}_k^j) \tag{4.3}$$

where K is total number of cells in an image and α_k is the weight for each

[2]In the reported experiments, the parameter for spatial and temporal information σ_s and σ_t are empirically selected as 0.5 and 0.2 respectively.

position. The parameter α depends on the prior knowledge of the scene. For example, α should be zero for the background position. Here, it is assumed that there is no prior information about the scene, and hence the value is fixed to 1 for all cells. The distance, $d(\mathbf{h}_k^i, \mathbf{h}_k^j)$, can be any distance measure between two histograms of corresponding locations in frames, i and j. In this method, the distance measure between two histograms is computed as follows:

$$d(\mathbf{h}_k^i, \mathbf{h}_k^j) = 1 - \frac{\mathbf{h}_k^i \cdot \mathbf{h}_k^j}{\|\mathbf{h}_k^i\| \|\mathbf{h}_k^j\|} \tag{4.4}$$

where $\mathbf{h}_k^i$ refers to the vector of weighted histogram for cell k from frame i and $\mathbf{h}_k^j$ refers to the vector of corresponding location from frame j. When all edges are assigned with appropriate weights, the low-dimensional embedding space of the video is computed by minimising the following cost function:

$$\phi(\mathbf{Y}) = \sum_{ij}^{M} \omega_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2 \tag{4.5}$$

where M is total number of images in the training data-set. The minimum of this cost function is given by the eigen-decomposition and the low dimensional representation of video frames is obtained by using the Laplacian Eigenmaps method.

The embedding process maps each video frame into a low dimensional vector, $\mathbf{y}_i$. The number of dimensions for the subspace is selected based on the relative difference between two adjacent eigen-values. The temporal order of video frames defines a path in the embedded space. As a result, the above step transforms a video segment of T frames into a set of T points in the low-dimensional space, where each point corresponds to the successive frames of the original video.

4.2.3 Analysing Video Manifolds in Temporal Domain

The proposed ST-LE discovers the internal structure of the video and produces a trajectory for each video sequence. Given a trajectory in the embedded space, i.e. a set of sequential data points or image frames in a video, the next step is to analyse these data points for different problems of video understanding. In addition, given that some video frames contain labelled information, the crowd events happening in the rest of the video can be identified. In order to incorporate the temporal smoothness, a trajectory is considered for each data point as $\mathrm{S}_i = \{\mathrm{y}_1, \mathrm{y}_2, \cdots, \mathrm{y}_T\}$, where T is the length of a temporal window and y_i is the low-dimensional representation of a video frame. That is, the information from temporally adjacent T frames is considered in determining the possible event of a new frame. Assuming that there are K_e video events and each event can be represented by a single Gaussian, the event happening in a new video frame can be computed as:

$$k_e = \arg\max_{k_e} P(\mathrm{S}_{new}/\mu_{k_e}\Sigma_{k_e}) \tag{4.6}$$

where $P(\mathrm{S}_{new}/\mu_{k_e}\Sigma_{k_e})$ is the probability of the new video segment belonging to the crowd event k_e where $k_e \in (1, K_e)$ is the index of the interested crowd events and K_e is the total number of interested crowd events. The parameters for crowd event, μ_{k_e} and Σ_{k_e}, are learnt using the labelled information in the embedded space.

Similarly, video frames containing abnormal activities can be identified for a given video sequence, where abnormal events are rare and dissimilar from regular instances. This problem can be seen as a two-class clustering problem, where only one class (normal class) has labelled information. To address this problem, video trajectories corresponding to normal frames are modelled using a Gaussian mixture model where the number of Gaus-

sian components C is determined using the Bayesian information criterion (BIC) [134]. Then, the normality score for each new video segment S_{new} is computed as follows:

$$p(S_{new}) = \sum_{r=1}^{C} \omega_r \frac{1}{\sigma_r \sqrt{2\pi}} \exp\left(-\frac{(S_{new} - \mu_r)^2}{2\sigma_r^2}\right) \tag{4.7}$$

where μ_r, σ_r and ω_r are the mean, variance and weight of the r Gaussian component learned using the labelled information. Based on a fixed threshold on the normality score, each segment is labelled as normal or abnormal. Here, the range of the threshold value is set to be in the [0:1; 0:9] to generate multiple sets of true positive and false positive rates and the best threshold can be selected where the minimum equal error rate is obtained.

4.2.4 Experimental Results

4.2.4.1 Segmenting a Video Trajectory

The first experiment evaluates the performance of this method for recognition and identification of crowd events using video sequences from the PETS data set [9]. This data set consists of four video sequences (768×576, 7 frames per second) with time stamps $14-16$, $14-27$, $14-31$ and $14-33$. These video sequences contain one or more of the following crowd events: walking, running, evacuation (rapid dispersion), local dispersion, crowd forming and splitting. The ground truth is generated by manually labelling each video sequence into different crowd events based on the definition provided in [56] (please refer to Table 4.1). In addition, one additional crowd event labelled as "local movement" is defined to represent a crowd with small movements. Then, one-third of labelled frames are randomly

selected for training and the rest are used for testing.

Event	Description	Video Frames [start frame: end frame]
crowd splitting	split to 2 or more directions	T:14-31[51-130]
crowd forming	merge of individuals	T:14-33[0-196]
walking	most moving at low speed	T:14-16[0-36, 108-161], T:14-31[0-50]
running	most moving at high speed	T:14-16[37-107, 162-223]
local dispersion	localised rapid movement	T:14-27[0:184, 280:333]
evacuation	rapid dispersion	T:14-33[340-377]
local movement	little or no movement	T:14-33[197-339] T:14-27[185:279]

Table 4.1: Ground Truth for the PETS data set [9].

Table 4.2 shows the confusion matrix for the recognition of crowd events. The reported results are averaged results over ten runs using randomly selected training sets. It can be observed that this method achieves promising recognition accuracy rate. Figure 4.1 and 4.2 show some qualitative results on crowd events recognition. The probabilities of predefined events occurring in the frame are given while the event with the highest probability is highlighted with a white bounding box.

Event	split	form	walk	run	eva.	local disp.	local mov
split	0.98 ± 0.0006	0	0.02 ± 0.006	0	0	0	0
form	0	0.84 ± 0.004	0	0	0	0	0.16 ± 0.004
walk	0.01 ± 0.0002	0	0.64 ± 0.02	0.35 ± 0.02	0	0	0
run	0	0	0.12 ± 0.038	0.88 ± 0.038	0	0	0
eva.	0	0	0	0	0.99 ± 0.0005	0	0.01 ± 0.0005
local disp	0	0	0	0	0	0.98 ± 0.0007	0.02 ± 0.0007
local mov.	0	0.022 ± 0.0023	0	0	0.019 ± 0.00	0.023 ± 0.0001	0.94 ± 0.0015

Table 4.2: Confusion matrix for crowd event recognition on four video sequences from PETS 2009 data-set (T $14-16$, $14-27$, $14-31$ and $14-33$).

Table 4.3 compares recognition error rates obtained by the proposed method and the state-of-the-art methods. Please note that there are no reported results for local movement event in [56] and [40]. It can be seen that this method has a better recognition accuracy for events such as splitting, forming, evacuation and local dispersion while [56] and [40] have better performance for recognising crowd walking and running event. This can be explained by the nature of this method in learning feature similarity

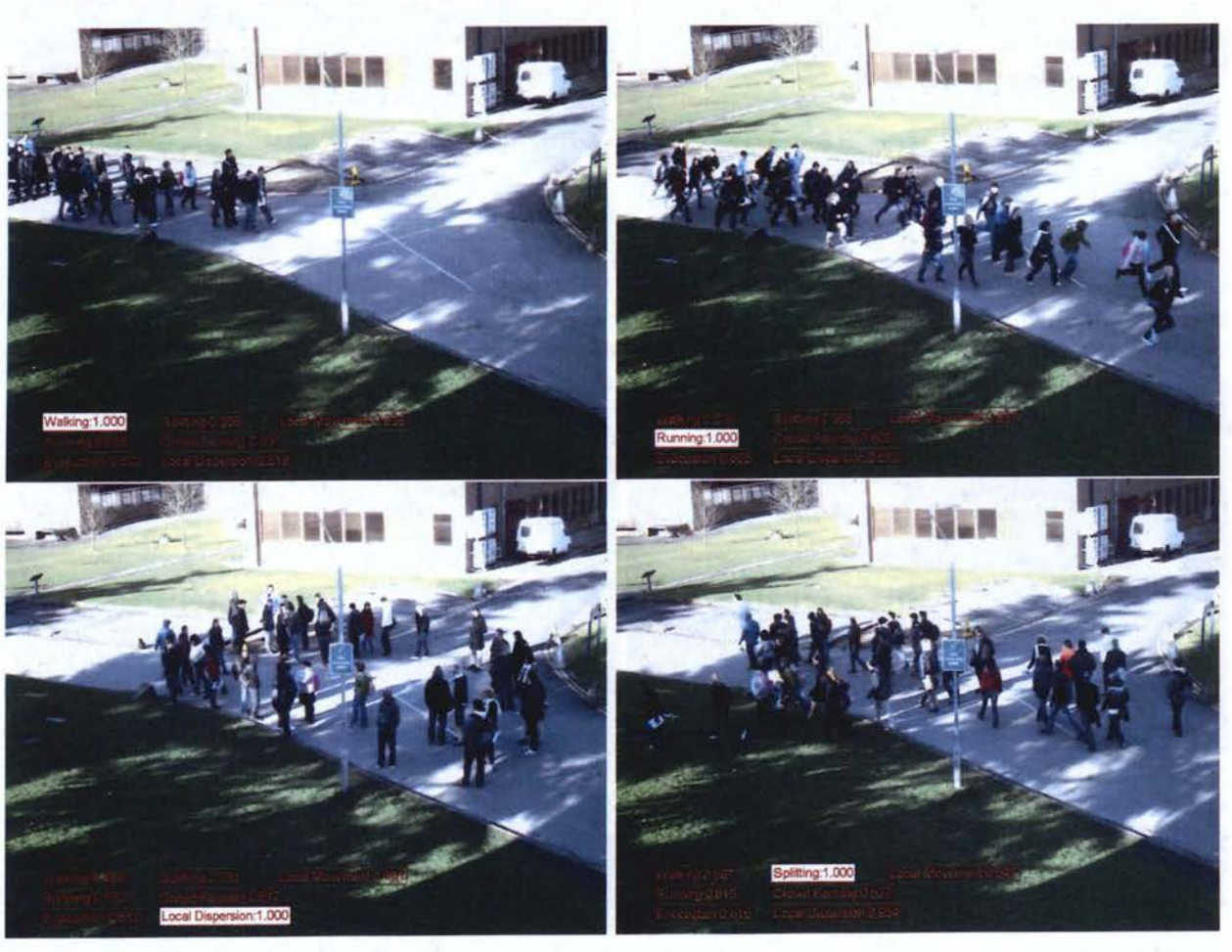


Figure 4.1: Some qualitative results for crowd event recognition on four video sequences from PETS 2009 data-set (T14 – 16, 14 – 27, 14 – 31 and 14 – 33).

Figure 4.2: Some qualitative results for crowd event recognition on four video sequences from PETS 2009 data-set (T14 – 16, 14 – 27, 14 – 31 and 14 – 33).

Event	[56]	[40]	proposed method (without temporal)	proposed method (wt temporal cons.)
split	0.21	0.33	0.17 ± 0.0023	$\mathbf{0.02 \pm 0.0006}$
form	0.40	0.31	0.04 ± 0.0187	$\mathbf{0.16 \pm 0.0044}$
walk	0.08	0.02	0.49 ± 0.0118	$\mathbf{0.36 \pm 0.0212}$
run	0.08	0.03	0.00 ± 0.0036	$\mathbf{0.12 \pm 0.0381}$
eva.	0.03	0.10	0.01 ± 0.0005	$\mathbf{0.01 \pm 0.0005}$
local disp	0.23	0.15	0.02 ± 0.0001	$\mathbf{0.02 \pm 0.0007}$
local mov.			0.42 ± 0.0035	$\mathbf{0.06 \pm 0.0015}$
avg.	0.17	0.16	0.16 ± 0.0058	$\mathbf{0.11 \pm 0.0096}$

Table 4.3: Comparison of the proposed method and the state-of-the-art methods on PETS 2009 data-set.

automatically. In contrast to [56] and [40], where the walking and running events are classified based on a manual threshold, this method learns the feature similarity automatically by finding the different between two histograms. It is observed that when the walking and running events happen continuously (adjacent in temporal domain) as in the provided data set, video frames in the transition period are projected closely in the manifold space and leads to misclassification. On average, this method provides the lowest error rate.

Results obtained without using the temporal information are also reported in Table 4.3 (please refer to the third column). It is observed that this method achieves comparable performance with the state-of-the-art methods without incorporating temporal information. A significant improvement on recognition accuracy is obtained when temporal information is incorporated. It can be observed that the temporal constraint, in general, improves the classification accuracy but not for the running event. This is because in this particular data set, the transition period adds ambiguity between these two events and drops recognition accuracy for the running event. Please note that Caroline et al. [56] reported results obtained using different types of classifiers and their best reported results

are selected for comparison.

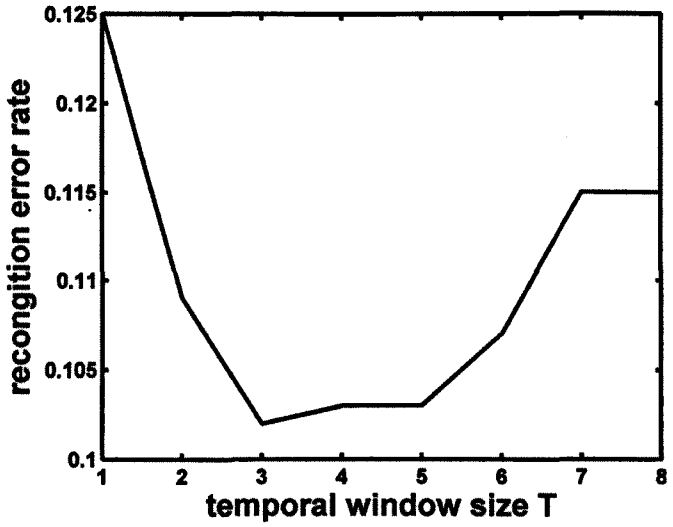


Figure 4.3: Error rate vs. temporal window size for crowd event recognition using PETS 2009 data-set.

Next, the influence of the size of the temporal window T on the performance of the method is evaluated. Figure 4.3 shows the average recognition error rate for different window sizes. It can be observed that the lowest error range is achieved with a window size of $T \in [3, 6]$ as shown in Figure 4.3. In this experiment, the event recognition accuracy is computed for each individual frame in order to compare with the state-of-the-art methods which used frame-based approach. For instance, when a segment $S_i = \{y_{i-T/2}, \cdots, y_i, \cdots, y_{i+T/2}\}$ is recognised as a walking event, the middle frame y_i is considered as a walking event. As a result, a longer window size can cause ambiguity when finding the exact location of the event frame. This leads the recognition accuracy of the experimental results to decrease when a very long window size is considered.

4.2.4.2 Abnormality Detection

The second experiment validates the performance of this method on abnormality detection in crowded scenes using the crowd activity data set from University of Minnesota [6]. This data set contains eleven video sequences recorded in three different indoor and outdoor scenes (2 sequences for the first scene, 6 sequences for the second scene and 3 sequences for the third scene). Figure 4.4 shows some sample frames of these different scenes. Each video sequence contains about 500 frames with a normal

(a) Sample frames from scene 1

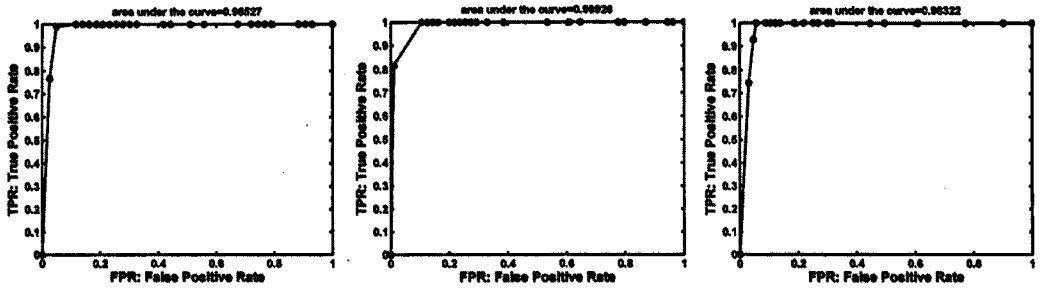
(b) Sample frames from scene 2

(c) Sample frames from scene 3

Figure 4.4: Sample frames from three different scenes of UMN data-set. Each row shows sample frames from different scenes.

starting section and abnormal ending section where each frame has a size of 320×240 pixels.

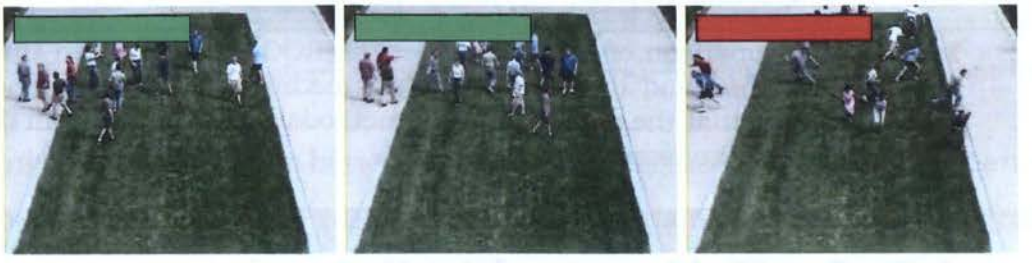
This data set is first divided into the training set and the testing set. The training set contains one video sequence of the first scene, two sequences of the second scene and one sequence of the third scene. The normal behaviour for each scene is learnt separately. The testing set contains one sequence of the first scene, four sequences of the second scene and two sequences of the third scene. The training set contains only the frames with normal activities while the testing set includes frames with both normal and abnormal activities.



(a) ROC curve for scene 1 (b) ROC curve for scene 2 (c) ROC curve for scene 3

Figure 4.5: ROC curves for abnormality detection for three different crowded scenes from UMN data-set.

Figure 4.5 shows the ROC curves for three different scenes, while Figure 4.6 shows the normal frames and abnormal frame of the corresponding video sequence. The green and red bars on the left corner of the frame indicate normality and abnormality respectively. A comparison of the area under the ROC curve between the proposed method and the state-of-the-art methods is shown in Table 4.4. In this experiment, the range of the threshold value is set to be in the $[0.1, 0.9]$ range for computing ROC curve. It can be seen in Table 4.4 that the experimental results either match or exceed the performance of existing state-of-the-art approaches.



(a) Sample images for normal and abnormal frames from testing data-set (scene 1). From left to right: frame index 35, frame index 140 and frame index 575.

(b) Sample images for normal and abnormal frames from testing data-set (scene 2). From left to right: frame index 190, frame index 240 and frame index 432.

(c) Sample images for normal and abnormal frames from testing data-set (scene 3). From left to right: frame index 40, frame index 250 and frame index 501.

Figure 4.6: Qualitative results for abnormality detection for three different crowded scenes from UMN data-set. Each row represents the results for a video sequence of different scenes. The green and red bars on the left corner of the frame indicate normality and abnormality respectively.

Method	scene 1	scene 2	scene 3	average
Cong *et al.* [45]	99.5%	97.5%	96.4%	97.8%
Shi *et al.* [138]	93.6%	77.5 %	96.6%	89.2%
Social Force Model [104]	-	-	-	96.0%
Optical flow [104]	-	-	-	84.0%
Proposed method	98.5%	96.9%	98.3%	97.9±0.76%

Table 4.4: Comparison of the area under the ROC curve between the proposed method and the state-of-the-art methods on UCM data-set. Please note that the results of other methods are extracted from their corresponding papers.

4.3 Local Abnormality Detection

The approach presented in the previous section analyses videos of crowded scene using the spatio-temporal Laplacian Eigenmaps method. The pairwise graph was constructed between video frames in the temporal domain. This approach demonstrates that the use of the manifold learning method leads to a better performance for visualising and detecting temporal events in crowded scenes. However, the use of global frame features prevents the previous approach to detect localised abnormal activities. In this section, a new approach that captures the spatial and temporal variations of local motions of a crowded scene is presented. This approach is composed of two major stages: a training stage and a testing stage. During the training stage, Laplacian Eigenmap based approach is employed to extract different crowd activities from the video. Then, a model of regular crowd behaviour is learnt based on the magnitude and direction of local motion vectors extracted from different crowd activities. Next, the learnt model is used to detect and localise regions having abnormal or unknown behaviour in a new testing video.

In this approach, a video is represented as a fully connected graph $G(V, E)$ where V is a set of local regions and E is a set of edges represent-

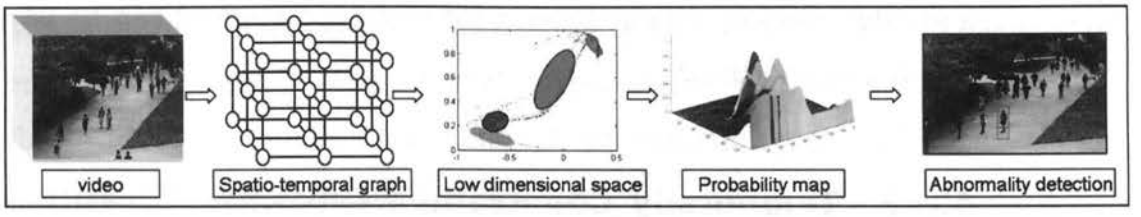


Figure 4.7: A diagram illustrating the overall flow of the proposed method.

ing the connectivity between these local regions. The weight of an edge between nodes (local regions) is computed using the similarity in feature, space and temporal domain between the nodes. This graph provides the global correlations of the local motions in the video and the spectral analysis on this graph yields the dominant eigenvectors as the coordinates of the embedding space. In the embedded space, the distribution of local motions is learnt to extract different crowd activities. Based on the representatives of these activity patterns, the regular activity of a crowd is represented using a local probability model. The overview of the method is illustrated in Figure 4.7.

4.3.1 Representation of Local Motion

This approach represents a video as a set of local regions, by subdividing the video into non-overlapping cuboids of a fixed size ($m \times n \times T$). The optical flow vectors, computed in each patch ($m \times n$), are then quantised into four different directions. The weight for each bin of the histogram is the magnitude of the optical flow in the corresponding direction. Hence, for each local region at time t, the motion information is represented as a histogram, $\mathbf{h}_t = [h^1 h^2 h^3 h^4]$. Next, histograms at each local region over a period of T frames are concatenated as $\mathbf{x}_i = \{\mathbf{h}_i\}_{t-\frac{T}{2}}^{t+\frac{T}{2}}$. The main motivating factor for concatenating these motion vectors instead of averaging is to incorporate motion consistency over time. For instance, a local region with

a sudden motion change (even with a short duration) can be captured by this representation.

4.3.2 Temporally Constrained Laplacian Eigenmaps

This approach considers the relation of local regions based on their joint similarity in feature, space and temporal domain. Mathematically, the weight for an edge connecting two local regions is defined as:

$$\omega_{ij} = \exp\left(\frac{-d_f^{ij}}{\sigma_t^i \times \sigma_t^j}\right) \cdot \exp\left(\frac{-d_s^{ij}}{\sigma_s}\right) \tag{4.8}$$

The first term in the above equation yields the feature similarity between local regions based on the visual context. The feature distance d_f is defined as:

$$d_f^{i,j} = \alpha^{i,j} \times d_f(\mathbf{x}_i, \mathbf{x}_j) \tag{4.9}$$

where $\alpha^{i,j}$ is the cosine distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ while $d_f(\mathbf{x}_i, \mathbf{x}_j)$ is the distance based on the histogram intersection. The idea is to consider both direction and magnitude of optical flow vectors in measuring distance between two feature vectors. The sigma values, σ_i^t and σ_j^t are defined as local variances at r_i and r_j respectively. These local variances are computed using δ_i number of temporal neighbours which have feature distances $d_f^i(t, t+\delta_i) \leq d_f^i(t, t+1)$. Here, $d_f(t, t+\delta)$ is the distance between two feature vectors for the region i, at time t and $t+\delta$ while $d_f(t, t+1)$ is the feature distance between two adjacent temporal neighbours at time t and $t+1$.

The second term in equation (4.8) yields the spatial similarity where d_s is the Euclidean distance between the centroid locations of two local regions. The sigma value σ_s is the spatial scaling factor that controls the amount

of spatial information while considering the manifold structure. Here σ_s is empirically selected. Now, all edges are assigned with appropriate weights, the next step is to find a low-dimensional embedding space by minimising the following cost function:

$$\phi(\mathbf{Y}) = \sum_{ij}^{M} \omega_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|^2, \tag{4.10}$$

where M is total number of local regions in the training set, ω_{ij} is given by equation (4.8) and $\mathbf{Y}$ is the low-dimensional embedding of the entire video. Each entry, $\mathbf{y}_i \in \mathbf{Y}$ is the low-dimensional representation of a local region. It turns out that the minimisation problem is equivalent to finding the optimum $\mathbf{Y}$:

$$\mathbf{Y}_{opt} = \arg\min_{\mathbf{Y}} (\mathbf{Y}^T L \mathbf{Y}) \quad \text{subject to } \mathbf{Y}^T \mathbf{D} \mathbf{Y} = 1 \tag{4.11}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian graph matrix. $\mathbf{D}$ is the diagonal weight matrix in which each entry is a total sum of each row of the weight matrix, $\mathbf{W}$, and computed as $d_{ii} = \sum_j \omega_{ij}$. The solution is provided by the matrix of eigenvectors corresponding to the smallest k_s non-zeros, eigenvalues of the generalised eigenvalue problem $\mathbf{L}\mathbf{y} = \lambda \mathbf{D}\mathbf{y}$. Similar to the approach discussed in Section 4.2, k_s is automatically selected based on the relative difference between two adjacent eigenvalues.

4.3.3 Representation of Regular Motion Pattern

The above process embeds local motion patterns into different spatial locations where similar patterns are usually close and different patterns are far apart. This allows us to cluster embedding points and discover different motion patterns in the monitored scene. Assuming that abnormal

instances are rare and dissimilar from regular instances, the cluster with small data points or outliers in the embedded space can be considered as abnormal instances. However, clustering results in the embedded space do not directly provide a way to detect abnormality in an unseen video.

In this method, the regular behaviour of a crowd is modelled using the clustering results obtained in the embedded space. In other words, each local region of the monitored scene is represented by an expected motion information learnt during the training process. In order to find the regular motion patterns, the embedded data points are first clustered using the state-of-the-art clustering algorithms. Here the k-means algorithm is employed for clustering where the number of clusters is automatically decided by the method in [165]. In addition, those clusters with the size smaller than a particular threshold[3] are also removed.

Next, clustering results are employed to represent regular motion patterns. First, each group of local motion patterns is represented as a single Gaussian, $N(\mu_k, \sigma_k)$ where the parameters μ_k and σ_k are computed as follows:

$$\mu_k = \frac{1}{N_k}\sum_{i=1}^{N_k} \mathbf{x}_i, \quad \sigma_k = \frac{1}{N_k}\sum_{i=1}^{N_k}(\mathbf{x}_i - \mu_k)^2 \tag{4.12}$$

and N_k is the total number of data points belonging to the k^{th} group.[4] Then, the regular behaviour of a scene is modelled as a multiple single-Gaussian and the weight for each Gaussian is computed based on the co-occurrence of activities at a given location over time:

$$\omega_{k,i} = \frac{n_{k,i}}{\sum_{\varepsilon} n_{\varepsilon,i}}, \text{ where } \varepsilon = \{1, 2, \cdots, K\} \tag{4.13}$$

[3]The threshold is defined as one-fifth of the average number of members of all clusters.

[4]Please note that the variance is employed instead of covariance matrix. This is because the correlation among different motion directions is small for the given data-set employed in this experiment. However, the covariance matrix can be considered in general.

and $n_{k,i}$ is the total number of times activity k occurs at region i while K is the total number of activity patterns. As a result, different local regions will have different weights for one particular activity. For example,

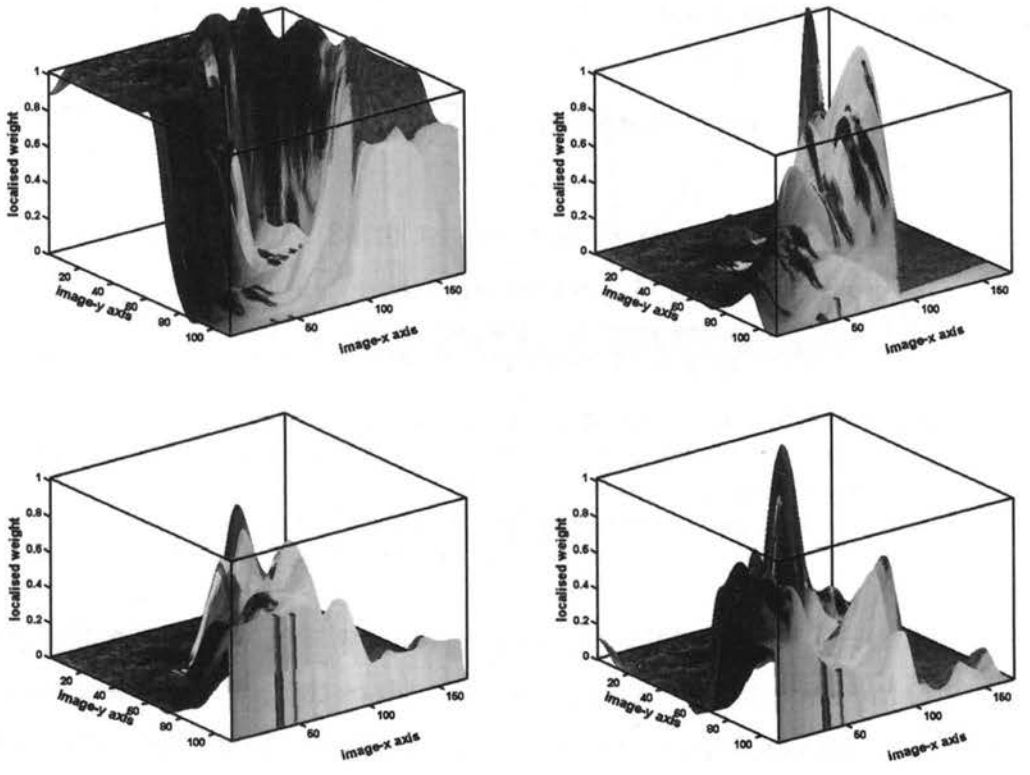


Figure 4.8: Example of localised weights for four different crowd behaviours observed in the monitored scene (UCSD ped1 data-set). The $z-$ axis indicates the localised weights $\omega_{k,i}$ while $x-y$ axis represent the image regions. Each local region i has different weights ω_k (a-d) for four different crowd behaviours.

Figure 4.8(a) shows computed weights for the first crowd behaviour $(k = 1)$ observed in the scene. It can be seen that the weights are locally adapted $\omega_{1,i} \neq \omega_{1,j}$ for $i \neq j$.

4.3.4 Abnormality Detection

Given an unseen sequence, the motion for local regions are first computed as discussed in Section 4.3.1. The normality score for each local region is then computed as:

$$p(\mathrm{x}_i^{new}|\text{normality}) = \sum_{k=1}^{K} \mathrm{w}_{k,i} \frac{1}{\sigma_k \sqrt{2\pi}} \exp(-\frac{(\mathrm{x}_i^{new} - \mu_k)^2}{2\sigma_k^2}) \tag{4.14}$$

where $\mathrm{w}_{k,i}$, μ_k and σ_k are weight (4.13), mean and variance of the k^{th} behaviour pattern (4.12) learned during the training.

The normality score for each video frame, called as the global score, is then computed as the average of all the local scores and it is defined as:

$$g(\mathrm{x}^{new}|\text{normality}) = \frac{1}{N} \sum_{i=1}^{N} p(\mathrm{x}_i^{new}|\text{normality}) \tag{4.15}$$

where N is the total number of regions while $p(\mathrm{x}_i^{new}|\text{normality})$ is computed using equation (4.14). Figure 4.9 shows the normality score computed for one test video sequence from the UCSD ped1 data set. The corresponding ground truth bars (where the indices for abnormal frames are highlighted) shown at the bottom demonstrates that the current method detects the abnormal frames accurately. A video frame is considered as being abnormal if the global normality score for the video frame is smaller than a particular threshold. Different thresholds are applied to generate multiple sets of true positive and false positive rates. The best threshold can be selected where the minimum equal error rate is obtained.

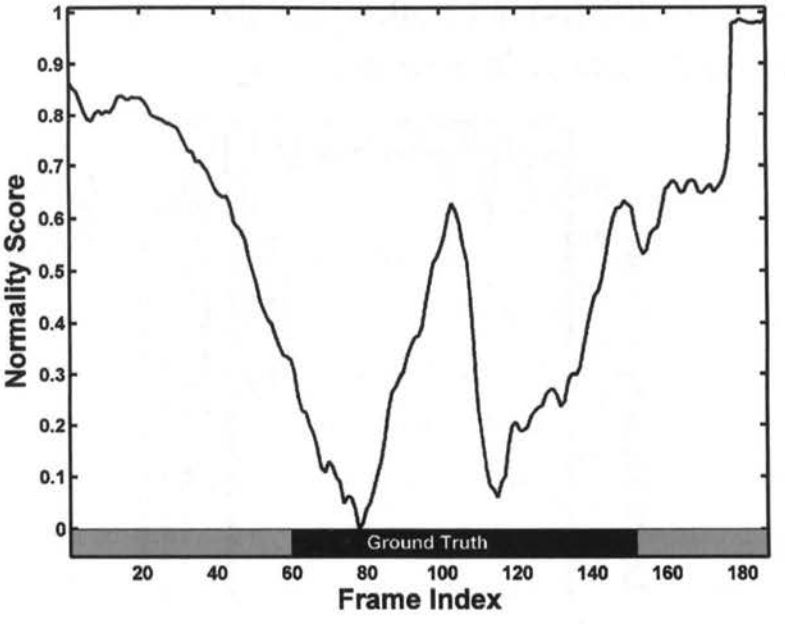


Figure 4.9: Global probability score for one sample video sequence from UCSD ped1 data-set. At the bottom, the corresponding ground truth bars are shown where the highlighted regions (darker colour) denote the indices for abnormal frames.

4.3.5 Abnormality Localisation

Given that a video frame is identified as an abnormal frame by the above process, the anomaly regions can be identified by analysing the normality scores contributed by each local regions of an abnormal scene. Regions with low normality scores (as shown in Figure 4.10) are most likely to be abnormal. Hence, a local region is classified as abnormal if its normality score is smaller than the global score for the whole frame:

$$\mathbf{x}_i^{new} = \begin{cases} \text{abnormal}, & \text{if } p(\mathbf{x}_i^{new}|\text{normality}) < th \\ \text{normal}, & \text{otherwise} \end{cases} \tag{4.16}$$

where th is a threshold and $p(\mathbf{x}_i^{new})$ is computed using equation (4.14). Figure 4.10 shows the local normality score for one sample frame from the test video sequence (Figure 4.9) of UCSD Ped1 data set. At the bottom, the

corresponding ground truth bar is shown where the highlighted regions denote the indices for abnormal regions.

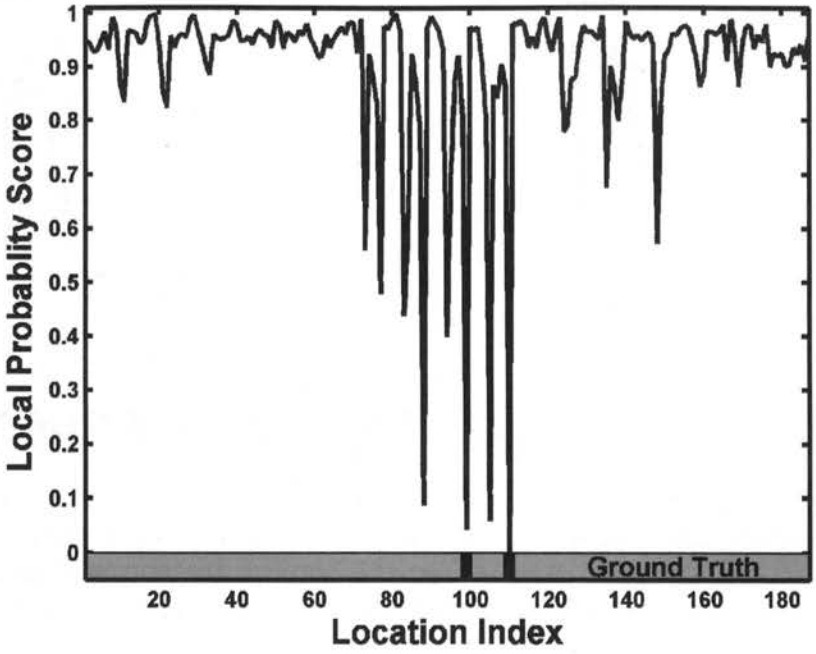


Figure 4.10: Local probability score for one sample frame of a test video sequence from UCSD ped1 data-set. At the bottom, the corresponding ground truth bars are shown where the highlighted regions (darker colour) denote the indices for abnormal regions.

4.3.6 Experimental Results

This experiment evaluates the performance of the proposed method on detecting and localising abnormal regions in the scene using the recently released UCSD data set [5]. This data set contains 98 video sequences from two different scenes (ped1 and ped2): 70 sequences from first scene (ped1) and 28 sequences from the second scene (ped2). Each sequence contains about 200 frames where each frame is a size of 238×158 pixels for ped1 and 360×240 for ped2. The training set (34 sequences from ped1 and 16 from ped2) is provided for learning normal activities of a crowd while the testing

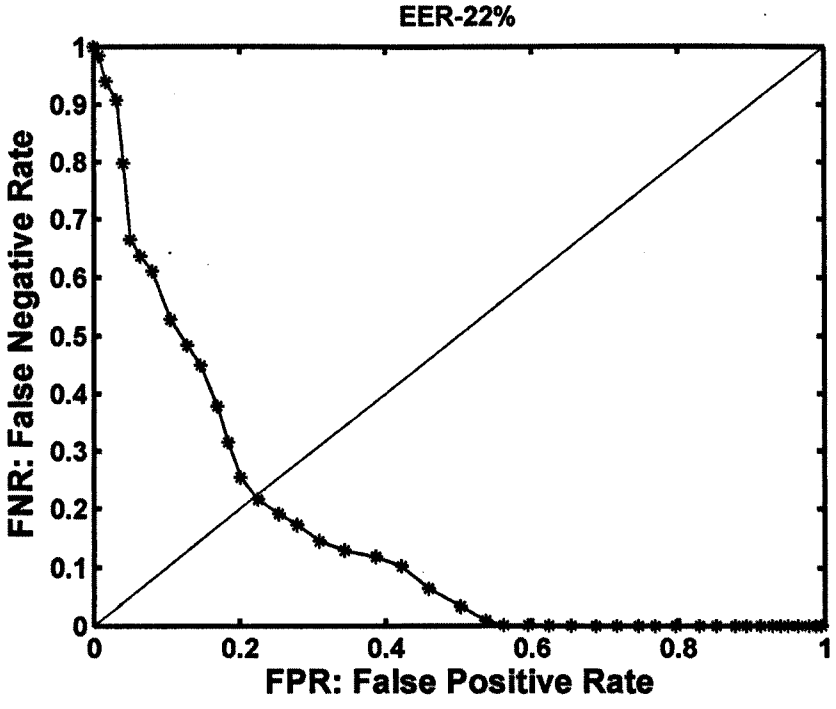
set contains 48 sequences (36 sequences from ped1 and 12 sequences from ped2) in which some of the frames have one or more abnormality activities present.

The prescribed evaluation procedure by [5] is followed to compare the performance of the proposed method with the state-of-the-art methods. This procedure [5] involves two types of evaluations: (i) frame-level abnormality detection, and (ii) within-frame anomaly localisation. For frame-level abnormality detection, all test sequences are associated with ground-truth at frame-level in the form of a binary flag, indicating the presence or absence of an abnormal event in each frame. For within-frame abnormality localisation, some of the frames from a subset of test sequences (10 sequences in Ped1 and 9 sequences in Ped2) have the marked anomalous regions. If at least 40% of detected pixels (belonging to a detected anomaly) match the marked anomalous (ground-truth) pixels, it is considered that the anomaly has been localised correctly, otherwise it is treated as a "miss".

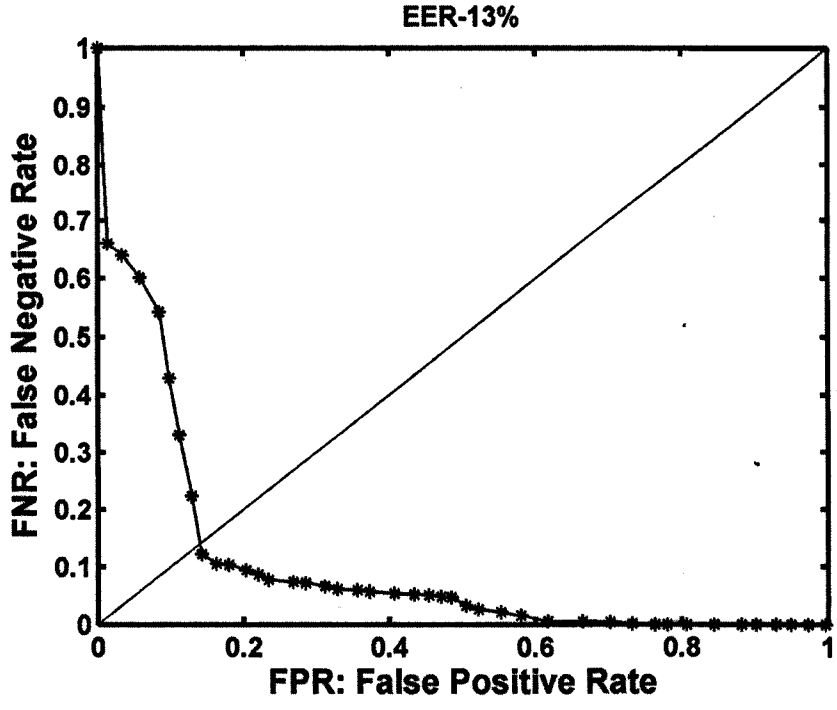
In this experiment, the patch size ($n \times n$) is fixed at 10 for ped1 and ped2 data set where the spatial factor σ_s is set at 0.5 for ped1 data set and 0.4 for ped2 data set. The effect of using different spatial effect σ_s and the patch size ($n \times n$) are studied in Section 4.3.6.4.

4.3.6.1 Abnormality Detection

Table 4.5 shows the comparisons of equal error rates (EER) for frame-level local abnormality detection obtained with the proposed method and the state-of-the-art methods. A smaller EER indicates a better performance of the method. The presented results for other state-of-the-art methods were extracted from the corresponding papers.



(a)



(b)

Figure 4.11: Results of frame level local abnormality detection using UCSD data set. (a) Frame level abnormality detection on ped1 sequence (b) Frame level abnormality detection on ped2 sequence.

	ped1	ped2	avg
SF [104]	31.0%	42.0%	37.0%
MPPCA [81]	40.0%	30.0%	35.0%
SF-MPPCA [102]	32.0%	36.0%	34.0%
Adam *et al.* [10]	38.0%	42.0%	40.0%
MDT [102]	25.0%	25.0%	25.0%
Reddy *et al.* [123]	22.5%	20.0%	21.3%
Ryan *et al.* [127]	23.1%	13.3%	18.2%
Cong *et al.* [45]	19.0%	-	-
proposed method	22.0%	13.5%	17.8%

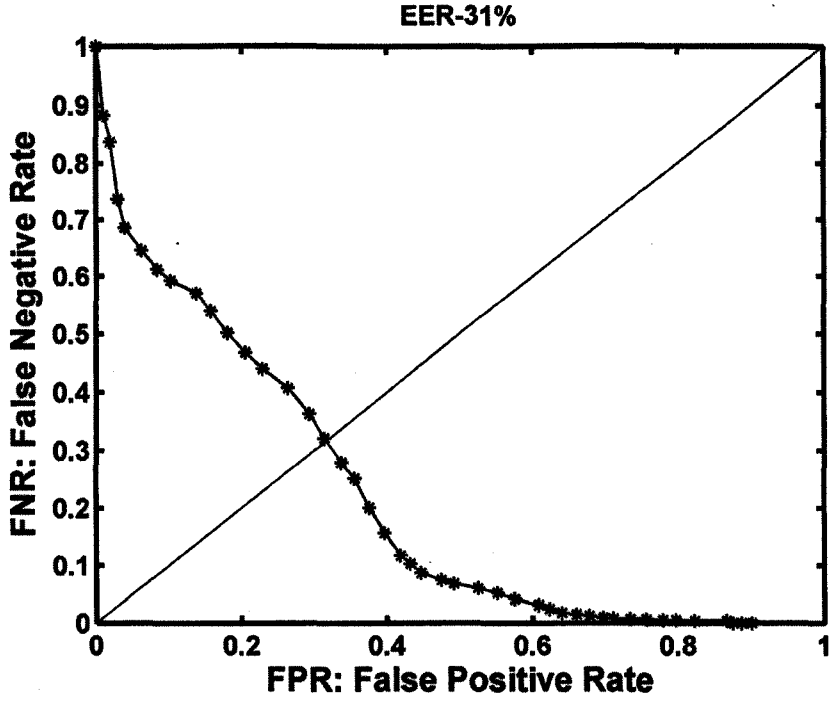
Table 4.5: Equal Error Rate for Local Abnormality Detection on UCSD data set. Please note that [45] tested only on the Ped1 sequence.

It can be observed that this method outperforms or at least is comparable with the state-of-the-art methods by just using the motion feature. Figure 4.11 presents the ROC curves for local abnormality detection with frame-level ground truth. The lowest EER rate is obtained at $th = 0.69$ and $th = 0.82$ for ped1 and ped2 data set respectively. As in [123], the false negative rate is reported instead of true positive rate.

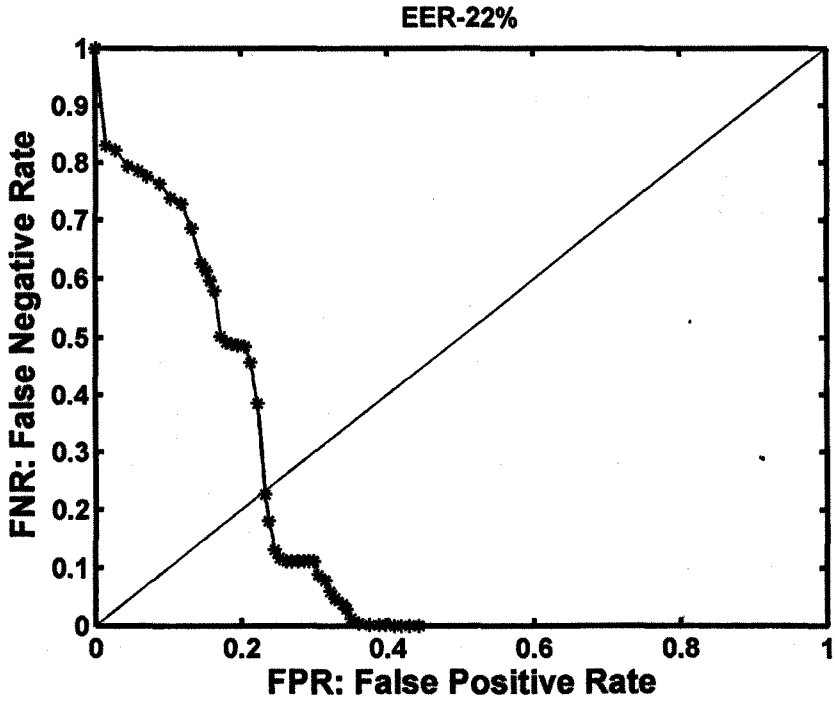
4.3.6.2 Abnormality Localisation

Next, the performance of this method is compared with the state-of-the-art methods for the localisation of abnormal regions. The comparisons of equal error rate for abnormality localisation is given in Table 4.6. The pixel-level ground truth is provided in [102] and the results are compared with the ground truth based on their prescribed procedures. Please note that the other state-of-the-art methods reported their abnormality localising results (equal error rate or detection rate) only for the ped1 sequence. The ROC curves for abnormality localisation are shown in Figure 4.12.

Figure 4.13 shows some example frames for detecting local abnormality. The abnormality of the frame is indicated with a red bar on the left-



(a)



(b)

Figure 4.12: Results of abnormality localisation using UCSD data set. (a) Pixel level abnormality localisation on ped1 (b) Pixel level abnormality localisation on ped2.

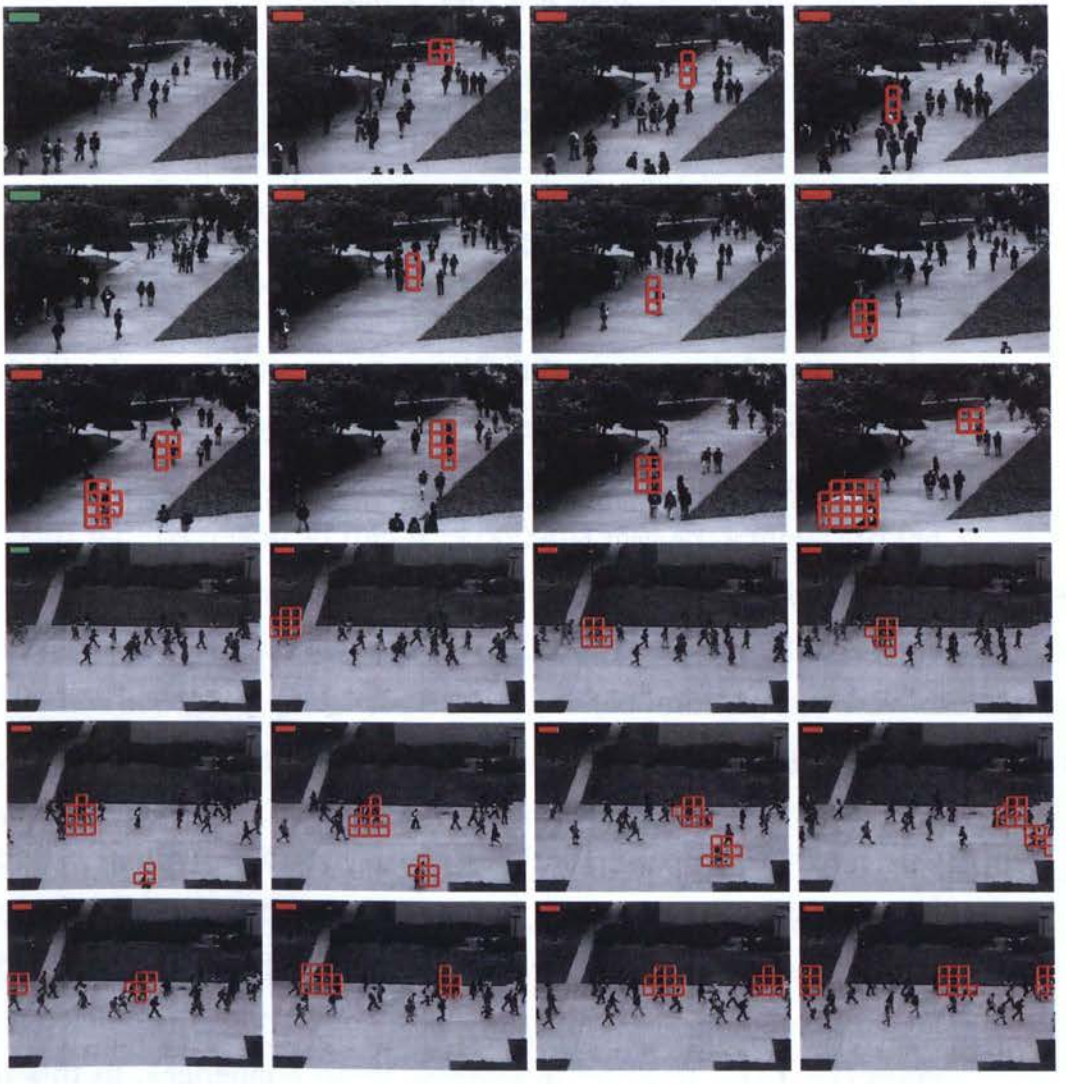


Figure 4.13: Example frames from UCSD data-set for abnormality localisation. The corresponding regions that caused anomalies are highlighted using bounding boxes. More qualitative results are available at `http://youtu.be/b7JYOAH63TQ` and `http://youtu.be/mnt6uhZVFrk`.

	ped1	ped2	avg
SF [104]	79%	-	-
MPPCA [81]	82%	-	-
SF-MPPCA [102]	72%	-	-
Adam *et al.* [10]	76%	-	-
MDT [102]	55%	-	-
Reddy *et al.* [123]	32%	-	-
Cong *et al.* [45]	54%	-	-
proposed method	31%	22%	26.5%

Table 4.6: Equal Error Rate for Abnormality Localisation on UCSD data set. Please note that other methods reported their abnormality localising results only for the ped1 sequence.

top corner of the image while the regions, which are the likely source of the abnormal dynamics, are highlighted using bounding rectangles. More qualitative results are available at `http://youtu.be/b7JYOAH63TQ` and `http://youtu.be/mnt6uhZVFrk`. It can be observed that this method accurately localises abnormal regions without incurring high computational cost.

4.3.6.3 Computation Time

One major concern of the Laplacian Eigenmap method is the computational cost of the pair-wise distance for each pair of local regions where the cost is commemorate with the number of data-point. In addition, the large size of pair-wise matrix is also a concern for the efficiency of the program [72, 111]. In order to address these challenges, in this method, an incremental approach is employed for learning crowd activities. The model for regular crowd activities is first learnt using a small training data-set (section 4.3.2 and 4.3.3).

Given the rest of the training data, the probability score of new region descriptors belonging to the current model is computed as discussed in

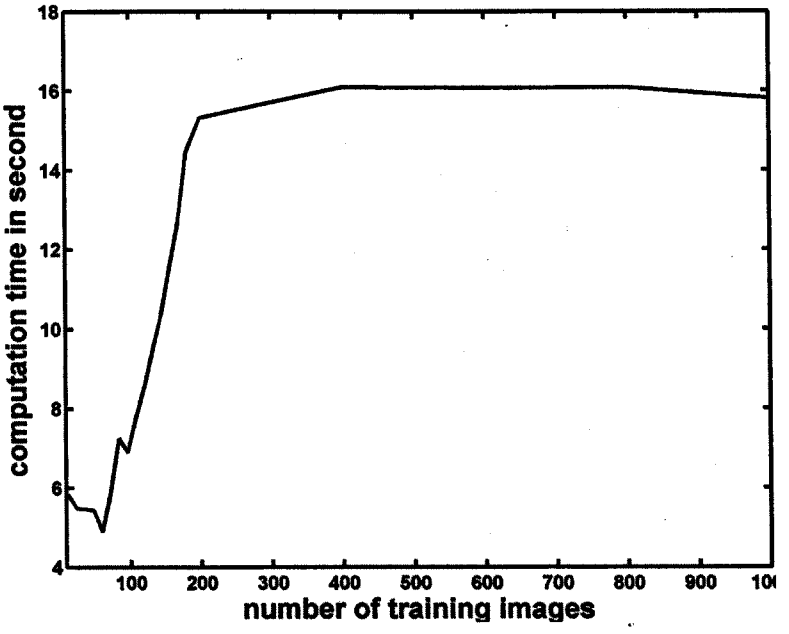


Figure 4.14: Computation time for training images for UCSD ped1 data set

section 4.3.4. Then, the new region descriptors, which have a significant discrepancy from the learned model, are accumulated over time and new activities are learnt from these accumulated data. In this way, the regular behaviour of a given crowded scene can be trained efficiently. The training time for UCSD data-set (ped1) is given in Fig 4.14. As expected, the computational cost increases as the number of training images increases. However, the cost reaches a stable point after some time as re-training is required only for a sub-set of images.

Figure 4.15 shows the average testing time on the UCSD data set (ped1) with an image size of 238×158. It can be observed that this method takes less than 0.006 seconds (with the standard deviation of 0.2 millisecond) to test a new video frame while the dynamic texture model [102] took about 25 seconds, the combined feature model [123] took about 0.08 seconds and Cong [45] took about 3.8 seconds respectively. The platform employed in this experiment has a dual-Core $3GHz$ processor with $4GB$ RAM.

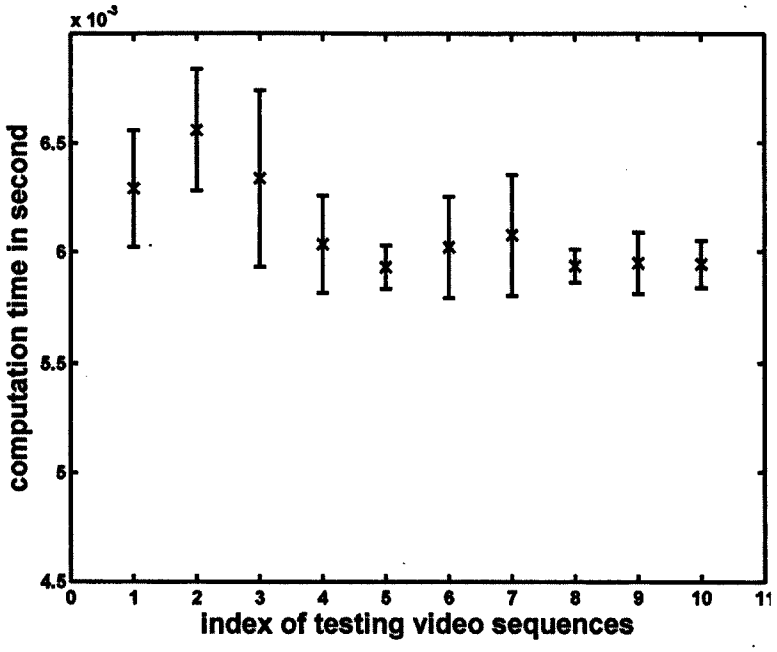


Figure 4.15: Average testing time and standard deviation for video sequences in UCSD ped1 testing data set.

4.3.6.4 Performance Evaluation

This section describes additional experimental results conducted to further analyse the performance of the proposed method. First, the performance is evaluated with respect to spatial information σ_s (equation 4.8) and the patch size n.

patch size($n \times n$)	$\sigma_s = 0.3$	$\sigma_s = 0.4$	$\sigma_s = 0.5$	$\sigma_s = 0.6$	$\sigma_s = 0.7$	$\sigma_s = 0.8$	$\sigma_s = 0.9$	$\sigma_s = 1.0$
8×8	20.81%	20.67%	20.92%	20.50%	20.75%	20.84%	21.11%	20.59%
10×10	20.03%	**13.52%**	**13.81%**	**14.13%**	19.25%	19.69%	19.87%	19.82%
13×13	15.77%	19.14%	18.37%	14.61%	15.03%	19.51%	19.35%	19.32%
16×16	19.91%	19.04%	19.33%	19.22%	19.41%	19.83%	19.49%	19.29%

Table 4.7: Equal error rates for abnormal detection on UCSD ped2 data set using different spatial effect σ_s and different patch size n.

Table 4.7 lists equal error rates for abnormality detection on the UCSD ped2 data set using different σ_s vales and patch size n. Please note that the value of σ_s is from 0 to 1 as the Euclidean distances between data-points

have already been normalised. It is observed that the equal error rates vary only slightly based on different patch sizes and σ_s values. The best results are achieved at patch size (10×10) and the values of σ_s between 0.4 and 0.6.

EER	feature+space+time	feature+space	feature+time
Abnormality Detection on UCSD Ped1	22.0%	22.0%	24.0%
Abnormality Detection on UCSD Ped2	13.5%	22.9%	19.0%
Localization on UCSD Ped1	31.0%	33.0 %	35.0%
Localization on UCSD Ped2	22.0%	34.8 %	33.0%

Table 4.8: Equal error rates for abnormal detection and localisation on UCSD data-set using different combination of three components in equation 4.8.

Next, the effect of spatial and temporal information on the detection performance is studied. Evaluations are made using different combinations of three components in equation (4.8): (i) motion, space and time(default configuration), (ii) motion and space (iii) motion and time. EER obtained by different configurations are shown in Table 4.8. It can be observed that the incorporation of spatial and temporal information improves the localisation of the actual regions that cause an anomaly.

In addition, the impact of localised weights on the abnormality detection accuracy is further evaluated. As mentioned earlier, in this method, weights for different crowd behaviours are computed using co-occurrence of activities at a given location (equation 4.13). It is observed that the current approach using localised weights provides a high accuracy (low equal error rates) for both abnormality detection and localisation of abnormal regions in the scene. To further demonstrate the strength of localised weights, this experiment computes the detection accuracy using the weights given by the EM algorithm where the weight ω_k represents the percentage of data points in the k^{th} cluster. In contrast to the proposed configuration, the weight ω_k is the same for all local regions, i.e.

$\omega_{k,i} = \omega_{k,j}$ for $i \neq j$. Figure 4.16 shows ROC curves for local abnormality detection and localisation using region-adapted weights (proposed configuration) and the non-adapted weights given by the EM algorithm. It is observed that the proposed adaptation scheme by equation (4.13) provides a better detection performance.

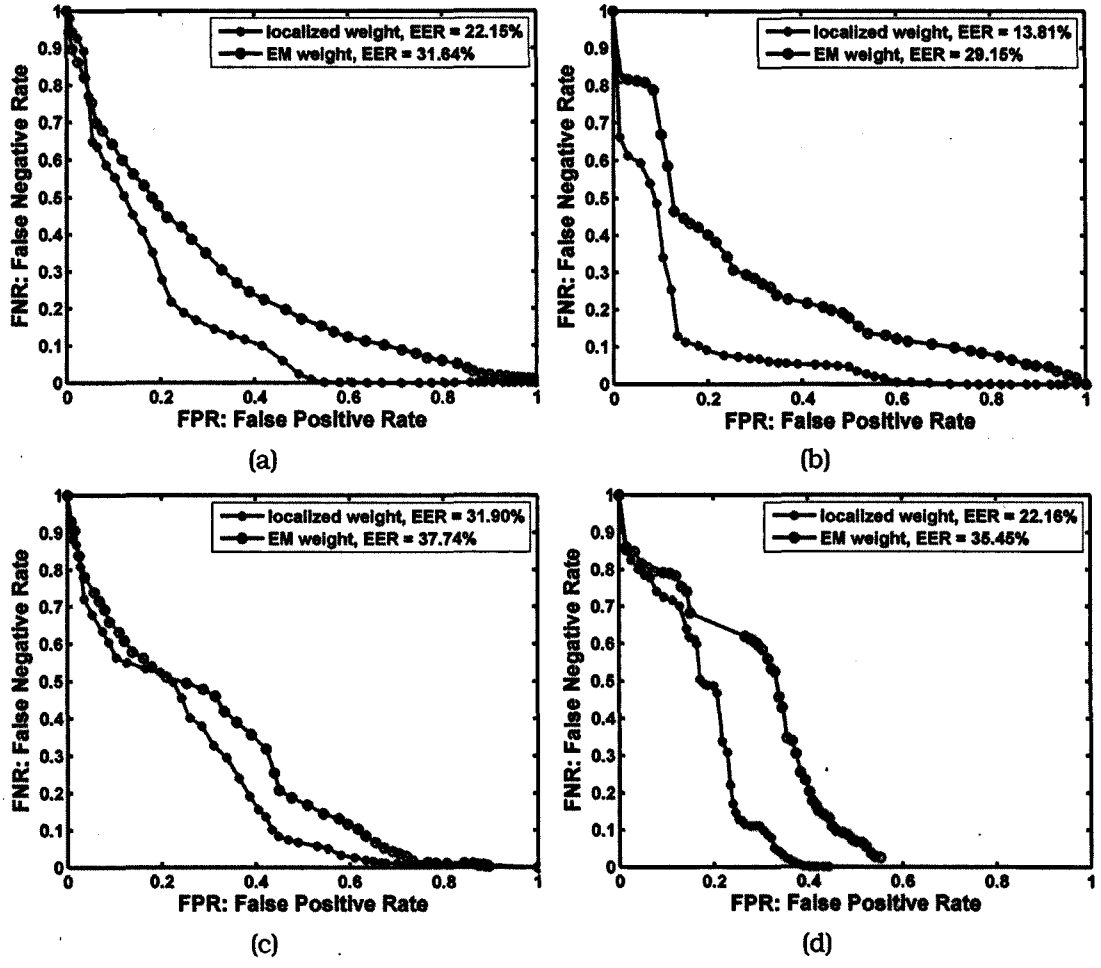


Figure 4.16: Comparisons of the proposed method with the results using weights obtained by the EM algorithm. (a) ROC curves for frame-level local abnormality detection on UCSD ped1 data-set (b) ROC curves for frame-level local abnormality detection on UCSD ped2 data-set (c) ROC curves for pixel-level abnormality localisation on UCSD ped1 data-set (d) ROC curves for pixel-level abnormality localisation on UCSD ped2 data-set.

4.4 Summary

In this chapter, two manifold learning-based methods have been presented for the detection of anomalies in a crowded scene. While the first approach focusses on detecting global abnormality, visualising and segmenting crowd events, the second one emphasises on detecting and localising abnormal regions in a crowded scene.

The first approach introduces a manifold learning method to achieve visualisation and modelling of video events in a low dimensional space. The proposed manifold learning method preserves the spatial temporal property as well as the characteristic of the video. It has been demonstrated that the low-dimensional representation of a video serves as a compact, yet informative, representation for analysing the content of a crowded video. Different tasks of video content analysis such as visualisation, video event segmentation and abnormality detection are achieved by analysing these video trajectories in the embedded space. Qualitative and quantitative results show the promising performance of the proposed method.

The second approach models the local motion of a crowded scene by employing temporal constrained *Laplacian Eigenmaps*. The pair-wise graph is constructed by considering the visual context of multiple local patches in both spatial and temporal domain. A local probabilistic model is proposed to represent the regular behaviour of a crowd. This local probabilistic model allows for the detection of abnormalities in both local and global context but also for an accurate localisation of abnormal regions. Experimental results have shown that this approach successfully detects and localises abnormal regions in a crowded scene.

"There is always a choice in the way we do our work, even if there is no choice in the work itself. Enthusiasm Makes A Difference!"

Stephen Lundin

5

Conclusion

The work carried out in this thesis aimed to address three significant problems encountered in numerous computer vision applications. The problems are: (i) tracking multiple targets in a complex scene, (ii) detection and localisation of abnormal regions in crowded scenes.

Chapter 2 provided a review and systematic comparison of the state of the art on crowd video analysis. The emphasise is on existing literature for tracking individuals in a crowd and understanding crowd behaviour. This review provides a reference point to computer vision researchers currently working on crowd analysis. The merits and weaknesses of various approaches for each topic are discussed and some possible future directions are recommended to improve the existing methods.

Chapter 3 described a novel multi-target tracking method that employed a particle swarm optimisation algorithm. This method contributes to the state of the arts by: (1) introducing an idea of multiple interactive swarms to the standard PSO to track multiple pedestrians in a crowd, (2) incorporating higher level information such as social behaviour (motion information among pedestrians) in the process of finding optima in a high dimensional space, (3) integrating constraints provided by temporal continuity of target tracks and the strength of person detection, and (4) initialis-

ing a separate swarm for each new person entering the scene. Experiments on videos from CAVIAR, PETS, OXFORD data sets, have indicated that the proposed method obtains considerably better performance (both qualitatively and quantitatively) than the state-of-the-art methods. Specifically, on the PETS data set, this method achieves 86% of tracking precision which is nearly 10% higher than the best reported results. It is also shown that the proposed method is able to track targets with illumination changes and heavy occlusions in both indoor and outdoor environment.

Chapter 4 described techniques for detecting and localising abnormal regions in a crowded scene. In particular, two new approaches have been developed for the detection of anomalies in a crowded scene. The first approach proposed a novel manifold learning method that preserves the spatial temporal property and the characteristic of the video. It has been demonstrated that the low-dimensional representation of a video serves as a compact, yet informative, representation for analysing the content of a crowded video. Different tasks of video content analysis such as visualisation, video event segmentation and abnormality detection were performed by analysing these video trajectories in the embedded space. Next, this approach was further extended to accurately localise abnormal regions having unknown behaviour. Experiments with the recently published UCSD data-sets have shown that the proposed method achieves comparable results with the state-of-the-art methods without sacrificing computational simplicity.

5.1 Future Directions

The methods developed in this work can be improved and extended along a number of directions. Some of these ideas are discussed in this section.

The algorithm presented in Chapter 3 has demonstrated that incorporating social interactions among targets improves the tracking accuracy, especially in heavy occlusion. The performance of this algorithm can be further improved by incorporating more sophisticated behaviour models, by analysing the flow dynamics and the structure of the scene. In addition, future work is required to dynamically evaluate the performance of the tracking algorithm and to update the optimisation process based on the reliable scores provided by the evaluation process.

The local abnormality detection techniques in Chapter 4 open different research directions to explore. One direction is to divide the image space into cells of different sizes using a spatial pyramid approach. This should improve the accuracy for localising regions with different abnormal behaviours. Another interesting direction is to extend the proposed techniques to the area of multi-camera visual context analysis. The local probability model of the monitored scene must be improved to combine the information captured by multiple cameras.

A

Manifold Learning Algorithms

A.1 Gaussian Process Latent Variable Models

Gaussian process latent variable models (GPLVM) [87] is a probabilistic nonlinear extension of principal component analysis (PCA) [143] which is a well established method for linear dimensional reduction. Given a high dimensional data $\mathbf{X} \in \Re^{N \times D}$,the linear relationship between the latent (embedded) variables and the data points is:

$$\mathbf{x}_n = \mathbf{W}\mathbf{y}_n + \eta_n \tag{A.1}$$

where $\mathbf{y}_n$ is a $q-$ dimensional latent variable associated with the data point $\mathbf{x}_n$, $\mathbf{W}$ is a $D \times q$ matrix and the noise values $\eta_n \in \Re^{D \times 1}$ is an independent sample from a spherical Gaussian distribution with zero mean and covariance $\beta^{-1}\mathbf{I}$.

In GPLVM, a spherical Gaussian distribution (with zero mean and covariance I) is selected as a prior distribution for W:

$$p(\mathbf{W}) = \prod_{i=1}^{D} \aleph(\mathbf{w}_i|\mathbf{0}, \mathbf{I}) \tag{A.2}$$

and then W is marginalised given a likelihood:

$$p(\mathbf{X}|\mathbf{Y},\beta) = \prod_{d=1}^{D} p(\mathbf{x}_{:,d}|\mathbf{Y},\beta) \quad \text{(A.3)}$$

$$= \prod_{d=1}^{D} \aleph(\mathbf{x}_{:,d}|\mathbf{0},\mathbf{Y}\mathbf{Y}^T + \beta^{-1}\mathbf{I}) \quad \text{(A.4)}$$

$$= \frac{1}{(2\pi)^{\frac{DN}{2}}|\mathbf{K}|^{\frac{D}{2}}} \exp\Big[-\frac{1}{2}tr(\mathbf{K}^{-1}\mathbf{X}\mathbf{X}^T)\Big] \quad \text{(A.5)}$$

where $\mathbf{x}_{:,d}$ represents the d^{th} column of $\mathbf{X}$ and $\mathbf{K} = \mathbf{Y}\mathbf{Y}^T + \beta^{-1}\mathbf{I}$. Then, the objective function is the log likelihood given by the log of (A.5):

$$L = -\frac{DN}{2}\ln(2\pi) - \frac{D}{2}\ln|\mathbf{K}| - \frac{1}{2}tr(\mathbf{K}^{-1}\mathbf{X}\mathbf{X}^T) \quad \text{(A.6)}$$

By optimising the log likelihood with respect to the latent variables Y is given by [101] as

$$\frac{\partial L}{\partial \mathbf{Y}} = \mathbf{K}^{-1}\mathbf{X}\mathbf{X}^T\mathbf{K}^{-1}\mathbf{Y} - D\mathbf{K}^{-1}\mathbf{Y} \quad \text{(A.7)}$$

and then the latent variables Y where the gradient is zero is given by:

$$D\mathbf{K}^{-1}\mathbf{Y} = \mathbf{K}^{-1}\mathbf{X}\mathbf{X}^T\mathbf{K}^{-1}\mathbf{Y} \quad \text{(A.8)}$$

$$\mathbf{K}^{-1}\mathbf{Y} = \frac{1}{D}\mathbf{K}^{-1}\mathbf{X}\mathbf{X}^T\mathbf{K}^{-1}\mathbf{Y} \quad \text{(A.9)}$$

$$\mathbf{Y} = \frac{1}{D}\mathbf{X}\mathbf{X}^T\mathbf{K}^{-1}\mathbf{Y} \quad \text{(A.10)}$$

Computing the eigen-decomposition results:

$$\mathbf{Y} = \mathbf{U}_q\mathbf{L}\mathbf{R}^T \quad \text{(A.11)}$$

where $\mathbf{U}_q \in \Re^{N\times q}$ is a matrix whose columns are the first q eigenvectors of $\mathbf{X}\mathbf{X}^T$. $\mathbf{L}$ is a $q \times q$ diagonal matrix whose j^{th} element is given by $l_j = (\lambda_j - \beta^{-1})^{-1/2}$ where λ_j is the eigenvalue associated with the j^{th} element of $D^{-1}\mathbf{X}\mathbf{X}^T$ and R is a $q \times q$ rotation matrix.

A.2 Isometric Feature Mapping

Isometric feature mapping (ISOMAP) [146] finds a low dimensional representation that preserves geometric features of high-dimensional observations. This method can be seen

as an extended version of multi-dimensional scaling (MDS) [46] which is a classical technique for embedding dis-similarity information into Euclidean space. This method can be generalised in two steps: 1) estimating the geodesic distances and 2) finding a low dimensional space where the Euclidean distances between data points in the space match the geodesic distances found in step 1.

The first step constructs a neighbourhood graph G in which each data point $\mathbf{x}_i$ is connected with its k nearest neighbours in the given data set $\mathbf{X} \in \Re^{N \times D}$. The shortest path between each pair of data points in the graph is then computed using Dijkstra's [49] or Floyd's [53] algorithm. Here, the shortest path can estimate the geodesic distance (distance on manifold) between these data points. Given the pairwise geodesic distances between all data points in the given data set $\mathbf{X}$, the second step finds a low dimensional representation of $\mathbf{y}_i$ of the data point $\mathbf{x}_i$ by minimising the following cost function:

$$\phi(\mathbf{Y}) = \sum_{i=1}^{N}\sum_{j=1}^{N}(d(\mathbf{x}_i, \mathbf{x}_j) - \|\mathbf{y}_i - \mathbf{y}_j\|^2) \quad . \tag{A.12}$$

where N is the total number of data points in the data set, $d(\mathbf{x}_i, \mathbf{x}_j)$ is the geodesic distance between high-dimensional data points $\mathbf{x}_i$ and $\mathbf{x}_j$ and $\|\mathbf{y}_i - \mathbf{y}_j\|^2$ is the squared Euclidean distance between low dimensional data points $\mathbf{y}_i$ and $\mathbf{y}_j$. The minimum of this cost function (A.12) is given by the eigen-decomposition of the product matrix $\mathbf{X}\mathbf{X}^T$ of the high dimensional data [54].

A.3 Local Linear Embedding

Local linear embedding (LLE) [132] is a technique that finds a low dimensional space of high dimensional data points by preserving local properties of the data. In contrast to Isomap, the preservation of local properties allows LLE to successfully embed non-convex manifolds. In LLE, each data point is constructed as a linear combination of its neighbours and the reconstruction error is computed by the following cost function:

$$\varepsilon = \|\mathbf{x}_i - \sum_{j=1}^{N_k} \omega_{ij}\mathbf{x}_j\|^2 \tag{A.13}$$

where $j \in N_k(\mathbf{x}_i)$ is a neighbour point of $\mathbf{x}_i$ and N_k is total number of neighbour points. In general, the nearest neighbours N_k are defined either by 1) finding the k closest points or 2) selecting all points $\mathbf{x}_j \in \mathbf{X}$ such that $d(\mathbf{x}_j, \mathbf{x}_i) < \epsilon$. The local linearity assumption

implies that the reconstruction weights are invariant to translation, rotation and scaling. Hence, the characterisation of local geometry in the original data space can be expected to be equally valid in a low dimensional space. Thus, finding the low dimensional data $\mathbf{y}_i$ amounts to minimising the cost function:

$$\phi(\mathbf{Y}) = \sum_{i=1}^{N} \|\mathbf{y}_i - \sum_{j=1}^{N_k} \omega_{ij}\mathbf{y}_j\|^2 \tag{A.14}$$

It is observed that the low dimensional data point $\mathbf{y}_i$ can be translated without effecting the cost $\phi(\mathbf{Y})$. This degree of freedom is removed by forcing the co-ordinates to be centred on the origin $\sum_i^N \mathbf{y}_i = 0$. By constraining the embedding vector to have a unit covariance($\frac{1}{N}\sum_i^N \mathbf{y}_i\mathbf{y}_i^T = \mathbf{I}$) and imposing the sum-to-one constraint $\sum_j \omega_{ij} = 1$, the equation A.14 can be reformulated as:

$$\begin{aligned}
\phi(\mathbf{Y}) &= \sum_i \mathbf{y}_i\mathbf{y}_i^T - \sum_i \mathbf{y}_i(\sum_j \omega_{ij}\mathbf{y}_j) - (\sum_j \omega_{ij}\mathbf{y}_j)\sum_i \mathbf{y}_i + \sum_i (\sum_j \omega_{ij}\mathbf{y}_j)^2 \\
&= \mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T(\mathbf{W}\mathbf{Y}) - (\mathbf{W}\mathbf{Y})^T\mathbf{Y} + (\mathbf{W}\mathbf{Y})^T(\mathbf{W}\mathbf{Y}) \\
&= ((\mathbf{I}-\mathbf{W})\mathbf{Y})^T((\mathbf{I}-\mathbf{W})\mathbf{Y}) \\
&= \mathbf{Y}^T(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})\mathbf{Y} \\
&= \mathbf{Y}^T\mathbf{M}\mathbf{Y} \qquad \text{where} \quad \mathbf{M} = (\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})
\end{aligned} \tag{A.15}$$

where $\mathbf{I}$ is an identity matrix and $\mathbf{W}$ is a reconstruction weight matrix where each entry ω_{ij} gives the contribution of the j^{th} data point to the reconstruction of i^{th} data point. Then, the low dimensional representation $\mathbf{y}_i$ that minimises the above cost function $\phi(\mathbf{Y})$ (A.15) are found by computing the eigenvectors corresponding to the smallest d_s nonzero eigenvalues of the product $(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$ [126].

A.4 Laplacian Eigenmaps

Laplacian Eigenmaps (LE) [28] is another manifold learning algorithm which computes a low dimensional, neighbourhood preserving embedding of high dimensional data. Similar to LLE, Laplacian Eigenmaps computes a low-dimensional data representation by minimising the cost (distances) between a data point and its k nearest neighbours. The cost function is defined in a weighted manner, i.e., the distance between a data-point and its first nearest neighbour contributes more to the cost function than the distance between the data-point and its second nearest neighbour. LE algorithm can be generalised into

three steps: constructing an adjacency graph, computing weights and mapping to a low dimensional space. In the first step, a neighbourhood graph G in which each point $\mathbf{x}_i$ is connected to its k nearest neighbours. As in LLE, a neighbour point can be defined either by $k-$nearest rule where $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected if i is among k nearest neighbours of point j or ϵ rule where $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected if their distance is less than ϵ.

In the second step, each connected edge between i and j is weighted by ω_{ij}:

$$\omega_{ij} = \begin{cases} \exp(\frac{-\|d(\mathbf{x}_i,\mathbf{x}_j)\|^2}{\sigma^2}, & \text{if } i \text{ and } j \text{ are connected;} \\ 0, & \text{otherwise.} \end{cases} \tag{A.16}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is a distance between i and j and σ is a parameter for Gaussian function and defined manually. On the other hand, the weight can be defined simply such that $\omega_{ij} = 1$ if i and j are connected and equals to zero otherwise. Next, the low dimensional representation $\mathbf{y}_i$ is computed by minimising the following cost function.

$$\phi(\mathbf{Y}) = \sum_{i=1}^{N}\sum_{j=1}^{N}\omega_{ij}\|\mathbf{y}_i - \mathbf{y}_j\|^2, \tag{A.17}$$

where ω_{ij} is computed with above equation (A.16) and $\mathbf{y}_i$ is the low-dimensional representation of each data point $\mathbf{x}_i$. The above equation can be expanded as:

$$\begin{aligned} \phi(\mathbf{Y}) &= \sum_{ij}\omega_{ij}[\|\mathbf{y}_i\|^2 + \|\mathbf{y}_j\|^2 - 2\mathbf{y}_i\mathbf{y}_j], && \text{(A.18)} \\ &= \sum_{ij}\omega_{ij}\|\mathbf{y}_i\|^2 + \sum_{ij}\omega_{ij}\|\mathbf{y}_j\|^2 - 2\sum_{ij}\omega_{ij}\mathbf{y}_i\mathbf{y}_j, && \text{(A.19)} \\ &= \sum_{i}m_{ii}\|\mathbf{y}_i\|^2 + \sum_{j}m_{jj}\|\mathbf{y}_j\|^2 - 2\sum_{ij}\omega_{ij}\mathbf{y}_i\mathbf{y}_j, && \text{(A.20)} \\ &= 2\mathbf{Y^T}\mathbf{M}\mathbf{Y} - 2\mathbf{Y^T}\mathbf{W}\mathbf{Y}, && \text{(A.21)} \\ &= 2\mathbf{Y^T}\mathbf{L}\mathbf{Y}, && \text{(A.22)} \end{aligned}$$

where $\mathbf{W}$ is the weighted neighbourhood matrix and $\mathbf{M}$ is the diagonal weight matrix in which each entry is a total sum of each row of weight matrix, $\mathbf{W}$, and computed as $m_{ii} = \sum_j \omega_{ij}$. The Laplacian graph $\mathbf{L}$ of the weighted neighbour graph $\mathbf{W}$ is computed by $\mathbf{L} = \mathbf{M} - \mathbf{W}$. The objective function for both LE and LLE are similar and differ only in how the matrix L is computed. Hence, the low dimensional space can be searched by finding the optimum $\mathbf{Y}$:

$$\mathbf{Y}_{\text{opt}} = \arg\min_{\mathbf{Y}}(\mathbf{Y}^T L\mathbf{Y}) \quad \text{subject to } \mathbf{Y}^T M\mathbf{Y} = 1, \tag{A.23}$$

and this can be reduced to solving the generalised eigenvalue problem,

$$\mathbf{L}\mathbf{y} = \lambda \mathbf{M}\mathbf{y}. \tag{A.24}$$

for the smallest k_s non-zero eigenvalue problems.

B

Group Motion Analysis for Video Content Clustering and Abnormality Detection

Chapter 4 presented how to address the problem of abnormality detection and localisation in a crowded scene. Crowd dynamics are employed to model the regular motion patterns of a crowd in the monitored scene and abnormal events are detected by finding deviations from the learnt motion patterns. These methods can be extended for understanding group behaviours in an enclosed scene.

B.1 Introduction

The emphasis of this work is on group behaviour which lies at an intermediate level, between individual and crowd, with a countable number of targets. Compared to behaviour analysis of a single target [58, 69] or analysis of crowd dynamics [83, 85], the problem of group behaviour analysis is less studied. In [130, 174], group activities are recognised by modelling the interactions among targets. However, most of these methods are limited to group activities with a fixed number of group members. Recently, more

researchers [41,47,94,109] have studied on modelling different group activities with varying number of targets. However, explicit modelling of group activities is hard to achieve when complex interactions occur.

Methods that provide group activity analysis are useful for applications where many interacting targets need to be monitored over time. In particular, they have important implications for vision-based analysis of living organisms, which has countless applications in biology and medicine. While more generally applicable, this work specifically focusses on analysing a group of living organisms such as fish in a confined area. Work presented here is part of a larger research project to monitor water security in Singapore [52]. Compared to the sensor-based approaches [1,4], key advantages of monitoring water toxicity using living organisms are its rapid response and ability to perform continuous monitoring at critical locations. Hence, over the last decade, computer vision approaches [20,106,151] that provide understanding behaviour of living organisms have gained prominence in the water security and environment domain. The objective is to detect abnormal behaviour of fish and alert officials to take necessary actions in time. In addition, studying recorded video sequences can provide information to the public safety team to predict the trend of fish behaviour and to prepare actions for unusual events.

However, this domain introduces many challenges that are quite different from the domains in which most multi-target behaviour analysis algorithms are evaluated. One important challenge is the unpredictable nature of targets in a confined area. As there is no scene layout, movements of the targets are random and confined in a small area. Therefore, optical flow-based methods presented in the previous chapter are not directly applicable for the given scenario. Frequent occurrence of occlusions among targets and their similar appearance also make tracking of individuals and point of interest challenging tasks.

This chapter presents a new technique to detect abnormality behaviour of a group of interacting targets. A macroscopic representation is employed where the whole group is considered as a single entity. The representation of group patterns is achieved by using a signed distance map, which combines both shape and motion information of multiple targets. By studying the variations of signed-distance-maps over time, a long video sequence is divided into short video clips. Then each video clip is represented by a set of representative key frames, extracted by a spectral clustering method. Given a library of video segments represented by a set of key frames, the next step projects video segments into a low dimensional space. This provides a compact and effective representation of a long video sequence and a way to automatically group video segments

into different video content and identify the video segments contained abnormal events.

The remainder of this chapter is organised as follows. Section B.2 describes the steps involved in the activity representation of targets, followed by an explanation of the process used to extract key video frames in Section B.3.1. The natural grouping of video segments is presented in Section B.3, which also explains how to detect video segments with abnormal group events. The effectiveness and robustness of the proposed method is demonstrated through various experiments in Section B.4. Specifically, the proposed method is evaluated to detect abnormal group events of multiple interacting targets in a confined area for the water security domain. Finally, a summary of this chapter is provided in Section B.5.

B.2 Activity Representation

This section presents the proposed activity representation of targets where shape and motion information of the whole group are combined to represent the whole group as a single entity. Given a video frame, the contours of foreground objects are first extracted by a background subtraction algorithm and represented using a signed distance map.

B.2.1 Motion Segmentation

Background subtraction has been widely used in motion segmentation where a fixed camera is used to observe dynamic scenes. In this method, the background modelling and foreground subtraction are performed using the approach presented in [97]. In the learning phase, a sequence of video frames $\mathbf{I} = \{\mathbf{I}_t(i,j) \| t = 1, 2, \cdots, T\}$ is first collected in a scene with both moving and stationary targets. Then, a background scene B is generated by removing foreground targets using a median filter as follows:

$$\mathbf{B}(i,j) = \arg \min_{\mathbf{I}_p(i,j)} \sum_{t=1}^{T} \|\mathbf{I}_t(i,j) - \mathbf{I}_p(i,j)\| \tag{B.1}$$

where $p = \{1, 2, \cdots, T\}$ and $I_t(i,j)$ is the colour vector[1] of t^{th} image at position (i,j). This generates a clean background scene where the foreground pixels corresponding to targets are removed. Then, the background scene B is further decomposed into C number of ho-

[1]It is observed that red-green-blue (RGB) colour space is good enough for the scenarios given in this work.

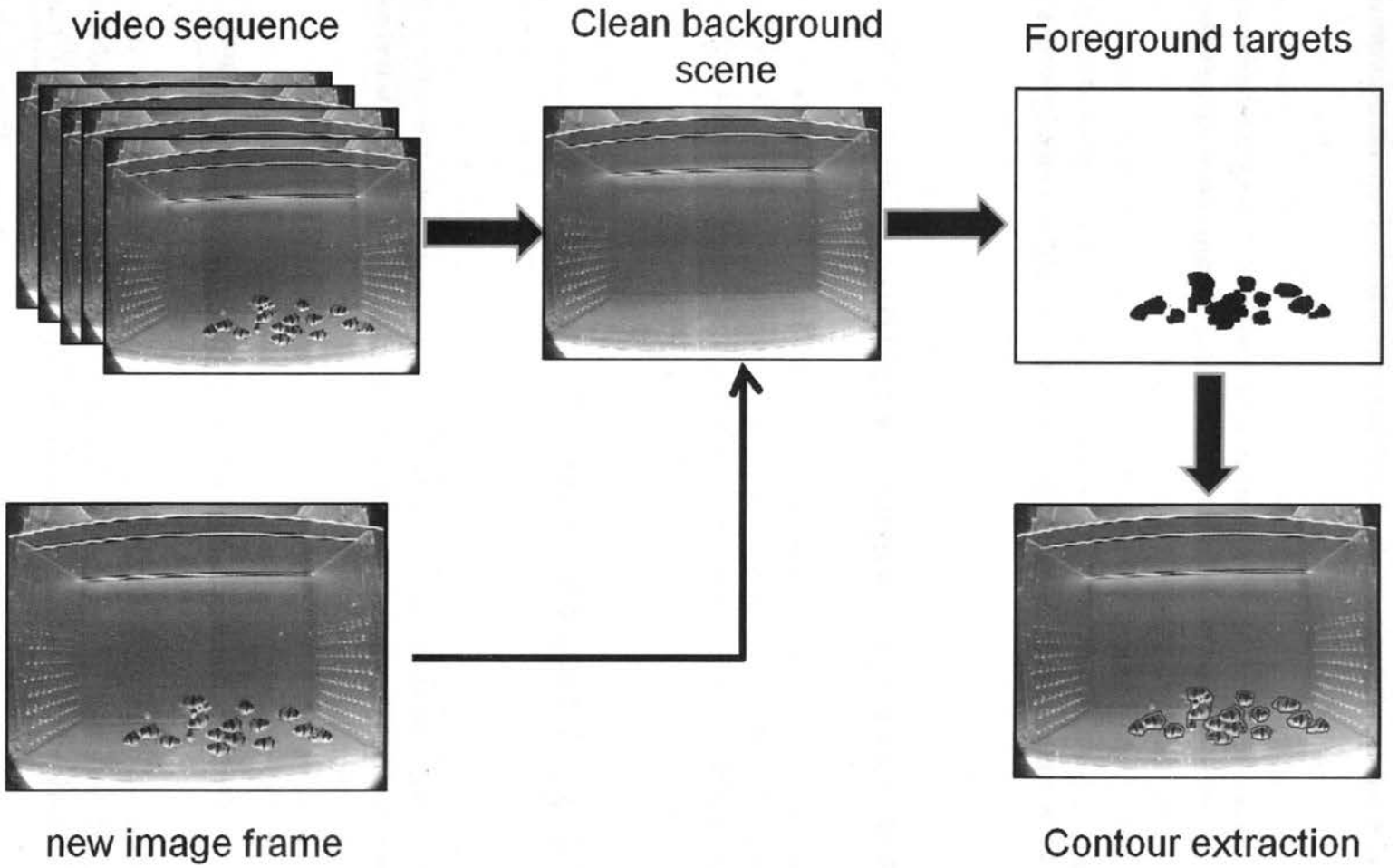


Figure B.1: An example of motion segmentation and contour extraction. The clean background scene is first generated from a set of training images containing moving and stationary targets. The contours of foreground regions are then extracted by a threshold-based background subtraction algorithm and a connected component analysis.

mogeneous colour regions by using a hierarchical k-means clustering, where each region is represented as a single Gaussian. Given a new video frame, the foreground targets are detected by computing the probability of a current pixel (i, j) belonging to the background colour model as:

$$p(\mathbf{I}(i,j)\|background) = \max_k \left[\exp\left\{ -\frac{1}{2}(\mathbf{I}(i,j) - \mu_k)\Sigma_k^{-1}(\mathbf{I}(i,j) - \mu_k) \right\} \right] \quad \text{(B.2)}$$

where μ_k and Σ_k denotes the mean and covariance matrix of each Gaussian component k. The probability of a pixel at (i, j) to be recognised as a foreground pixel is given as

$$p(\mathbf{I}(i,j)\|foreground) = 1 - p(\mathbf{I}(i,j)\|background) \quad \text{(B.3)}$$

Then, pixels with higher likelihood of containing foreground targets are extracted by a simple threshold-based algorithm and the pixels in close regions are grouped using a connected component analysis. Figure B.1 shows the process of extracting foreground regions and the corresponding contours. Given a video of a group of fish in a water tank, the clean background scene is first generated from a sequence of training frames containing moving and stationary targets (equation B.1). Then, the foreground regions (fishes) and the corresponding contours are extracted by a threshold-based background subtraction algorithm and a connect component analysis (equation B.3).

B.2.2 Silhouette Representation

In the next step, foreground regions are represented using an implicit function by computing a distance transform as given below:

$$\mathbf{x}(i,j) = \begin{cases} +d((i,j),C), & (i,j) \text{ lies outside C;} \\ 0, & (i,j) \text{ lies on the contour, C;} \\ -d((i,j),C), & (i,j) \text{ lies inside C,} \end{cases} \quad \text{(B.4)}$$

where $d((i,j),C)$ is the Euclidean distance between pixel (i, j) and the nearest pixel on the contour C as shown in Figure B.2.

This yields signed distance maps for corresponding frames as shown in Figure B.3. Darker colour (dark blue) to brighter colour (red) represents values of x increasing from the negative to the positive. Compared to a binary map representation of targets, the resulting representation enhances positions, motion and shape information of the targets

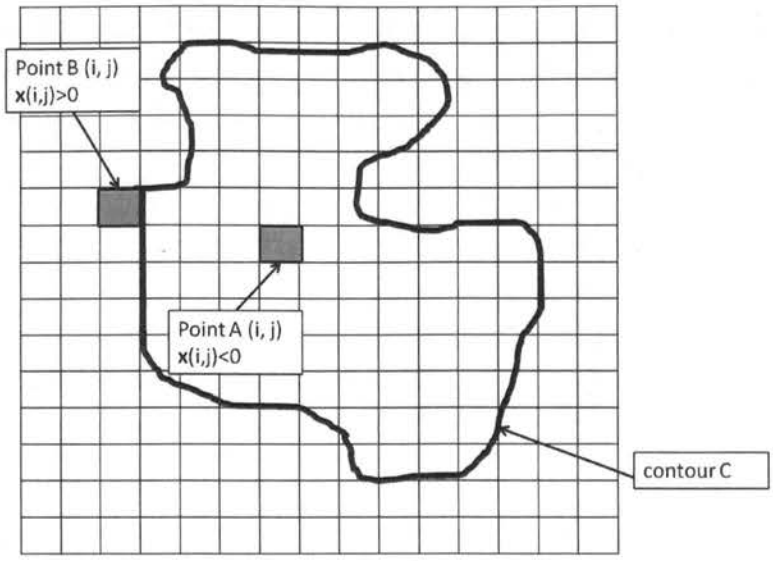


Figure B.2: Motion segmentation and contour extraction.

in the scene. The analysis of these signed distance maps, movements of blue-and-red patches and creation or destruction of blue patches, shows evolution of the foreground silhouettes.

B.3 Unsupervised Abnormality Detection

Given a sequence of video frames represented by signed distance maps, the next step is to extract a set of key video segments that represent informative contents of the source video and detect video segments that contain potentially dangerous events. In this method, a long video is first divided into short segments by studying the motion and appearance variation of targets in the scene. Then, each video segment is presented by a set of key representative frames. To detect abnormal events, the video segments are projected into a low dimensional space. Then, spatially isolated embedded data points corresponding to abnormal events are detected using a local density-based clustering algorithm. Figure B.4 shows a diagrammatic illustration of key components of the proposed abnormality detection method.

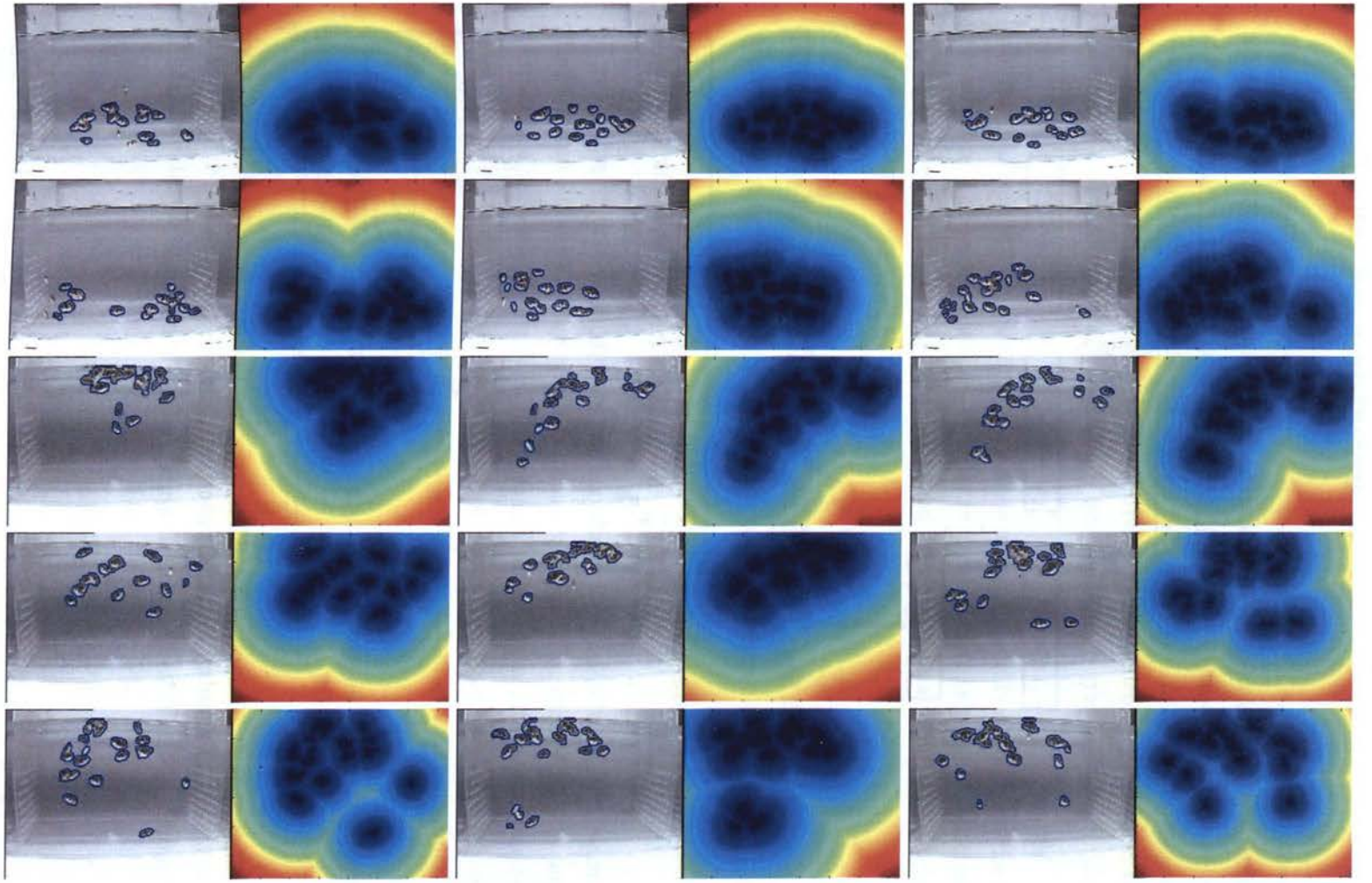


Figure B.3: Representation of a video as a series of signed distance maps. Darker colour (blue) to brighter colour (red) represents values of x increasing from the negative to the positive. This representation implicitly incorporates information such as positions, movement directions and the compactness (or dispersion) of the group over time.

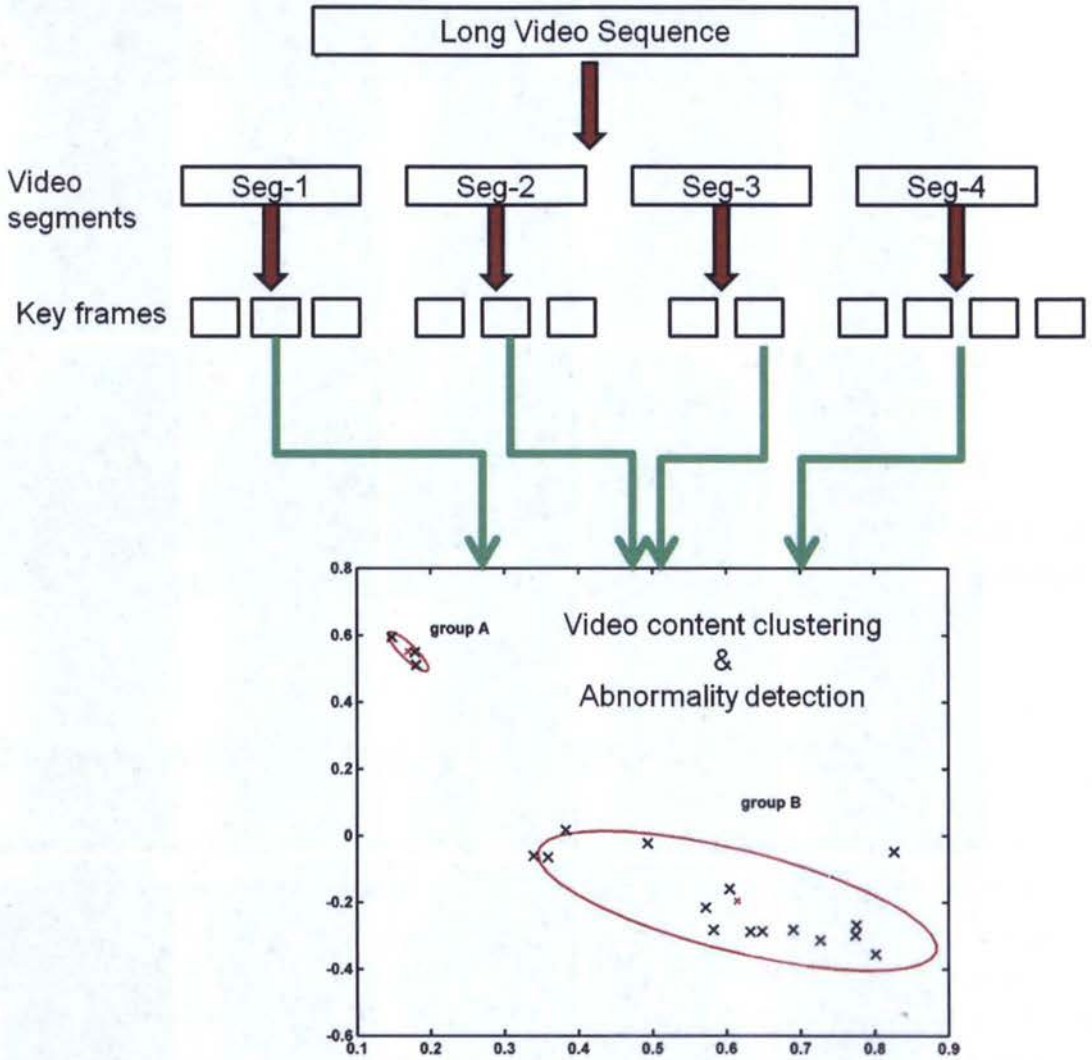


Figure B.4: A block diagram illustrating key components of the proposed abnormality detection method.

B.3.1 Video Segmentation

The objective is to segment a long video sequence into a set of short video segments such that, ideally, each segment contains a single event. In general, a video sequence is divided into segments by applying a sliding time widow with a fixed scale. However, this method is inefficient and determining the time scale is a non-trivial problem for many applications. Hence, in this method, an unsupervised video segmentation algorithm is developed to divide the video sequence by studying the motion and appearance variation of targets in the scene.

The first step is to study shape variations of targets and find time instants when there are significant changes in scenes. In order to check if there is a significant change at time t, the minimum distance between the signed distance map at time t and that of previous frames are first computed as:

$$\delta(\mathbf{x}_t) = \min_{\tau} d(\mathbf{x}_t, \mathbf{x}_{t-\tau}) \tag{B.5}$$

where $\mathbf{x}_t$ is the signed distance map representing image t and $d(\mathbf{x}_t, \mathbf{x}_{t-\tau})$ for $\tau = \{1, 2, \cdots, T\}$ is the distance between two signed distance maps at time t and $t - \tau$ and computed as:

$$d(\mathbf{x}_t, \mathbf{x}_{t-\tau}) = \frac{1}{M \times N} \sum_{(i,j)} \|\mathbf{x}_t(i,j) - \mathbf{x}_{t-\tau}(i,j)\| \tag{B.6}$$

where $i = 1 : N$ and $j = 1 : M$ are the coordinates of pixels in an image size of $M \times N$. Then, frame t can be considered as a change moment in time if $\delta(\mathbf{x}_t)$ is larger than a particular threshold. However, this approach suffers from over-segmentation in scenes where complex and changeable actions occur. In order to cope with the over-segmentation problem, statistical properties of scene changes over T frames are studied using a kernel-based density estimation.

Assuming that scene changes over T frames are given as $\{\delta_\tau\}_{\tau=1}^{T}$, the kernel density function is modelled as:

$$f(\delta) = \frac{1}{T} \sum_{\tau=1}^{T} K(\delta - \delta_\tau) \tag{B.7}$$

where $K(.)$ is a Gaussian kernel function centred at δ_τ for $\tau = \{1, 2, \cdots, T\}$. Then, the probability density function of δ_t (the change in time t) is estimated as follows:

$$p(\delta_t) = \frac{1}{T} \sum_{\tau=1}^{T} \exp[-(\delta_t - \delta_\tau)/(2\sigma^2)] \tag{B.8}$$

where σ is the standard variation of δ_τ over T frames. If $p(\delta_t)$ is less than a particular

threshold [2], the current frame t is defined as a change point.

B.3.2 Key-Frame Extraction

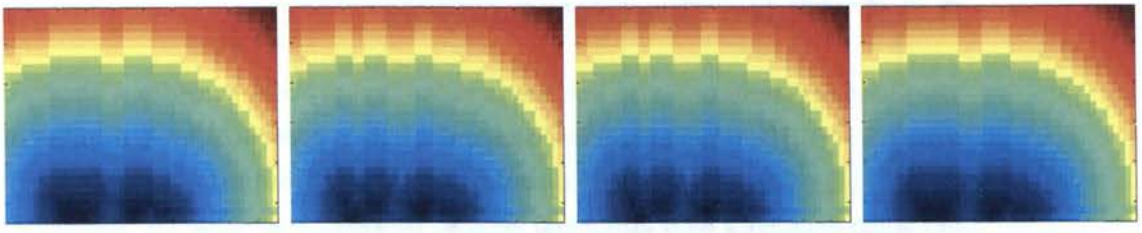
Given a short video segment, a spectral clustering method is employed to extract representative key frames. The objective is to extract a small set of key frames that can provide the most of informative contents of the given video segment. This approach starts by constructing a weighted neighbourhood graph W for a given video segment $\mathbf{S}_i = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_n]$ where n is the total number of frames in the segment $\mathbf{S}_i$. Here, the weight of the edge connecting $\mathbf{x}_p$ and $\mathbf{x}_q$, where $p, q \in [1, n]$ and $p \neq q$ is defined as:

$$w_{pq} = \exp\left\{-\frac{d_{pq}^2}{2\sigma_p\sigma_q}\right\}. \tag{B.9}$$

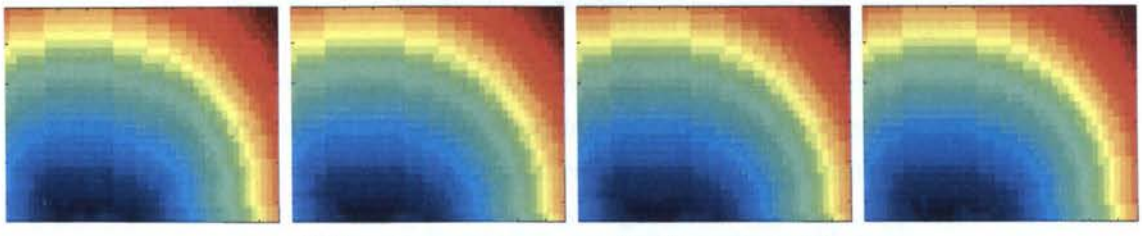
where d_{pq} is the distance measure between two signed-distance maps, $\mathbf{x}_p$ and $\mathbf{x}_q$ and parameters σ_p and σ_q are local variances computed at $\mathbf{x}_p$ and $\mathbf{x}_q$ respectively. To compute σ_p, distance measures of $\mathbf{x}_p$ with respect to its 10 nearest neighbouring points are first computed and then σ_p is defined as the variance of these distance measures. The same step is applied to compute σ_q.

Once the neighbourhood graph is obtained, the next step finds an optimum low-dimensional space Y such that $\mathbf{Y}_{opt} = \arg\min_{\mathbf{Y}}(\mathbf{Y}^T\mathbf{L}\mathbf{Y})$, subject to $\mathbf{Y}^T\mathbf{M}\mathbf{Y} = 1$. Equivalently, this is to solve a generalised eigenvalue problem of $\mathbf{L}\mathbf{y} = \lambda\mathbf{M}\mathbf{y}$, where $\mathbf{L} = \mathbf{M} - \mathbf{W}$ is the Laplacian graph of W and M is the diagonal weight matrix. Each diagonal entry of M is the sum of the corresponding row of W. The K-means clustering algorithm is then performed on the extracted eigenvectors to group the video frames into K clusters. The matrix perturbation theory [170] is employed to decide the value of K. The theory states that the number of clusters depends on the stability of eigenvalues which is determined by the gap δ_e between two consecutive eigenvalues. Based on this theory, the number K is defined by finding the maximum gap δ_e over a set of eigenvalues, i.e., $K = \arg\max_i \mid \lambda_i - \lambda_{i-1} \mid$, where λ_i and λ_{i-1} are two consecutive eigenvalues. Hence, the segment $\mathbf{S}_i$ can be represented by K key frames where each key frame is the median element of the corresponding cluster. Figure B.5 shows some key frames extracted from a long video sequence recorded a group of fish in a water tank. This sequence contains 1005 frames and the proposed approach generates 18 video segments where the number of key frames for each video segment is different.

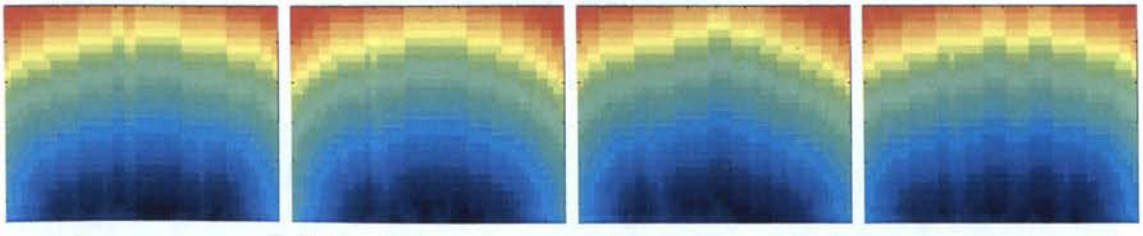
[2]The threshold value is selected empirically in the tested scenarios



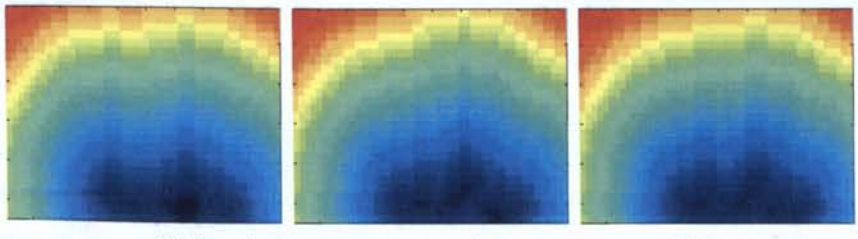
(a) key frames corresponding to segment 1



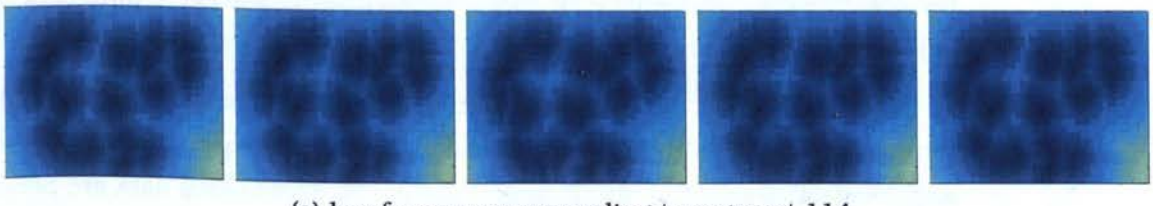
(b) key frames corresponding to segment 2



(c) key frames corresponding to segment 24



(d) key frames corresponding to segment 80



(e) key frames corresponding to segment 114

Figure B.5: Some example of key frames extracted from a long video sequence. The video is recorded a group of fish in a water tank and contains 1005 frames and the proposed approach generates 18 video segments where each segment has different number of key frames.

B.3.3 Video Content Analysis

The video segmentation and key frame extraction process provides a library of video segments $\mathbf{S} = \{\mathbf{S}_i\}_{i=1}^{N}$ where each video segment $\mathbf{S}_i$ is represented by K number of key frames $\mathbf{I}_1, \mathbf{I}_2, \cdots, \mathbf{I}_K$. The next step is to discover the natural grouping of video segments and identify video segments with abnormal events. However, conventional clustering approaches such as $k-$means and Gaussian mixture models cannot be applied directly as each video segment is of different lengths.

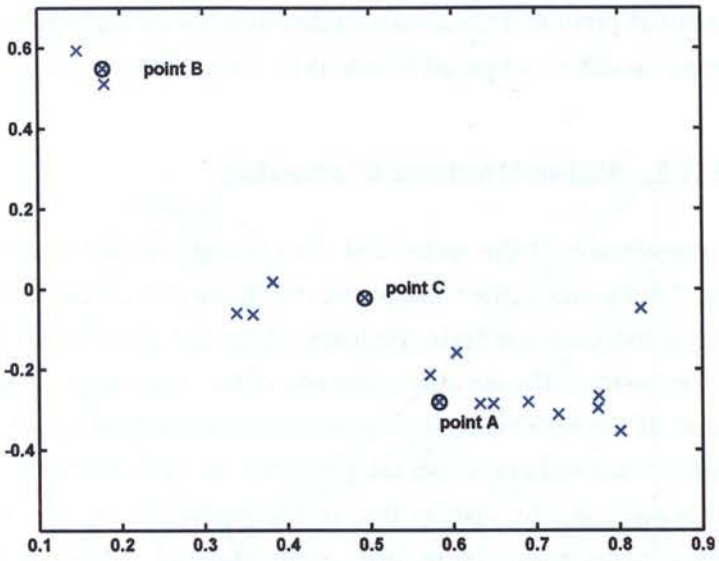
In order to perform effective clustering, high dimensional video segments are first projected into a low dimensional space using the Laplacian eigenmaps (LE). As discussed in Chapter 4 (please refer to Section 4.2), the first step of LE is to measure the affinity between different video segments. Since the video segments are of different lengths, in this method, a modified Hausdorff distance measure [3] is considered to compute the affinity matrix. Specifically, the distance between two video segments is given by:

$$\begin{aligned} d(\mathbf{S}_1, \mathbf{S}_2) &= \min(d(\mathbf{S}_1, \mathbf{S}_2), d(\mathbf{S}_2, \mathbf{S}_1)) \\ d(\mathbf{S}_1, \mathbf{S}_2) &= \frac{1}{K_1} \sum_{i=1}^{K_1} \min_{j \in (1:K_2)} d(\mathbf{I}_i, \mathbf{I}_j) \\ d(\mathbf{S}_2, \mathbf{S}_1) &= \frac{1}{K_2} \sum_{j=1}^{K_2} \min_{i \in (1:K_1)} d(\mathbf{I}_j, \mathbf{I}_i) \end{aligned} \tag{B.10}$$

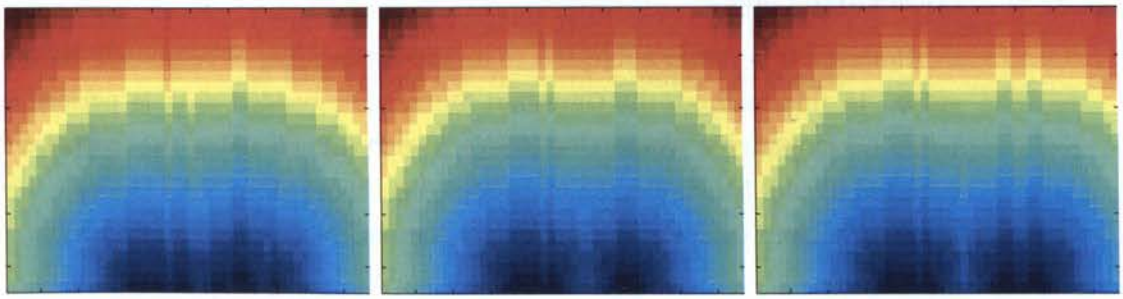
where $\mathbf{I}_i, i \in [1, K_1]$ and $\mathbf{I}_j, j \in [1, K_2]$ are key frames for video segments $\mathbf{S}_1$ and $\mathbf{S}_2$ respectively.

Given the $N \times N$ affinity matrix, LE finds a low dimensional representation of segments using the normalised Laplacian matrix. This provides a new representation of the long video sequence $\mathbf{V}_s = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N\}$ where each segment $\mathbf{v}$ is represented by k_s eigenvectors where k_s is automatically selected based on the relative difference between two adjacent eigenvalues.

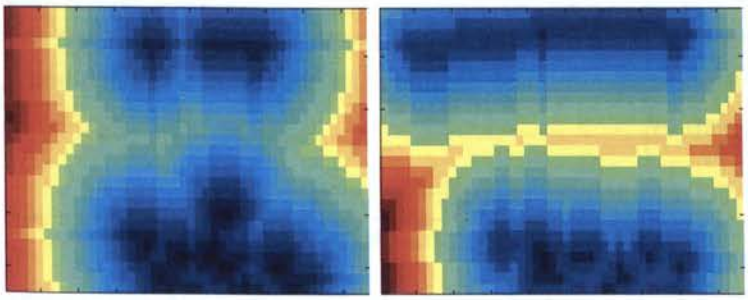
Figure B.6 shows the embedded video data in a 2−dimensional space and key frames for corresponding video segments of a long video sequence. For ease of visualisation and discussion, only the first 2 dimensions of embedded video data are presented. Each point in a low dimensional space given in (a) corresponds to a short video segments where each of them is represented by different number of key frames. In (b-d), key frames corresponding to points A, B and C are shown. It can be observed that points A and C are located nearer to each other in the low dimensional space as video segments corresponding to points A and C demonstrate closer similarity compared to the video



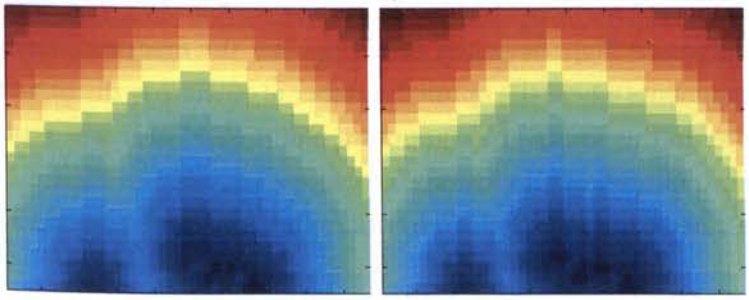
(a) 2D low dimensional space



(b) key frames corresponding to point A



(c) key frames corresponding to point B



(d) key frames corresponding to point C

Figure B.6: The embedded video data in a low dimensional space and key frames for video segments corresponding to point A, B and C.

segment at point B. This demonstrates that the LE embeds video segments with different events in a different spatial locations in the embedded space.

B.3.3.1 Video Content Clustering

The distribution of the embedded data provides a way to understand the context of the video. For instance, the video shown in Figure B.6 contains the swimming behaviour of a group of fish over one hour. Typically, these fish swim in the lower part of the water tank but they swim to the surface at the end of the video due to lack of oxygen in the tank. The content of the video is well reflected in the embedded space where video segments with abnormal movements of fish are projected far away from the most of the video segments. In this method, the distribution of the embedded data is modelled using a Gaussian mixture model where the number of components C determines the number of behaviour classes in the video.

However, it is rather subjective to decide the number of behaviour classes for a given video. In this method, the Gaussian fitting of embedded data is performed iteratively using C components where $C \in [2, N/5]$ is determined using the Bayesian information criterion (BIC) [134]. Given the embedded data of the video $\mathbf{V}_s = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N\}$, the BIC score for a given Gaussian model with C components can be computed as:

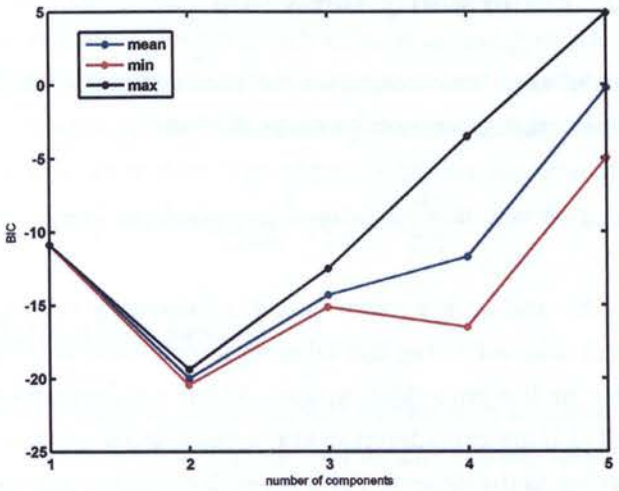
$$BIC(\Theta) = -\log p(\mathbf{V}|\Theta) + C_\Theta \log(N). \tag{B.11}$$

where k_Θ is the number of free parameters in the mixture model and N is total number of video segments. The likelihood of observing the video using C number of components is given by

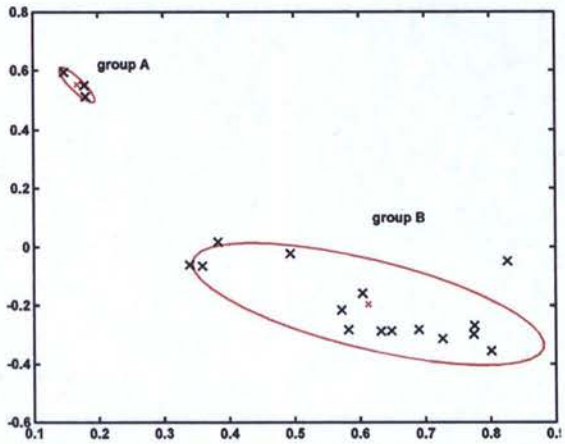
$$p(\mathbf{V}|\Theta) = \sum_{n=1}^{N}\sum_{r=1}^{C} \omega_r \frac{1}{(2\pi)^{k_s/2}|\Sigma_r|^{1/2}} \exp\Big\{-\frac{1}{2}(\mathbf{v}_n - \mu_r)^T \Sigma_r^{-1}(\mathbf{v}_n - \mu_r)\Big\} \tag{B.12}$$

where μ_r, Σ_r and ω_r are mean, co-variance matrix and weight of the r Gaussian component and k_s is the dimension of the video segment v in the embedded space.

Figure B.7(a) shows the BIC scores for the mixture model using different number of components where the optimal number of clusters is the one that gives the minimum BIC score. In this study, the computation of the BIC scores is repeated 5 times and the maximum, minimum and average scores are provided in Figure B.7(a). Figure B.7(b) shows the video context in 2 clusters where the first cluster (group A) describes video segments



(a) BIC scores for different number of components.



(b) grouping video content in 2 clusters.

Figure B.7: A simple illustration for video content clustering by Gaussian mixture model. (a) the BIC scores for different number of components. The minimum score gives the optimal number of components (b) the Gaussian fitting of the embedded data using 2 components.

with abnormal events while the second one (group B) presents segments corresponding to normal events.

B.3.3.2 Abnormality Detection

Given the labelled information for normal events, the likelihood of a new segment belonging to the normal group can be computed as:

$$p(\mathbf{v}_{new}|normal) = \sum_{r=1}^{C} \omega_r \frac{1}{(2\pi)^{k_s/2}|\Sigma_r|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{v}_{new} - \mu_r)^T \Sigma_r^{-1}(\mathbf{v}_{new} - \mu_r)\} \quad \text{(B.13)}$$

where μ_r, Σ_r and ω_r are mean vector, co-variance matrix and weight of the r Gaussian component learned using the labelled information and k_s is the dimension of the video segment v in the embedded space. Video segments which have a lower score than a threshold $[0, 1]$ are considered as abnormal video segments. The threshold value should be set according to the detection and false alarm rate required by each particular application.

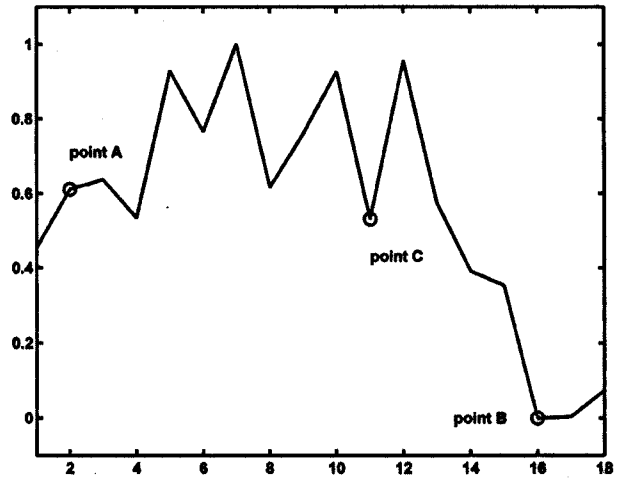


Figure B.8: The normality scores of video segments. The scores for video segments corresponding to point A, B and C are marked with circles.

Figure B.8 shows the normality scores for video segments shown in Figure B.6 and B.7(b). In this experiment, half of video segments with normal events (group B) are employed to model the normal behaviour while the rest are used as testing segments. It can be observed that point B (abnormal segment) has a lower score compared to point C from the normal group.

B.4 Experimental Results

This experiment aims to provide the understanding of the behaviour of living organisms in the water security and environment domain. The objective is to provide information to the public safety team to predict trends of fish behaviour and prepare actions for unusual events. It is observed that the behaviour of fish tends to change when they are exposed to contaminated water. Traditionally, human observers are assigned to monitor and report on any abnormal movements of fish. The objective of this experiment is to evaluate the performance of the proposed method on detecting the status of fish.

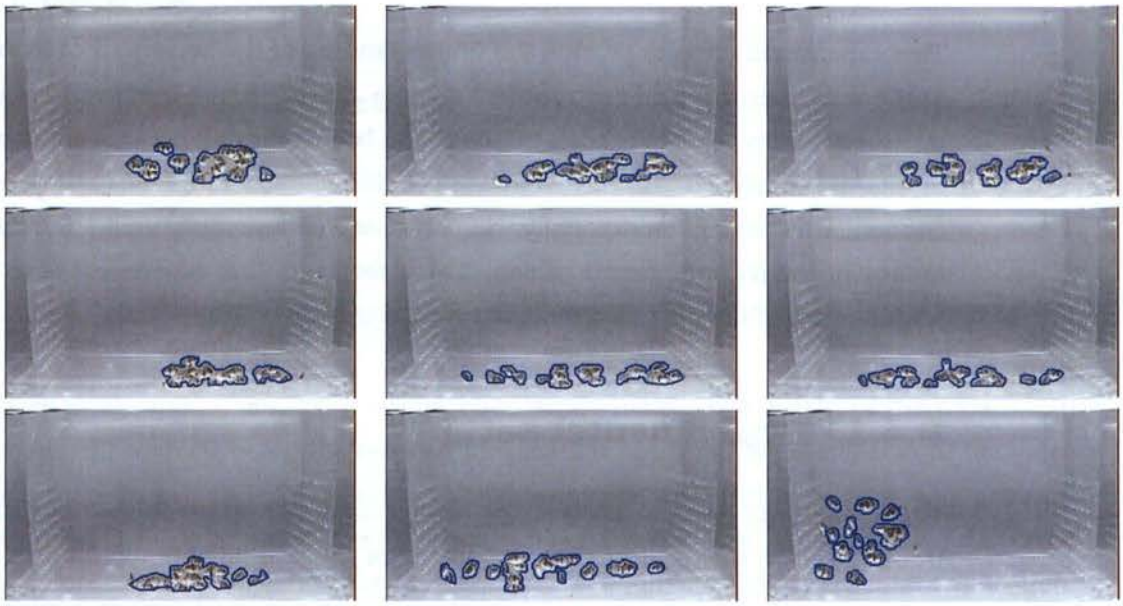
B.4.1 Experimental Setup

This experiment aims to analyse the behaviour of fish in a confined environment. Figure B.9 shows the experimental apparatus of the fish activity monitoring system. The

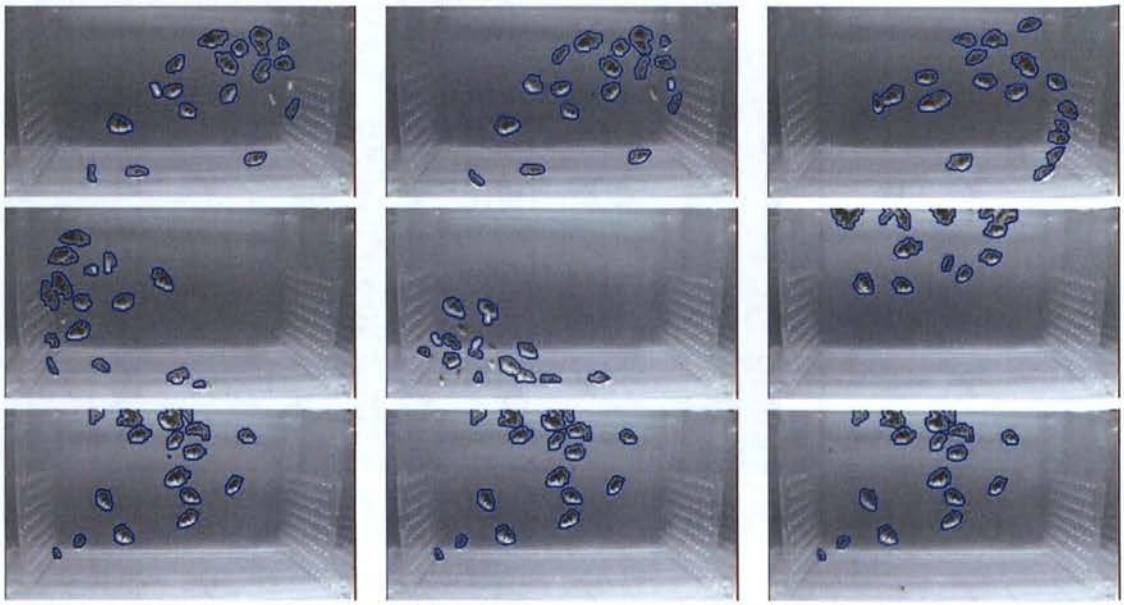
Figure B.9: Experimental apparatus of the fish activity monitoring system.

apparatus consists of a water tank, two cameras and an automatic monitoring and quantification system using computer vision techniques. Two cameras are installed to capture the fish from a front view and a top view. In this experiment, the video data from the front view is employed to study the behaviour of fish.

Videos are recorded at a resolution of 288×384 pixels at 6 frames per second under four different scenarios. In the first scenario, a group of 20 fish are kept in clean water for 12



(a) sample images of fish group swimming in the tank with clean water



(b) sample images of fish group swimming in the tank with cyanide

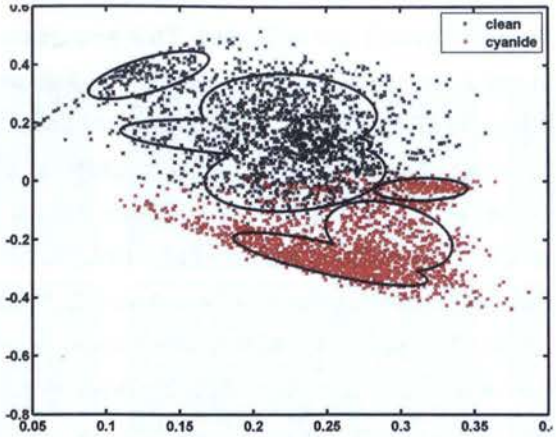
Figure B.10: Some example images of fish in different water contaminations. (a) Sample images from the video of fish in clean water (b) sample images of fish group swimming in the tank with cyanide.

hours and their behaviour are collected. This sequence contains 294,603 images with the frame size of 288×384. In other three scenarios, fish are exposed to water contaminated with different types of chemicals: chloramine, aldicarb (a concentration of $1.5mg/L$) and cyanide (a concentration of $150\mu g/L$). The duration of the video recording for the tank with chloramine is set to 3 hours (total of 63,397 images) as, with chloramine in the tank, fish are bound to die after about 3 hours. The duration of the recording for the other tanks with contaminated water are set to 9 hours and 12 hours respectively (202,074 images for cyanide and 262,016 images for aldicarb). Figure B.10 illustrates some sample images from two video sequences. The first three rows show sample images from the video of fish in clean water while the rest shows sample images of fish group swimming in the tank with cyanide.

B.4.2 Video Content Understanding

This experiment presents the analysis of four video sequences of a group of fish in four different tanks with different contaminations. The first sequence of fish swimming in clean water is noted as "clean sequence" while the second, third and fourth sequences are referred as 'cyanide', 'chloramine' and 'aldicarb' sequences respectively. The aim of this experiment is to analyse whether fish from tanks with contaminants exhibit different behaviour from the normal situation when they are kept in clean water. The unsupervised temporal segmentation approach described in Section B.3.1 divides the 'clean sequence' and the cyanide, chloramine and aldicarb sequences into 1795, 1812, 416 and 846 segments respectively.

Figure B.11 shows the low dimensional representations of the clean and cyanide sequence in the same embedded space. For ease of visualisation and discussion, only the first 2 dimensions of the video are presented here. It can be observed that the presence of contamination has spiked the subjects and they have shown different group behaviours in response to the contamination. The distribution of embedded data indicates that the normal and contaminated scenarios are separable by observing the Gaussian mixture models. The Gaussian mixture model clusters video contents into different groups where the number of clusters for each scenario are automatically determined based on the BIC scores. In Figure B.11, the first four clusters mainly contain samples corresponding to the 'clean sequence' while the last three clusters contain samples from cyanide sequences. Similarly, the comparison of the behaviour of fish in clean water against the behaviour of fish in contaminated water with chloramine and aldicarb are also shown in Figure B.12(a)



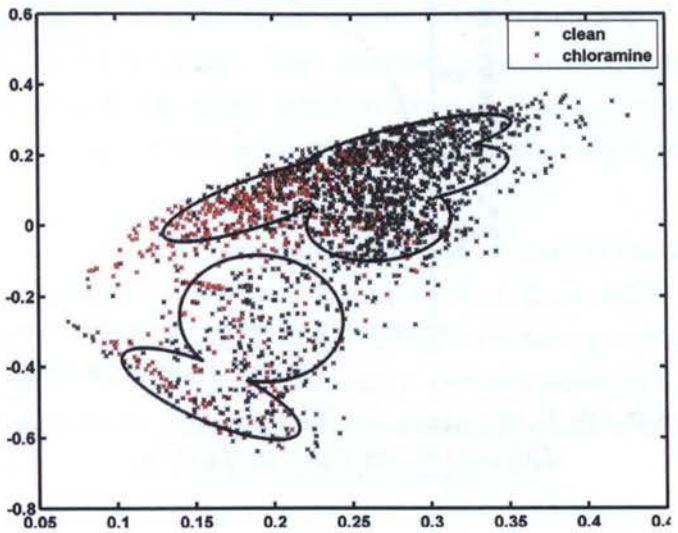
(a) clean vs. cyanide

Figure B.11: Unsupervised video content clustering of clean and contaminated sequences (cyanide) where the number of clusters is automatically estimated as 7. The video segments corresponding to 'clean sequence' are marked with black colour while the video segments corresponding to segments from the video with contaminated water are marked with red colour.

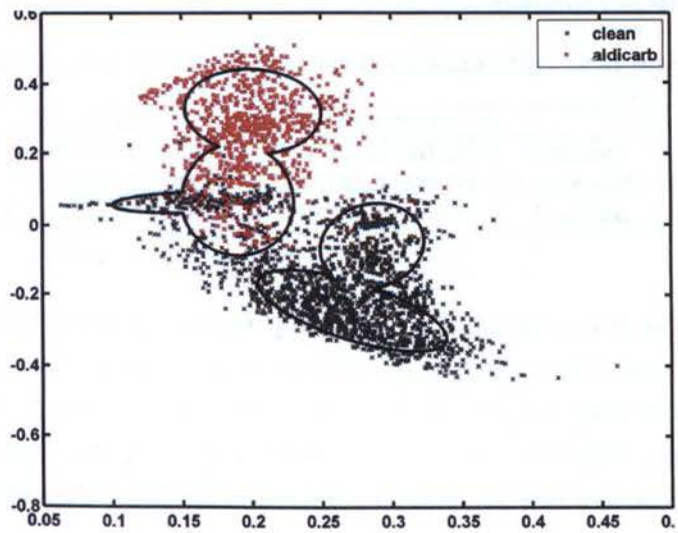
and B.12(b) respectively. It can be observed that the proposed approach provides a way to differentiate different fish behaviours in a reduced space.

B.4.3 Abnormality Detection

This experiment presents the performance of the proposed method on water toxicity detection using the fish behaviour. Given the embedded data shown in Figure B.11 and B.12, the regular behaviour of fish for each scenario is modelled using 1000 segments which are randomly selected from the 'clean sequence'. Then, the rest of the 'clean sequence' and the video segments from each 'contaminated' sequence are used as a testing set. This process is repeated for 20 trials where the training set is randomly selected in each trail. Figure B.13 shows the performance of the proposed method on three different scenarios of FISH data set. The detection rate and false alarm rate are shown in the form of a receiver operating characteristic (ROC) curve by varying the abnormality detection threshold. It is important to minimise the false positive rate (FPR) while maximising the true positive rate (TPR). This is because the system should be able to detect water toxicity accurately, if possible, 100% as this leads to health issue. On the other hand, the system should minimise the FPR for economic reason as stopping the water plant for manual



(a) clean vs. chloramine



(b) clean vs. aldicarb

Figure B.12: Unsupervised video content clustering of clean and contaminated sequences ((a) chloramine and (b) aldicarb) where the number of clusters is automatically estimated as 5. The video segments corresponding to 'clean sequence' are marked with black colour while the video segments corresponding to segments from the video with contaminated water are marked with red colour.

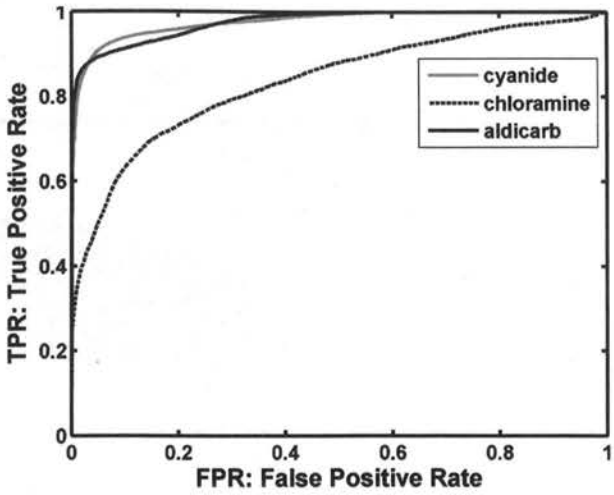


Figure B.13: ROC curves for the detection of abnormal group behaviours under different water contaminations.

analysis is expensive.

Table B.1: Abnormality detection rate and false alarm rate for fish data set

	accuracy(%)	detection rate (%)	false alarm rate (%)
cyanide	92.62±0.0070%	92.62±0.007%	7.38±0.0072%
chloramine	63.99±0.0024%	75.99±0.0035%	24.01±0.0034%
aldicarb	90.69±0.0027%	90.65±0.0026%	9.29±0.0028%

Table B.1 summaries the resulting statistics of the proposed method on detecting the behaviour of fish exposed to different contaminants. Here, 'cyanide', 'chloramine' and 'aldicarb' refer to the condition where fish are exposed to the contaminated water with cyanide, chloramine and aldicarb respectively. Results show that this method provides a high accuracy rate for detecting the abnormal behaviour of fish exposed to 'cyanide' and 'aldicarb'. The lower accuracy rate for detecting chloramine can be explained by the similar behaviour exhibited by fish in clean water and contaminated water with chloramine as shown in Figure B.12(a).

B.5 Summary

This chapter presented a new approach for detecting abnormal group events in a long video sequence. The shape and motion of all moving targets in the scene are combined to

capture the properties of a group as a whole. This provides an efficient representation of targets without the need to detect and track individual targets. An unsupervised temporal segmentation divides a long video into short segments where each segment is represented by a different number of key frames. Then, the normal behaviour of the group is modelled using GMM with automatic model selection based on Bayesian information criterion score. This provides a way to detect video segments with potentially dangerous abnormal activities.

Experimental results have demonstrated that this method is capable to learn and recognise group behaviours of a school of fish under normal as well as scenarios where they are exposed to chemical contamination. The low dimensional representation of video sequences provides an effective way to differentiate between different behaviours of fish. Experiments have shown the potential of the proposed approach to be used as an early warning system for detecting contamination in drinking water.

Publications

[**ThidaVS09**] M. Thida, P. Remagnino, H.-L. Eng, A particle swarm optimisation approach for multi-objects tracking in crowded scene, in: proceedings of IEEE International Workshop on Visual Surveillance, 2009, pp. 1209 - 1215.

[**ThidaMVA09**] M. Thid, H.-L. Eng, and Bong Fong. Chew, "Automatic analysis of fishes behaviour and abnormality detection," in In proceedings of IAPR Conference on Machine Vision Applications, May 2009, pp. 278-282.

[**Thida2MVA09**] Bong Fong. Chew, H.-L. Eng, and M. Thida, "Vision-based real-time monitoring on the behaviour of fish school," in In proceedings of IAPR Conference on Machine Vision Applications, May 2009, pp.90–94.

[**ThidaACCV10**] M. Thida, H.-L. Eng, D. N. Monekosso, and P. Remagnino, "Learning video manifold for segmenting crowd events and abnormality detection," in Lecture Notes in Computer Science: Computer Vision, vol. 6492/2011. Springer-Verlag, November 2010, pp. 439–449.

[**ThidaACM10**] H. Lu, H.-L. Eng, M. Thida, and K. Plataniotis, "Visualisation and clustering of crowd video content in MPCA subspace," in ACM Conference on Information and Knowledge Management, October 2010, pp. 1777–1780.

[**ThidaISSPR11**] M. Thida and H.-L. Eng, "Learning group behaviour : Detecting water toxicity by biological monitoring," in International Summer School on Pattern Recognition, vol. Springer: Best Poster Award, 2011.

[**ThidaCVA12**] M. Thida, H.-L. Eng, D. N. Monekosso, and P. Remagnino, "Learning video manifolds for content analysis of crowded scenes," Information Processing Society of Japan (IPSJ) Transactions on Computer Vision and Applications, vol. 4, pp. 71–77, May 2012.

[**ThidaASC12**] M. Thida, H.-L. Eng, D. N. Monekosso, and P. Remagnino, A particle swarm optimisation algorithm with interactive swarms for tracking multiple targets, Applied Soft Computing, vol. 13, pp. 3106-3117, 2013.

[**ThidaSMC12**] M. Thida, H.-L. Eng, and P. Remagnino, Laplacian Eigenmap with Temporal Constraints for Local Abnormality Detection in Crowded Scenes, in IEEE Transactions on Systems, Man, And Cybernetics Part B, vol. 99, pp. 1–10, February 2013.

Bibliography

[1] Bio-sensor: Water security and monitoring system. http://www.biomon.com/biosenso.html.

[2] CAVIAR data-set. http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/.

[3] Hausdorff: Haudsdorff distance measure. http://en.wikipedia.org/wiki/Hausdorff-distance.

[4] Toxprotect 64 - the online instrument for drinking water protection. http://www.bbe-moldaenke.de/toxicity/toxprotect64/.

[5] UCSD anomaly detection dataset. http://www.svcl.ucsd.edu/projects/anomaly.

[6] Unusual crowd activity dataset. http://mha.cs.umn.edu.

[7] CLEAR evaluation. http://clear-evaluation.org/, 2007.

[8] Particle filter colour tracker. http://www.mathworks.com/matlabcentral/fileexchange/17960, 2007.

[9] PETS. http://www.cvg.rdg.ac.uk/PETS2009, 2009.

[10] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and David Reinitz. Robust Realtime Unusual Event Detection using Multiple Fixed-Location Monitors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):555–560, March 2008.

[11] F. J. Aherne, N. A. Thacker, and P. Rockett. The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 34(4):363–368, 1988.

[12] Irshad Ali and Matthe. N. Dailey. Multiple Human Tracking in High-density Crowds. In *Advanced Concepts in Intelligent Vision Systems*, pages 540–549, 2009.

[13] Saad Ali and Mubarak Shah. A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2007.

[14] Saad Ali and Mubarak Shah. Floor Fields for Tracking in High Density Crowd Scenes. In *Proceedings of European Conference on Computer Vision*, pages 1–14, 2008.

[15] Yilmaz Alper, Javed Omar, and Shah Mubarak. Object tracking: A Survey. *ACM Computing Surveys*, 38(4):13–58, 2006.

[16] Ernesto Andrade and Robert Fisher. Simulation of Crowd problems for Computer Vision. In *Proceedings of 19th International Conference on Pattern Recognition*, volume 3, pages 71–80, November 2005.

[17] Ernesto Andrade, Robert Fisher, and Scott Blunsden. Modelling Crowd Scenes for Event Detection. In *Proceedings of 19th International Conference on Pattern Recognition*, volume 1, pages 175–178, September 2006.

[18] Anton Andriyenko, Stefan Roth, and Konrad Schindler. An analytical formulation of global occlusion reasoning for multi-target tracking. In *Proceedings of IEEE International Workshop on Visual Surveillance*, pages 1839 – 1846, Barcelona, November 2011. IEEE.

[19] Anton Andriyenko and Konrad Schindler. Multi-target tracking by continuous energy minimization. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1265–1272, Colorado, June 2011. IEEE.

[20] Stefan Anitei. Fishes used against terrorist attacks. http://news.softpedia.com/news/Fishes-Used-Against-Terrorist-Attacks-36818.shtml.

[21] Luis Anton-Canalis, Mario Hernandez-Tejera, and Elena Sanchez-Nielsen. SWARMTRACK: A particle swarm approach to visual tracking. In *Proceedings of the International Conference on Computer Vision Theory and Applications*, volume 2, pages 221–228, Portugal, February 2006.

[22] Gianluca Antonini, Santiago Venegas Martinez, Michel Bierlaire, and Jean Philippe Thiran. Behavioral Priors for Detection and Tracking of Pedestrians in Video Sequences. *International Journal on Computer Vision*, 69(2):159–180, August 1998.

[23] Mittal Anurag and Davis Larry S. M2Tracker: A Multi-View Approach to Segmenting and Tracking People in a Cluttered Scene. *International Journal of Computer Vision*, 51(3):189–203, February 2003.

[24] M. S. Arulampalam, N. Gordon S. Maskell, 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.

[25] K. Bai. Particle filter tracking with Mean Shift and joint probability data association. In *Image Analysis and Signal Processing (IASP), 2010 International Conference on*, pages 607–612. IEEE, 2010.

[26] 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.

[27] Arslan Basharat, Alexei Gritai, and Mubarak Shah. Learning Object Motion Patterns for Anomaly Detection and Improved Object Detection. In *Proceedings of Computer Vision and Pattern Recognition*, pages 1–8, Anchorage, June 2008. IEEE.

[28] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, June 2003.

[29] Yassine Benabbas, Nacim Ihaddadene, and Chabane Djeraba. Global Analysis of Motion Vectors for Event Detection in Crowd Scenes. In *Proceedings of Eleventh International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 109–116, USA, June 2009. IEEE.

[30] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3457–3464, June 2011.

[31] Jérôme Berclaz, François Fleuret, Engin Türetken, and Pascal Fua. Multiple object tracking using K-shortest paths optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806 – 1819, September 2011.

[32] Michel Bierlaire, Gianluca Antonini, and Mats Weber. Behavioural Dynamics for Pedestrians. *Lecture Notes in Computer Science: Moving through nets: the physical and social dimensions of travel*, pages 81–105, August 2003.

[33] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as Space-Time Shapes. In *The Tenth IEEE International Conference on Computer Vision*, pages 1395–1402, 2005.

[34] Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Robust tracking-by-detection using a detector confidence particle filter. In *IEEE International Conference on Computer Vision*, pages 1515–1522, October 2009.

[35] William Brendel and Sinisa Todorovic. Learning spatiotemporal graphs of human activities. In *Proceedings of IEEE International Conference on Computer Vision*, pages 778–785. IEEE, 2011.

[36] Franois Brmond, Monique Thonnat, and Marcos Zniga. Video-understanding Framework for Automatic Behavior Recognition. *Behavior Research Methods*, 3(38):416–426, 2006.

[37] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, volume 3024 of *LNCS*, pages 25–36, Prague, May 2004. Springer.

[38] Hilary Buxton and Shaogang Gong. Visual Surveillance in a Dynamic andUuncertain world. *Artificial Intelligen*, 78(1-2):431–459, October 1995.

[39] Yizheng Cai, Nando de Freitas, and James J. Little. Robust Visual Tracking for Multiple Targets. In *Proceedings of Eighth European Conference on Computer Vision*, volume 3954, pages 107–118, 2006.

[40] Antoni B. Chan, Mulloy Morrow, and Nuno Vasconcelos. Analysis of Crowded Scenes Using Holistic Properties. In *Proceedings of Eleventh International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 101–108, USA, June 2009. IEEE.

[41] Zhongwei Cheng, Lei Qin, Qingming Huang, Shuqiang Jiang, and Qi Tian. Group activity recognition by gaussian processes estimation. In *International Conference on Pattern Recognition*, pages 3228 – 3231, August 2010.

[42] Maurice Clerc and James Kennedy. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, February 2002.

[43] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 142–151, 2000.

[44] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[45] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 3449 – 3456, June 2011.

[46] Trevor F. Cox and Michael A. A. Cox. *Multidimensional Scaling*. Chapman and Hall, second edition, 2001.

[47] Xinyi Cui, Qingshan Liu, Mingchen Gao, and Dimitris N. Metaxas. Abnormal detection using interaction energy potentials. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3161–3167, 2011.

[48] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[49] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[50] Lei Ding, Quan fu Fan, Jen-Hao Hsiao, and Sharath Pankanti. Graph based event detection from realistic videos using weak feature correspondence. In *Proceedings of IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1262–1265. IEEE, 2010.

[51] Shiloh L. Dockstader and A. Murat Tekalp. Multiple Camera Tracking of Interacting and Occluded Human Motion. *Proceedings of the IEEE*, 89(10):1441 –1455, 2001.

[52] How-Lung Eng. Let the fishes tell you if your water is safe. *Innovation Magazine: Environmental and Climate Change*, 9(1):46–47, 2010.

[53] Robert W. Floyd. Algorithm 97: Shortest path. *Communications of the Associate for Computing MachineryACM*, 5(6):345–350, 1962.

[54] Robert W. Floyd. On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19, 2002.

[55] A.P. French, A. Naeem, I.L. Dryden, and T.P. Pridmore. Using social effects to guide tracking in complex scenes. In *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance*, pages 212–217, 2007.

[56] Carolina Garate, Piotr Bilinski, and Francois Bremond. Crowd Event Recognition Using HOG Tracker. In *Proceedings of Eleventh International Workshop on Performance Evaluation of Tracking and Surveillance*, pages 1–6. IEEE, December 2009.

[57] Andrew Gilbert and Richard Bowden. Multi Person Tracking within Crowded Scenes. In *Proceedings of Workshop on Human Motion*, pages 166–179, 2007.

[58] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as Space-Time Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, December 2007.

[59] James Hafner, Harpreet S. Sawhney, Will Equitz, Myron Flicker, and Wayne Niblack. Efficient colour histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, July 1995.

[60] Dirk Helbing and Pter Molnr. Social Force Model for Pedestrian Dynamics. *Physical Review E*, 51(5):4282–4286, May 1995.

[61] Somboon Hongeng and Ramakant Nevati. Multi-agent Event Recognition. In *Proceedings of Eighth IEEE International Conference on Computer Vision*, volume 2, pages 84–91, Canada, July 2001. IEEE.

[62] Anthony Hoogs, Steve Bush, Glen Brooksby, A. G. Amitha Perera, Mark Dausch, and Nils Krahnstoever. Detecting Semantic Group Activities Using Relational Clustering. In *Proceedings of IEEE Workshop on Motion and Video Computing*, pages 1–8, Colorado, January 2008.

[63] Berthold K. P. Horn and Brian G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[64] Shaobo Hou, Aphrodite Galata, Fabrice Caillette, Neil Thacker, and Paul Bromiley. Real-time body tracking using a gaussian process latent variable model. In *Proceedings of International Conference on Computer Vision*, pages 1–8, 2007.

[65] Min Hu, Saad Ali, and Mubarak Shah. Detecting Global Motion Patterns in Complex Videos. In *Proceedings of International Conference on Pattern Recognition*, pages 1–5, Florida, 2008. IEEE.

[66] Min Hu, Saad Ali, and Mubarak Shah. Learning Motion Patterns in Crowded Scenes Using Motion Flow Field. In *Proceedings of International Conference on Pattern Recognition*, pages 1–5, Florida, 2008. IEEE.

[67] N. Hu, H. Bouma, and M. Worring. Tracking individuals in surveillance video of a high-density crowd. In *Proceedings of SPIE*, volume 8399, page 839909, 2012.

[68] Weiming Hu, Xuejuan Xiao, Zhouyu Fu, Xie Dan, Tieniu Tan, and Maybank Steve. A System for Learning Statistical Motion Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, September 2006.

[69] Yuxiao Hu, Liangliang Cao, Fengjun Lv, Shuicheng Yan, Yihong Gong, and Thomas S. Huang. Action Detection in Complex Scenes with Spatial and Temporal Ambiguities. In *Proceedings of IEEE International Conference on Computer Vision*, pages 128 – 135, Kyoto, October 2009.

[70] Michael Isard and Andrew Blake. CONDENSATION conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[71] Odest Chadwicke Jenkins and Maja Mataric. Spatio-temporal isomap: A time-series extension to isomap dimension reduction. *Proceedings of International conference on Machine learning*, pages 441–448, 2004.

[72] Peng Jia, Junsong Yin, Xinsheng Huang, and Dewen Hu. Incremental Laplacian Eigenmaps by Preserving Adjacent Information between Data Points. *Pattern Recognition Letters*, 30(16):1457–1463, December 2009.

[73] Fan Jiang, Ying Wu, and Aggelos K. Katsaggelos. A dynamic hierarchical clustering method for trajectory-based unusual video event detection. *IEEE Transactions on Image Processing*, 18(4):907–913, 2009.

[74] Neil Johnson and David Hogg. Learning the Distribution of Object Trajectories for Event Recognition. *Image and Vision Computing*, 14(8):583–592, August 1996.

[75] Yan Ke, Rahul Sukthankar, and Martial Hebert. Event Detection in Crowded Videos. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1–8, October 2007.

[76] Yan Ke, Rahul Sukthankar, and Martial Hebert. Spatio-temporal Shape and Flow Correlation for Action Recognition. In *Proceedings of International Workshop on Visual Surveillance*, pages 1 – 8, June 2007.

[77] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks.*, volume 4, pages 1942–1948, Australia, 1995.

[78] Saad M. Khan and Mubarak Shah. A Multi-view Approach to Tracking People in Dense Crowded Scenes using a Planar Homography Constraint. In *Proceedings of Workshop on Human Motion*, pages 133–146, Graz, Austria, 2006.

[79] Saad M. Khan and Mubarak Shah. Tracking Multiple Occluding People by Localizing on Multiple Scene Planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):505–519, March 2009.

[80] Zia Khan, Tucker Balch, and Frank Dellaert. MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, November 2005.

[81] Jaechul Kim and Kristen Grauman. Observe Locally, Infer Globally: A space-time mrf for detecting abnormal activities with incremental updates. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2921–2928, 2009.

[82] Mathias Kölsch and Matthew Turk. Hand tracking with flocks of features. In *Proceedings of Computer Vision and Pattern Recognition.*, volume 2, page 1187, 2005.

[83] Louis Kratz and Ko Nishino. Spatio-temporal Motion Pattern Modelling of Extremely Crowded Scenes. In *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis*, France, October 2008.

[84] Louis Kratz and Ko Nishino. Anomaly Detection in Extremely Crowded Scenes Using Spatio-temporal Motion Pattern Models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1446–1453, Florida, June 2009.

[85] Louis Kratz and Ko Nishino. Tracking with Local Spatio-Temporal Motion Patterns in Extremely Crowded Scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 693–700, San Francisco, 2010.

[86] Bogdan Kwolek. Object tracking via multi-region covariance and particle swarm optimization. In *Proceedings of Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 418 – 423, September 2009.

[87] Neil Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research*, 6(10-12):1783–1816, 2005.

[88] Neil Lawrence and Joaquin Qui nonero Candela. Local distance preservation in the gp-lvm through back constraints. *Proceedings of International conference on Machine learning*, pages 513–520, 2006.

[89] L. Lee, R. Romano, and G. Stein. Monitoring Activities from Multiple Video Streams: Establishing a Common Coordinate Frame. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):758 – 767, August 2000.

[90] Sung Chun Lee, Chang Huang, and Ram Nevatia. Definition, detection, and evaluation of meeting events in airport surveillance videos. In *Proceedings of TRECVID: TREC Video Retrieval Evaluation*, University of Southern California, 2008.

[91] J. Li, X. Lu, L. Ding, and H. Lu. Moving Target Tracking via Particle Filter Based on Color and Contour Features. In *Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on*, pages 1–4. IEEE, 2010.

[92] Jin Li, Hong Yu, and Hong Liang. Efficient mean shift tracking via particle swarm optimization for multi-articulated human body features. In *Proceedings of International Conference on Mechatronics and Automation*, pages 781 – 787, August 2008.

[93] Yuan Li, Chang Huang, and Ram Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2953–2960, June 2009.

[94] Weiyao Lin, Ming-Ting Sun, Radha Poovendran, and Zhengyou Zhang. Group event detection with a varying number of group members for video surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(8):1057–1067, August 2010.

[95] Matthias Luber, Johannes a Stork, Gian Diego Tipaldi, and Kai O Arras. People tracking with human motion predictions from social forces. In *2010 IEEE International Conference on Robotics and Automation*, pages 464–469. IEEE, may 2010.

[96] Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of Image Understanding Workshop*, pages 121–130, April 1981.

[97] How lung Eng, Kar-Ann Toh, Alvin H. Kam, Junxian Wang, and Wei-Yun Yau. An automatic drowning detection surveillance system for challenging outdoor pool environments. In *Proceedings of the Conference International Conference on Computer Vision*, volume 1, pages 532 – 539, 2003.

[98] D. Makris M. Lewandowski, J. Martinez-del-Rincon and J.-C. Nebel. Temporal extension of laplacian eigenmaps for unsupervised dimensionality reduction of time series. In *Proceedings of International Conference on Pattern Recognition*, pages 161–164. Springer, 2010.

[99] L. Ma, J. Liu, J. Wang, J. Cheng, and H. Lu. A improved silhouette tracking approach integrating particle filter with graph cuts. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1142–1145. IEEE, 2010.

[100] Yunqian Ma and Petter Cisar. Activity Representation in Crowd. In *Proceedings of the* 2008 *Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 107 – 116, Florida, December 2008. Springer.

[101] Jan R. Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley and Sons, second edition, 1999.

[102] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly Detection in Crowded Scenes. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1975–1981, San Francisco, June 2010.

[103] Iain Mattews, Takarhiro Ishikawa, and Simon Baker. The template update problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):810 – 815, June 2004.

[104] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal Crowd Behavior Detection using Social Force Model,. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 935–942, Florida, June 2009. IEEE.

[105] Marco A. MontesdeOca, Thomas Sttzle, Mauro Birattari, and Marco Dorigo. Frankensteins PSO: A composite particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5):1120–1132, October 2009.

[106] Carlos Serra-Toroand Raúl Montoliu, V. Javier Traver, Isabel M. Hurtado-Melgar, Manuela Nú nez Redó, and Pablo Cascales. Assessing water quality by video monitoring fish swimming behavior. In *Proceedings of International Conference on Pattern Recognition*, pages 428–431, 2010.

[107] Pradeep Natarajan and Ramakant Nevatia. Coupled Hidden Semi Markov Models for Activity Recognition. In *Proceedings of the IEEE Workshop on Motion and Video Computing*, pages 10–10, February 2007.

[108] Nandita M. Nayak, Ricky J.Sethi, Bi Song, and Amit K. Roy-Chowdhury. Motion pattern analysis for modeling and recognition of complex human activities. In *Visual Analysis of Humans: Looking at People*, pages 289–310. Springer, 2011.

[109] Bingbing Ni, Shuicheng Yan, and Ashraf Kassim. Recognising human group activities with localised causalities. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1470–1477, 2009.

[110] Z. Ni, S. Sunderrajan, A. Rahimi, and BS Manjunath. Distributed particle filter tracking with online multiple instance learning in a camera sensor network. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 37–40. IEEE, 2010.

[111] Huazhong Ninga, Wei Xub, Yun Chib, Yihong Gongb, and Thomas S. Huang. Incremental Spectral Clustering by Efficiently updating the Eigen-system. *Pattern Recognition*, 43(16):113–127, 2010.

[112] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An adaptive color-based particle filter. *Image and Vision Computing*, 21(1):99–110, 2003.

[113] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J. Little, and David G. Lowe. A Boosted Particle Filter: Multitarget Detection and Tracking. In *Proceedings of Eighth European Conference on Computer Vision*, pages 28–39. IEEE, 2004.

[114] Vasilis Papadourakis and Antonis Argyros. Multiple objects tracking in the presence of long-term occlusions. *Computer Vision and Image Understanding*, 114(7):835–846, July 2010.

[115] Daniel Parrott and Xiaodong Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation*, 10(4):440 – 458, August 2006.

[116] K. E. Parsopoulos and M. N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1:235–306, 2002.

[117] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti. Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1544–1554, November 2008.

[118] Robert Pless. Image spaces and video trajectories: Using isomap to explore video sequences. In *Proceedings of IEEE International Conference on Computer Vision*, volume 2, pages 1433–1440. IEEE, 2003.

[119] Riccardo Poli. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, 4:1–10, January 2008.

[120] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 728–735, June 2006.

[121] Z. Qi, R. Ting, F. Husheng, and Z. Jinlin. Particle Filter Object Tracking Based on Harris-SIFT Feature Matching. *Procedia Engineering*, 29:924–929, 2012.

[122] Nimmakayala Ramakoti, Ari Vinay, and Ravi Kumar Jatoth. Particle swarm optimization aided Kalman Filter for object tracking. In *Proceedings of International Conference on Advances in Computing, Control, and Telecommunication Technologies*, pages 531 – 533, December 2009.

[123] Vikas Reddy, Conrad Sanderson, and Brian Lovell. Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture. In *MLvMA Workshop, IEEE Conference on Computer Vision and Pattern Recognition*, pages 57–63, Colorado, June 2011. IEEE.

[124] Paolo Remagnino and G. A. Jones. Classifying surveillance events from attributes and behaviour. In *Proceedings of British Machine Vision Conference*, pages 685–694, 2001.

[125] Mikel Rodriguez, Saad Ali, and Takeo Kanade. Tracking in Unstructured Crowded Scenes. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1389–1396, Kyoto, 2009.

[126] SamT. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[127] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. Textures of optical flow for real-time anomaly detection in crowds. In *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, pages 230–235, September 2011.

[128] M. S. Ryoo and J. K. Aggarwal. Semantic Understanding of Continued and Recursive Human Activities. In *Proceedings of 18th International Conference on Pattern Recognition*, pages 379–382, Hong Kong, August 2006.

[129] M. S. Ryoo and J. K. Aggarwal. Recognition of High-level Group Activities Based on Activities of Individual Members. In *Proceedings of IEEE Workshop on Motion and Video Computing*, pages 1–8, Colorado, January 2008.

[130] M. S. Ryoo and J. K. Aggarwal. Semantic Representation and Recognition of Continued and Recursive Human Activities. *International Journal of Computer Vision*, 82(1):1–24, April 2009.

[131] Imran Saleemi, Lance Hartung, and Mubarak Shah. Scene Understanding by Statistical Modeling of Motion Patterns. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 2069 – 2076, San Francisco, June 2010.

[132] Lawrence K. Saul and SamT. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *The Journal of Machine Learning Research*, 4:119–155, 2003.

[133] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing Human Actions: A Local SVM Approach. In *Proceedings of 17th International Conference on Pattern Recognition*, volume 3, pages 32–36, Cambridge, 2004.

[134] Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

[135] S. Scott, T. Rawesak, and Irfan Essa. A System for Tracking and Recognizing Multiple People with Multiple Camera. Technical Report GIT-GVU-98-25, Georgia Institute of Technology, August 1998.

[136] Md. Haidar Sharif and Chabane Djeraba. PedVed: Pseudo Euclidian Distances for Video Events Detection. In *Proceedings of the Fifth International Symposium on Advances in Visual Computing*, pages 674–685, Berlin, Heidelberg, 2009. Springer-Verlag.

[137] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of Computer Vision and Pattern Recognition.*, volume 2, pages 593–600, June 1994.

[138] Yinghuan Shi, Yang Gao, and Ruili Wang. Real-time abnormal event detection in complicated scenes. In *International Conference Pattern Recognition*, pages 3653 – 3656, August 2010.

[139] C. Shu-hong and H. Chun-hai. Particle Filter Tracking Algorithm Based on Multi-Information Fusion. In *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, pages 1–4. IEEE, 2009.

[140] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and TRECVid. In *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321 – 330, New York, 2006. ACM Press.

[141] John R. Smith and Shih-Fu Chang. Visualseek: a fully automated content-based image query system. In *Proceedings of ACM international conference on Multimedia*, pages 87–98, 1997.

[142] Hiroki Sugano and Ryusuke Miyamoto. Parallel implementation of pedestrian tracking using multiple cues on GPGPU. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 900–906. IEEE, September 2009.

[143] Jolliffe Ian T. *Principal Component Analysis*, volume 489. Springer, second edition, 2002.

[144] Masaki Takahashi, Mahito Fujii, Masahiro Shibata, and Shin'ichi Satoh. Robust Recognition of Specific Human Behaviors in Crowded Surveillance Video Sequences. *EURASIP Journal on Advances in Signal Processing*, pages 1–14, March 2010.

[145] Sze Ling Tang, Zulaikha Kadim, Kim Meng Liang, and Mei Kuan Lim. Hybrid blob and particle filter tracking approach for robust object tracking. *Proceda Computer Science*, 1(1):2549–2557, 2010.

[146] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[147] Myo Thida, How-Lung Eng, D. N. Monekosso, and Paolo Remagnino. Learning video manifold for segmenting crowd events and abnormality detection. In *Proceedings of Tenth Asian Conference on Computer Vision*, pages 439–449. Springer, November 2010.

[148] Myo Thida, Paolo Remagnino, and How-Lung Eng. A particle swarm optimization approach for multi-objects tracking in crowded scene. In *proceedings of IEEE International Workshop on Visual Surveillance*, pages 1209 – 1215, September 2009.

[149] Ioannis Tziakos, Andrea Cavallaro, and Li-Qun Xu. Video event segmentation and visualisation in non-linear subspace. *Pattern Recognition Letter*, 30(2):123–131, January 2009.

[150] Ioannis Tziakos, Andrea Cavallaro, and Li-Qun Xu. Event monitoring via local motion abnormality detection in non-linear subspace. *Neurocomputing*, 73(10-12):1881–1891, June 2010.

[151] William H van der Schalie, Tommy R Shedd, Paul L Knechtges, and Mark W Widder. Using higher organisms in biological early warning systems for real-time toxicity detection. *Biosensors and Bioelectronics*, 16(7-8):457–465, 2001.

[152] Christian Vogler and Dimitris Metaxas. A Framework for Recognizing the Simultaneous Aspects of American Sign Language. *Computer Vision and Image Understanding*, 81(3):358 – 384, March 2001.

[153] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *Proceedings of Neural Information Processing Systems, NIPS*, pages 1441–1448, 2005.

[154] Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, pages 283–298, 2008.

[155] Xiaogang Wang, Kinh Tieu, and Eric Grimson. Learning Semantic Scene Models by Trajectory Analysis. In *Proceedings of European Conference on Computer Vision*, volume 3, pages 110–123, 2006.

[156] Peiliang Wu, Lingfu Kong, Fengda Zhao, and Xianshan Li. Particle filter tracking based on color and SIFT features. In *2008 International Conference on Audio, Language and Image Processing*, pages 932–937. IEEE, July 2008.

[157] Shandong Wu, Brian E. Moore, and Mubarak Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2054 – 2060, San Francisco, June 2010. IEEE.

[158] Dong Xu and Shih-Fu Chang. Video Event Recognition using Kernel Methods with Multilevel Temporal Alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1985–1997, May 2008.

[159] Kota Yamaguchi, Alexander C. Berg, Luis E. Ortiz, and Tamara L. Berg. Who are you with and where are you going? In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1345 – 1352, Colorado, June 2011. IEEE.

[160] Shuying Yang, Qin Ma, and Wenjuan Huang. Particle swarm optimized unscented particle filter for target tracking. In *Proceedings of International Congress on Image and Signal Processing*, pages 1 –5, October 2009.

[161] Yang Yang, Jingen Liu, and Mubarak Shah. Video Scene Understanding using Multi-scale Analysis. In *Proceedings of IEEE International Conference on Computer Vision*, pages 1669 – 1676, 2009.

[162] Gary G. Yen and Wen Fung Leong. Dynamic multiple swarms in multi-objective particle swarm optimization. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 39(4):890 – 911, July 2009.

[163] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM computing surveys*, 38(4):13, 2006.

[164] Lihi Zelnik-Manor and Michal Irani. Statistical Analysis of Dynamic Actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1530 – 1535, September 2006.

[165] Lihi Zelnik-Manor and Pietro Perona. Self-tuning Spectral Clustering. *In Advances in Neural Information Processing Systems*, 17:1601–1608, 2004.

[166] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja. Robust visual tracking via structured multi-task sparse learning. *International Journal of Computer Vision*, pages 1–17, 2012.

[167] Xiaoqin Zhang, Weiming Hu, Steve Maybank, Xi Li, and Mingliang Zhu. Sequential particle swarm optimization for visual tracking. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1 – 8, June 2008.

[168] Xiaoqin Zhang, Weiming Hu, Wie Qu, and Steve Maybank. Multiple object tracking via species-based particle swarm optimization. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(11):1590 – 1602, November 2010.

[169] Yuhua Zhang and Yan Meng. Adaptive object tracking using particle swarm optimization. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 43 – 48, June 2007.

[170] Xin Zheng and Xueyin Lin. Automatic determination of intrinsic cluster number family in spectral clustering using random walk on graph. In *International Conference on Image Processing*, volume 5, pages 3471 – 3474, 2004.

[171] Q. Zhong, Z. Qingqing, and G. Tengfei. Moving object tracking based on codebook and particle filter. *Procedia Engineering*, 29:174–178, 2012.

[172] Hao Zhou, Xuejie Zhang, Haiyan Li, and Jidong Li. Video object tracking based on swarm optimized particle filter. In *Proceedings of International Conference on Industrial Mechatronics and Automation*, volume 2, pages 702–706, May 2010.

[173] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. Object tracking using SIFT features and mean shift. *Computer Vision and Image Understanding*, 113(3):345–352, March 2009.

[174] Guangyu Zhu, Ming Yang, Kai Yu, Wei Xu, and Yihong Gong. Detecting Video Events Based on Action Recognition in Complex Scenes using Spatio-Temporal Descriptor. In *Proceedings of the seventeen ACM international conference on Multimedia*, pages 165–174, Beijing, October 2009.