Crowd Scene Analysis in Video Surveillance

Hanhe Lin

a thesis submitted for the degree of

Doctor of Philosophy

at the University of Otago, Dunedin,

New Zealand.

Abstract

There is an increasing interest in crowd scene analysis in video surveillance due to the ubiquitously deployed video surveillance systems in public places with high density of objects amid the increasing concern on public security and safety. A comprehensive crowd scene analysis approach is required to not only be able to recognize crowd events and detect abnormal events, but also update the innate learning model in an online, real-time fashion. To this end, a set of approaches for Crowd Event Recognition (CER) and Abnormal Event Detection (AED) are developed in this thesis.

To address the problem of curse of dimensionality, we propose a video manifold learning method for crowd event analysis. A novel feature descriptor is proposed to encode regional optical flow features of video frames, where adaptive quantization and binarization of the feature code are employed to improve the discriminant ability of crowd motion patterns. Using the feature code as input, a linear dimensionality reduction algorithm that preserves both the intrinsic spatial and temporal properties is proposed, where the generated low-dimensional video manifolds are conducted for CER and AED.

Moreover, we introduce a framework for AED by integrating a novel incremental and decremental One-Class Support Vector Machine (OCSVM) with a sliding buffer. It not only updates the model in an online fashion with low computational cost, but also adapts to concept drift by discarding obsolete patterns. Furthermore, the framework has been improved by introducing Multiple Incremental and Decremental Learning (MIDL), kernel fusion, and multiple target tracking, which leads to more accurate and faster AED.

In addition, we develop a framework for another video content analysis task, i.e., shot boundary detection. Specifically, instead of directly assessing the pairwise difference between consecutive frames over time, we propose to evaluate a divergence measure between two OCSVM classifiers trained on two successive frame sets, which is more robust to noise and gradual transitions such as fade-in and fade-out. To speed up the processing procedure, the two OCSVM classifiers are updated online by the MIDL proposed for AED.

Extensive experiments on five benchmark datasets validate the effectiveness and efficiency of our approaches in comparison with the state of the art.

Acknowledgements

I would like to express my sincere thanks and gratitude toward the following people who contributed their support and assistance to this thesis.

First and foremost, I am deeply grateful to my parents, especially my beloved father, who has passed away in the process of my research. As the first mentor in my life, his inspiration and encouragement stimulated me to pursue a PhD. I would not have finished this thesis without their financial and spiritual support. I also want to thank my wife, Shan Chen, who has taken on most of the housework so that I could concentrate on my research. My special thanks go to my three lovely kids, Baoer, newborn twins (Andi and Anna), I had a colourful PhD career because of them.

This thesis would not have been possible without continuous support and guidance from my conscientious supervisors. I owe a debt of gratitude to my primary supervisor, Associate Professor Dr. Jeremiah D. Deng, for his patient supervision, constant encouragement, and profound knowledge, which will benefit me for a lifetime. Furthermore, I wish to thank him for offering three research assistant jobs during my PhD study. I am greatly indebted to my co-supervisor Dr. Brendon J. Woodford. His academic advice, practical support, and constructive comments has been invaluable to my PhD study and research.

I especially want to acknowledge the past and present postgraduate members of the Pattern Recognition and Machine Learning (PRML) lab, Ahamad Shahi, Feng Zhou, Joyce Zhang, Juan Zhang, Michael Abrams, Munir Shah, Sean Lee, Sepideh Zareei, Xianbin Gu, and Yuwei Xu, for giving valuable feedbacks, engaging in discussion, as well as sharing their research ideas and experience in my research process.

I also want to thank the staff in the Department of Information Science, University of Otago, who have contributed to this thesis. Specifically, I am grateful to Associate Professor Peter Whigham for providing constructive feedbacks in my progress report meetings. Sincere thanks to the Technical Support Group (TSG) for offering IT support services whenever my computer encountered a problem. Deep gratitude towards administration staffs, Gail Mercer, Heather Cooper, Professor Michael Winikoff, and Stephen Hall-Jones, who have provided generous services and assistance for anything about my research and work.

Finally, besides the aforementioned people, I would like to thank all of those who has helped me complete my thesis no matter in whatever way.

Contents

List of Tables

List of Figures

Acronym

AED	Abnormal Event Detection
AMHOF	Adaptive Multi-scale Histogram of Optical Flow
AUC	Area Under the ROC Curve
BoW	Bag-of-Words
CER	Crowd Event Recognition
CRF	Conditional Random Field
EER	Equal Error Rate
EMD	Earth Movers Distance
GA	Global Anomaly
HI	Histogram Intersection
H-MDT	Hierarchical Mixture of Dynamic Texture
HMM	Hidden Markov Model
HOF	Histogram of Optical Flow
HOG	Histogram of Oriented Gradients
IFS	Immediate Future Set
IPS	Immediate Past Set
KF	Kernel Fusion
KKT	Karush-Kuhn-Tucker conditions
k-NN	k-Nearest Neighbours
LA	Local Anomaly
LDA	Latent Dirichlet Allocation
LE	Laplacian Eigenmaps
LPP	Locality Preserving Projections
MDT	Mixture of Dynamic Texture
MHT	Multiple Hypothesis Tracking
MHOF	Multi-scale Histogram of Optical Flow

MIDL	Multiple Incremental and Decremental Learning
MRF	Markov Random Field
MTT	Multiple Target Tracking
OCSVM	One-Class Support Vector Machine
PCA	Principal Component Analysis
PGM	Probabilistic Graphical Model
QBC	Quantized Binary Code
RBF	Radial Basis Function
RD	Rate of Detection
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
SBD	Shot Boundary Detection
SCL	Sparse Combination Learning
SIFT	Scale Invariant Feature Transform
SRC	Sparse Reconstruction Cost
STLPP	Spatial-Temporal Locality Preserving Projections
STPM	Spatial-Temporal Pyramid Matching
STVFF	Spatio-Temporal Viscous Fluid Field
SVM	Support Vector Machine
WHT	Walsh-Hadamard Transform

Mathematical Notation

Because there is a considerable overlap from Chapter 5 to Chapter 7, we use a consistent notation throughout these chapters.

$\mathcal{X}$	Training data
$I_{\mathcal{X}}$	Indexes of $\mathcal{X}$
$\mathbf{w}, \rho$	Optimal parameters for OCSVM
Φ	Feature map
α_i	Coefficient of data $\mathbf{x}_i$
$\mathcal{S}$	Margin support vector set
$\mathcal{E}$	Error support vector set
$\mathcal{O}$	The remaining set
$I_{\mathcal{S}}$	Indexes of $\mathcal{S}$
$I_{\mathcal{E}}$	Indexes of $\mathcal{E}$
$I_{\mathcal{O}}$	Indexes of $\mathcal{O}$
$\mathbf{x}_c$	Adding or removing data
$\Delta\alpha_c$	Coefficient change of $\mathbf{x}_c$
$k_{\mathcal{XY}}$	Kernel matrix, indexed by set $\mathcal{X}$ and $\mathcal{Y}$
Δg_i	Gradient change of $\mathbf{x}_i$
Q	Inverse matrix
$k(\cdot, \cdot)$	kernel function
$\mathcal{A}$	Addition set
$\mathcal{R}$	Removal set
$I_{\mathcal{A}}$	Indexes of $\mathcal{A}$
$I_{\mathcal{R}}$	Indexes of $\mathcal{R}$
$\Delta\mathcal{A}$	Coefficient change in set $\mathcal{A}$
$\Delta\mathcal{R}$	Coefficient change in set $\mathcal{R}$

B_1^t	IPS at time t
B_2^t	IFS at time t
$\mathbf{w}_1^t, \rho_1^t$	Optimal parameters for OCSVM learnt from B_1^t
$\mathbf{w}_2^t, \rho_2^t$	Optimal parameters for OCSVM learnt from B_2^t
$\boldsymbol{\alpha}_1^t$	Coefficient sets learnt from B_1^t
$\boldsymbol{\alpha}_2^t$	Coefficient sets learnt from B_2^t

Chapter 1

Introduction

1.1 Background

There is a rapid expansion of video surveillance systems due to the fact that recent advances in digital imaging, compression, transmission, and storage technologies have revolutionized the manner in which video information is analyzed, archived, and managed. Furthermore, the reduction in equipment prices resulting from the commoditization of video hardware further facilitates the proliferation of video surveillance systems.

Ubiquitous video surveillance systems bring about an exponential growth of raw, digital video data, exerting increasing pressure on conventional video monitoring and analysis processes, which are usually highly manpower intensive, inefficient and costly. Meanwhile, with the rapid advances of computer technologies, the ever-growing computational speed and memory capacity of computers has opened up new research directions that are intractable before due to various resource limits. As a result, there is an increasing interest in intelligent video surveillance research (Valera and Velastin, 2005; Kim *et al.*, 2010; Liu *et al.*, 2013; Wang, 2013).

Given a huge amount of videos collected by a set of surveillance cameras, an intelligent video surveillance system aims at detecting, tracking and recognizing objects of interest, and further analyzing and understanding the visual events of the scene. It has a wide range of applications both in public and private environments, e.g., access control, traffic enforcement (Tseng *et al.*, 2002), anomaly detection (Sodemann *et al.*, 2012), and health care (Rougier *et al.*, 2011).

Nowadays, video surveillance systems have been widely deployed in public places such as airports, train stations, shopping malls. Even though the high density of objects remains a challenge of tracking, recognizing and analysing the behaviours of

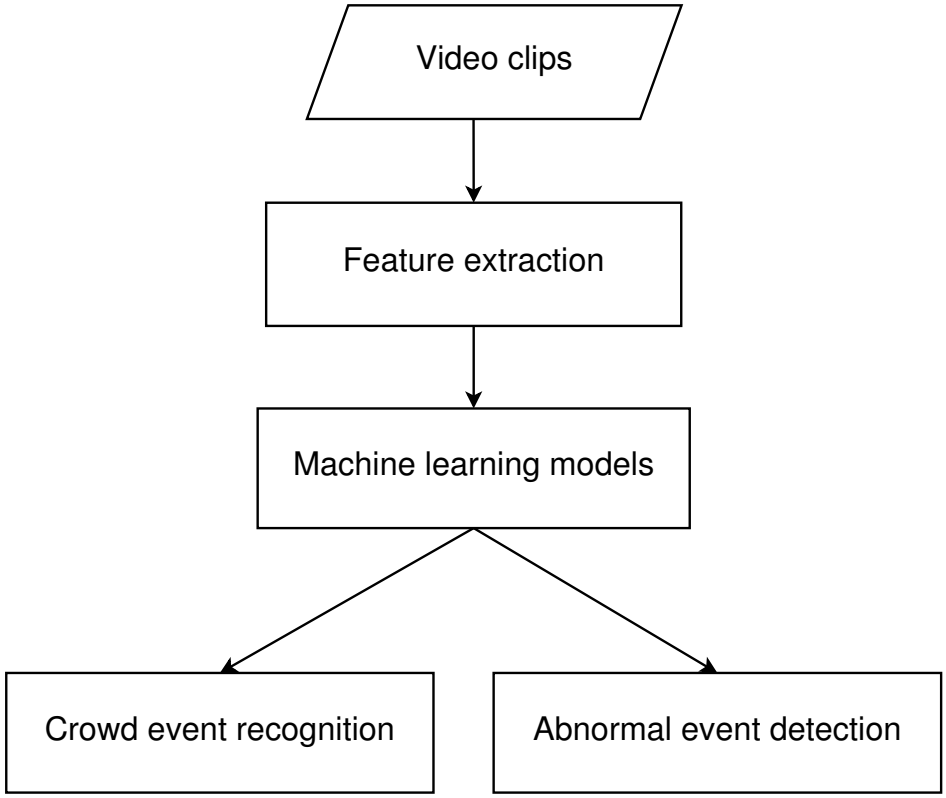


Figure 1.1: The flowchart of a conventional crowd scene analysis approach.

multiple objects in crowd, the research on crowd scene analysis (Li *et al.*, 2015) in video surveillance is attracting increasing attention along with the increasing concern on public security and safety. While crowd scene analysis may cover a wide range of research topics, this thesis mainly focuses on Crowd Event Recognition (CER) and Abnormal Event Detection[1] (AED).

A conventional crowd scene analysis approach is depicted in Figure 1.1. Given a set of video clips collected by a surveillance camera, the first step is to extract useful features from them. While a conventional intelligent video surveillance system would normally aim at tracking interest points on an individual object (Zhu *et al.*, 2006; Nguyen *et al.*, 2007), it cannot cope with frequently occurring conditions in crowd scene, e.g., severe occlusions, small object sizes, and strong similarity among objects. To overcome these difficulties, features that characterize crowd motions have been proposed, including optical flow (Andrade *et al.*, 2006; Adam *et al.*, 2008), spatial-temporal gradient (Kratz and Nishino, 2009), and volumetric shape matching (Ke *et al.*, 2007), to name a few. Some approaches may have a pre-processing step, e.g.,

[1]In the rest of the thesis, "abnormal event detection" is equivalent to "anomaly detection" in the context of crowd scene analysis. Likewise, "abnormal event" equals to "anomaly".

background subtraction, to filter noise and reduce computational cost simultaneously. With the extracted features as input, various machine learning models can be learnt in the training process, and applied to CER and/or AED in the testing process. Those models include social force model (Mehran *et al.*, 2009), sparse coding (Lee *et al.*, 2006), probabilistic graphical model (Koller and Friedman, 2009), etc. Besides, a post-processing step, e.g. conditional random field (Li *et al.*, 2014), may be involved to improve the detection result.

1.2 Challenges

In general, there are a few challenges for crowd scene analysis in video surveillance:

- As the high-dimensional video data suffer from the so-called problem entitled "curse of dimensionality" (Bellman and Corporation, 1957), various dimension reduction algorithms (Belkin and Niyogi, 2003) have been employed to generate low-dimensional manifolds for crowd scene analysis (Tziakos *et al.*, 2009, 2010; Thida *et al.*, 2010, 2012). However, the existing manifold learning research on crowd scene analysis is non-linear and operates in a batch mode, therefore, it is unclear how to embed new arriving video data and unsuitable for real-world applications.

- Most existing anomaly detection algorithms (Breunig *et al.*, 2000; Roth, 2006; Liu *et al.*, 2014) cannot be directly applied to AED as they operate by nature in batch mode, which requires the entire dataset to be made available in advance so that these algorithms can proceed. This not only makes the analysis of large-size video data rather time-consuming, it does not align well with the nature of video surveillance, where video data arrive sequentially and is required to be processed in real-time. Furthermore, changes in the monitoring environment also demand online adaptability of the detector. Essentially, an ideal video anomaly detection algorithm should be able to handle so-called "online learning" (Diethe and Girolami, 2013), namely, updating the innate models of the algorithms while streaming data arrive incrementally. In addition, it is required to be able to track slow changes of normal patterns, i.e., concept drift adaption (Gama *et al.*, 2014), as well as detect abrupt changes of abnormal patterns in real-time.

- Apart from the challenges of handling large amounts of streaming data, there are several challenges that have to be faced in AED exclusively. Firstly, the motion

feature of same object from different locations may vary due to perspective of a camera. As a result, an anomaly in distant view, for example, may be ignored due to small scale motion. Secondly, the diversity of crowd events increases the complexity of AED. For instance, a "normal" event appearing at a location where is has never occurred before is supposed to be detected as an anomaly in certain scenarios.

- An intelligent video surveillance system is supposed to provide real-time and automatic event and alarm notification. In other words, it should require less labor to supervise while monitoring and taking action from anywhere, anyhow and anywhere. To meet the requirement, a simple but effective feature extraction process combined with a fast online learning algorithm are required.

1.3 Research questions

Our main research questions hence emerge as follows:

1. Can we develop an efficient dimension reduction algorithm to consider both spatial and temporal similarities between frames when generating video manifolds?
2. Can we find a simple but efficient feature descriptor to capture the motion information in a crowded video surveillance stream?
3. Can an online learning algorithm be developed so that the learnt model can evolve on-the-fly from continuous data streams?
4. Combining the feature descriptor with the online learning algorithm, can we propose a unified framework for AED so that it meets the aforementioned challenges while providing real-time response?

1.4 Contributions

In this thesis, aiming at solving these research questions, there are four main contributions summarized within the respective chapters:

1. **Proposing a novel manifold learning algorithm for crowd scene analysis in video surveillance in Chapter 4.**

(a) A novel feature descriptor is proposed to encode regional optical flow features of video frames, where adaptive quantization and binarization of the feature code are employed to improve the discriminant ability of crowd motion patterns.

(b) We propose a novel linear dimension reduction algorithm to generate low-dimensional video manifolds that preserve both the intrinsic spatial and temporal properties.

(c) The foundation of this work has been published in Lin *et al.* (2013).

2. **Developing an online adaptive One-Class Support Vector Machines (OCSVM) algorithm for AED in Chapter 5.**

   (a) Integrating the incremental and decremental OCSVM with a sliding buffer offers an efficient and effective scheme, which not only updates the model in an online fashion with low computational cost, but also discards outdated patterns.

   (b) Providing a unified framework to detect both global and local anomalies.

   (c) The foundation of this work has been published in Lin *et al.* (2014, 2015).

3. **Improving efficiency and effectiveness of the proposed online adaptive OCSVM for AED in Chapter 6.**

   (a) To reduce the computational cost while enhancing the robustness of the anomaly detector, a background subtraction approach is adopted to detect regions of interest, in which two different features have been extracted, i.e., motion information and spatial location.

   (b) A novel feature descriptor is proposed to represent motion information, which adaptively decides its scale binning using a clustering process.

   (c) By combining motion information as well as spatial location information through kernel fusion, the proposed framework is able to detect motion anomalies as well as spatial location anomalies even under perspective distortions.

   (d) The processing speed has also been increased by replacing the single instance incremental and decremental algorithm with Multiple Incremental and Decremental Learning (MIDL) algorithm.

(e) The final detection result is further improved by a linear Multiple Target Tracking (MTT) approach.

4. **Proposing a computational framework for Shot Boundary Detection (SBD) on the basis of the MIDL algorithm in Chapter 7.**

   (a) Instead of comparing the difference between pair-wise consecutive frames at a specific time, we measure the divergence between two OCSVM classifiers, which are learnt from two contextual sets, i.e., immediate past set and immediate future set. The approach, which inherits the advantages of OCSVM, is robust to noises such as abrupt illumination changes and large object or camera movements, and capable of detecting gradual transitions as well.

   (b) The two OCSVM classifiers are updated in an online fashion by our proposed MIDL algorithm to speed up the processing procedure.

   (c) The foundation of this work has been published in Lin *et al.* (2016).

1.5 List of publications

Some parts of the thesis have been published in the following conferences/workshop:

- H. Lin, J. D. Deng, B. J. Woodford, and A. Shahi **Online Weighted Clustering for Real-time Abnormal Event Detection in Video Surveillance**. Accepted by ACM Multimedia 2016.

- H. Lin, J. D. Deng, and B. J. Woodford **Shot Boundary Detection Using Multi-instance Incremental and Decremental One-Class Support Vector Machine**. In Advances in Knowledge Discovery and Data Mining (PAKDD) 2016 Apr 19 (pp. 165-176). Springer International Publishing.

- H. Lin, J. D. Deng, and B. J. Woodford **Anomaly detection in crowd scenes via online adaptive one-class support vector machines**. In Image Processing (ICIP), 2015 IEEE International Conference on 2015 Sep 27 (pp. 2434-2438). IEEE.

- H. Lin, J. D. Deng, and B. J. Woodford **Spatial-Temporal Pyramid Matching for Crowd Scene Analysis**. In Proceedings of the MLSDA 2014 2nd

Workshop on Machine Learning for Sensory Data Analysis 2014 Dec 2 (p. 12). ACM.

- H. Lin, J. D. Deng, and B. J. Woodford **Event detection using quantized binary code and spatial-temporal locality preserving projections**. In AI 2013: Advances in Artificial Intelligence 2013 Dec 1 (pp. 123-134). Springer International Publishing. (Best paper and best student paper nomination)

- H. Lin, J. D. Deng, and B. J. Woodford **Video manifold modelling: finding the right parameter settings for anomaly detection**. In Proceedings of the 27th Conference on Image and Vision Computing New Zealand 2012 Nov 26 (pp. 168-173). ACM.

1.6 Organization of the thesis

The rest of the thesis is organized as follows:

Chapter 2 Literature review

Some fundamentals techniques about spatial-temporal feature detectors and descriptors are reviewed. Next, we review some classical machine learning models that have been employed for crowd scene analysis. Finally, we provide a comprehensive literature review on state-of-the-art approaches for CER and AED.

Chapter 3 Research methodology

We introduce a new generic approach adopted in this thesis. The performance evaluation methodologies correspond to different tasks are discussed here. Five different datasets with different tasks, i.e., CER, AED and SBD, that are conducted through the thesis are also introduced.

Chapter 4 Crowd Scene Analysis using adaptive quantization and manifold learning

A novel manifold learning algorithm for crowd scene analysis in video surveillance is proposed. We propose a novel feature descriptor to improve the discriminant ability of crowd motion patterns. A linear dimension reduction algorithm that preserves both intrinsic spatial and temporal properties of those feature descriptors is developed.

Chapter 5 Online Adaptive OCSVM for AED

In this chapter, a novel incremental and decremental OCSVM algorithm is proposed. Anomalies are divided into two categories, global anomalies and local anomalies. A unified framework that integrates the incremental and decremental OCSVM with a

sliding buffer to detect both anomalies is introduced as well.

Chapter 6 Multiple Incremental and Decremental Learning OCSVM for AED

An improved framework for AED is proposed in this chapter. We adopt a background subtraction approach to filter noise and extract two different type of features, namely motion feature and spatial location information, where kernel fusion is employed to improve detection results. Furthermore, the MIDL algorithm that replaces the single incremental and decremental algorithm is proposed. Finally, a MTT approach is introduced to improve the final detection results.

Chapter 7 MIDL OCSVM for Shot Boundary Detection

In this chapter, we extend the MIDL to another video content analysis application - SBD. A literature review on SBD is conducted. Later on, we define a function to measure the divergence between two OCSVM classifiers, which correspond to two consecutive but non-overlapping sets of frames. SBD is conducted on the divergence output.

Chapter 8 Conclusion and future work

We draw a conclusion of the research work carried out in this thesis, and some possible research directions for future work that extends this thesis are also discussed.

Chapter 2

Literature Review

2.1 Overview

Because a promising computer vision analysis model heavily relies on feature extraction and how to make use of the features for object detection, in this chapter we first discuss some fundamental techniques regarding video feature extraction, which is separated into two parts: feature detectors and feature descriptors. The feature detectors that are discussed here include Harris3D (Laptev, 2005), Cuboid (Dollár *et al.*, 2005), and dense sampling; and feature descriptors such as SIFT3D (Scovanner *et al.*, 2007) are also highlighted. Later on, we review a set of machine learning models that have been adopted for crowd scene analysis. Finally, a broad spectrum of approaches for crowd scene analysis proposed in the literature are reviewed. On the basis of the research objectives, these approaches are categorized into two types: Crowd Event Recognition (CER) and Abnormal Event Detection (AED).

2.2 Feature extraction

In general, a very short video can even generate a huge amount of feature descriptors (Boiman and Irani, 2007). How to detect those effective features and represent them is the crucial prerequisite for further video content analysis. In this section, we will introduce some well-known spatial-temporal interest point detectors briefly, and some representative spatial-temporal feature descriptors are discussed as well.

2.2.1 Feature detectors

The earliest exploration of feature detection occurred in image processing domain, e.g., Harris corner detector (Harris and Stephens, 1988), where points with a significant local variation and relative stability with respect to perspective transformation in image intensities, have been detected. Such points are referred to as "interest points". A decade later, due to the enhancement of computational power, these points have been extensively investigated (Schmid *et al.*, 2000; Lowe, 2004; Mikolajczyk and Schmid, 2004, 2005; Bay *et al.*, 2006; Nistér and Stewénius, 2008; Rublee *et al.*, 2011; Leutenegger *et al.*, 2011; Alahi *et al.*, 2012; Yu *et al.*, 2013). Among them, Scale Invariant Feature Transform (SIFT) (Lowe, 2004) is a significant milestone, where both a novel scale invariant feature detector and the corresponding feature descriptor are proposed.

Following a similar idea, these local interest points have been extended to the spatial-temporal domain, namely video (Laptev, 2005; Dollár *et al.*, 2005; Ke *et al.*, 2005; Willems *et al.*, 2008; Bregonzio *et al.*, 2009).

Harris3D

Harris3D (Laptev, 2005), which extends the 2D scale-invariant Harris-Laplace corner detector (Mikolajczyk and Schmid, 2004), has been extensively applied to action recognition (Laptev and Pérez, 2007; Laptev *et al.*, 2008). The basic concept is introducing a spatial-temporal second-moment matrix :

$$\mu = g(\cdot; \sigma_i^2, \tau_i^2) * \begin{bmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t L_t \end{bmatrix}, \tag{2.1}$$

where $g(\cdot; \sigma_i^2, \tau_i^2)$ is the Gaussian weighting function, given by:

$$g(x, y, t; \sigma_l^2, \tau_l^2) = \frac{1}{\sqrt{(2\pi)^3 \sigma_l^4 \tau_l^2}} \exp\left(- \frac{(x^2 + y^2)}{2\sigma_l^2} - \frac{t^2}{2\tau_l^2} \right). \tag{2.2}$$

L_x, L_y, and L_t are the first order spatial and temporal derivatives, defined as:

$$\begin{aligned} L_x(\cdot; \sigma_l^2, \tau_l^2) &= \partial_x (g * f), \\ L_y(\cdot; \sigma_l^2, \tau_l^2) &= \partial_y (g * f), \\ L_t(\cdot; \sigma_l^2, \tau_l^2) &= \partial_t (g * f), \end{aligned} \tag{2.3}$$

where σ_l and τ_l are the spatial and temporal scale, with $\sigma_i^2 = s\sigma_l^2$ and $\tau_i^2 = s\tau_l^2$.

As a result, the interest points in a video sequence f correspond to the regions having significant eigenvalues λ_1, λ_2, and λ_3 of μ, i.e.,

$$H = \det(\mu) - k \ \mathrm{trace}^3(\mu), \tag{2.4}$$

with the requirement of $H \geq 0$ and a typical value of $k = 0.005$. Eventually, the spatial-temporal interests points are given by the local maxima of H.

Cuboid

Let's assume a stationary camera or a process can remove the effect of camera motion, Cuboid (Dollár *et al.*, 2005) defines the response function as follows:

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2, \tag{2.5}$$

where $g(x, y; \sigma)$ is the 2D Gaussian smoothing kernel, applied on the spatial dimensions only. h_{ev} and h_{od} are a quadrature pair of 1D Gabor filters applied along temporal dimension, defined as:

$$\begin{aligned} h_{ev}(t; \tau, \omega) &= -\cos(2\pi t\omega)e^{-t^2/\tau^2}, \\ h_{od}(t; \tau, \omega) &= -\sin(2\pi t\omega)e^{-t^2/\tau^2}, \end{aligned} \tag{2.6}$$

where the default setting of ω is equal to $4/\tau$, and σ and τ correspond to the spatial and temporal scales of the detector. The spatial-temporal interest points correspond to the local maxima of R.

Generally speaking, Cuboid is able to extract more dense interest points than Harris3D, and has been employed for AED (Zhao *et al.*, 2011; Dutta and Banerjee, 2015).

Dense sampling

Although Cuboid is able to extract more interest points than Harris3D, the overall extracted points are still sparse. This is unsuitable for our work as in crowd scenes, we would like to detect as many features as possible to cover the whole scene so all the whole motion features can be captured and located accurately. Therefore, dense sampling, which extracts features at regular positions and scales, is the most common scheme to extract features in crowd scenes.

The same as these keypoint detectors, dense sampling is employed in image domain initially, such as object detection and categorization (Viola and Jones, 2001; Lazebnik *et al.*, 2006; Bosch *et al.*, 2007), face detection (Viola and Jones, 2004), human detection

(Dalal and Triggs, 2005; Viola *et al.*, 2005; Dalal *et al.*, 2006; Felzenszwalb *et al.*, 2010). Later on, this idea was extended to video domain, such as action recognition (Wang *et al.*, 2009; Tian *et al.*, 2013), AED (Boiman and Irani, 2007; Roshtkhari and Levine, 2013b). Empirical evidence in both object categorization (Bosch *et al.*, 2007) as well as action recognition (Wang *et al.*, 2009) indicates that dense sampling usually outperforms keypoint detectors even though some noisy features exist.

2.2.2 Feature descriptors

In recent years, a number of feature descriptors have been proposed in image processing, including color and texture (Manjunath *et al.*, 2001; Van De Sande *et al.*, 2010), wavelet (Torrence and Compo, 1998), Histogram of Oriented Gradient (HOG) (Dalal and Triggs, 2005), Local self-similarity (Shechtman and Irani, 2007), to name a few.

To extend it for video processing, scholars and researchers proposed a series of spatial-temporal feature descriptors, e.g., Grundmann *et al.* (2008). Given a sample point (x, y, t), a cube which contains the spatial-temporal windowed pixel values around that point is extracted, where various features, e.g., gradient, optical flow, have been investigated to describe the shape and/or motion information.

Gradient based descriptor

SIFT3D (Scovanner *et al.*, 2007) is an extension of the popular SIFT to video processing. It computes spatial-temporal gradients L_x, L_y, L_t of each pixel in a spatial-temporal cube. Correspondingly, the gradient magnitude and orientations in 3D are estimated as:

$$\begin{aligned} m_{3D}(x, y, t) &= \sqrt{L_x^2 + L_y^2 + L_t^2}, \\ \theta(x, y, t) &= \tan^{-1}(L_y / L_x), \\ \phi(x, y, t) &= \tan^{-1}(\frac{L_t}{\sqrt{L_x^2 + L_y^2}}), \end{aligned} \tag{2.7}$$

where each pixel contains a single magnitude value m_{3D} and two orientation angles θ and ϕ, which is different from SIFT, containing only one orientation value θ. The spatial-temporal cube is divided into $n \times n \times n$ sub-volumes, in each sub-volumes a weighted 8×4 sub-histogram is constructed by dividing θ and ϕ into equally sized bins. The final descriptor is the concatenation and normalization of the sub-histograms.

Analogous to SIFT3D, Klaser *et al.* (2008) proposed the HOG3D on the basis of HOG (Dalal and Triggs, 2005) as well. It first computes each mean gradient us-

ing integral videos, where the gradient orientation is quantized by means of regular polyhedrons. The histograms of oriented gradients, which are computed in each cell separately, are concatenated and normalized as a final descriptor.

Kratz and Nishino (2009) proposed a 3D Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ to model the distribution of spatial-temporal gradients in a cube:

$$\begin{aligned} \mu &= \frac{1}{N}\sum_{i}^{N} \nabla I_i, \\ \Sigma &= \frac{1}{N}\sum_{i}^{N} (\nabla I_i - \mu)(\nabla I_i - \mu)^T, \end{aligned} \tag{2.8}$$

where $\nabla I_i = [L_x, L_y, L_t]$ is the spatial-temporal gradients of pixel i. Eventually, a cube at spatial location s and temporal location t is defined as $O_t^s = \{\mu_t^s, \Sigma_t^s\}$.

Optical-flow based descriptor

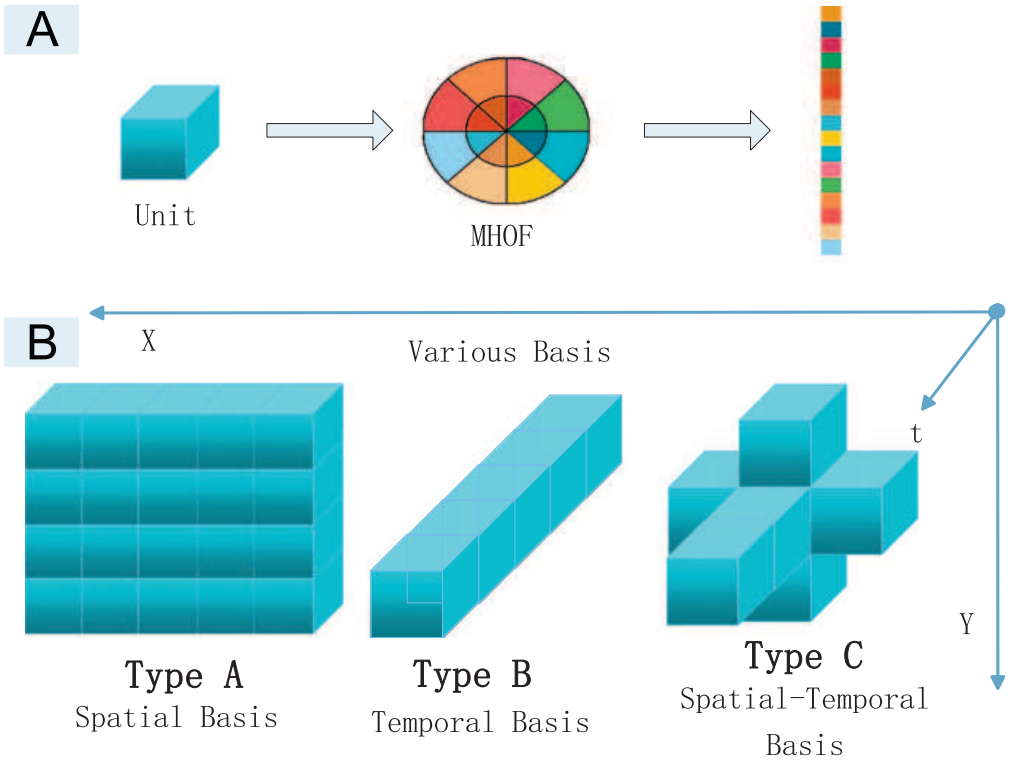


Figure 2.1: (A) The process of MHOF extraction. (B) Different event types based on anomaly scenarios. This figure is reproduced from Cong *et al.* (2013).

Optical flow (Lucas *et al.*, 1981; Horn, 1986), defined as the observed apparent motion of objects caused by the relation motion between an observer and the objects,

has been extensively used in computer vision tasks, such as background subtraction (Chen *et al.*, 2014), motion estimation (Liu *et al.*, 2008; Wedel *et al.*, 2011), etc.

To characterize the motion pattern, Kim and Grauman (2009) first computed a 9-dimensional optical flow vector, i.e., 8 orientations bin and 1 speed bin, for each pixel. Then a cube is divided into $M \times N$ cells, where each cell is represented as a 9 dimension vector by summing the flow from all pixels within it. Finally, a $9MN$ dimensional descriptor is constructed by concatenating the flow from all cells.

Cong *et al.* (2013) propose a Multi-scale Histogram of Optical Flow (MHOF) descriptor, as depicted in Figure 2.1. The spatial location of an image is partitioned into a few basic units, where a 16-bin MHOF is extracted in each unit. Specifically, all the flows in each unit are quantized into 16 bins, where the first 8 bins denote eight directions with motion magnitude less than a threshold, and the second 8 bins denote eight directions with motion magnitude equal and greater than a threshold. Three different event types are defined to model motion patterns with spatial and/or temporal structure, i.e., spatial basis, temporal basis, and spatial-temporal basis.

Dynamic texture

Dynamic texture (Doretto *et al.*, 2003) represents a video clip of moving scenes as certain stationarity properties in time. More specifically, a dynamic texture is a generative probabilistic model that is represented as a linear dynamic system:

$$\begin{cases} y_t = Ax_t + w_t \\ x_{t+1} = Bx_t + v_t \end{cases} \tag{2.9}$$

where $y_t \in \mathbb{R}^m$ is the vector of pixels in video frame at time t, $x_t \in \mathbb{R}^n$ is a hidden state variable ($n < m$) that represents the dynamic of y_t, with the independent and identically distributed noise w_t and v_t. The observation matrix $A \in \mathbb{R}^{m \times n}$ projects the hidden state space to the observation, and the transition matrix $B \in \mathbb{R}^{n \times n}$ controls the evolution of the hidden state over time. The optimal parameters can be estimated by a second-order stationary process (Ljung, 1998).

2.2.3 Discussion

On the one hand, although the spatial-temporal key point detectors such as Harris3D have achieved promising performance in human action recognition, the extracted points are still sparse. Under the scenario of action recognition, this may work well because

there is usually one or two humans in a video clip. However, these detectors are not suitable for our work because they may result in some missing detection as there are so many moving objects are in a crowd scene. On the other hand, the gradient based descriptor as well as the dynamic texture have shown the discriminative power but are computation-intensive. As a result, the feature extraction process employed in our work is dense sampling together with an optical-flow based descriptor.

2.3 Machine learning models

After some key spatial-temporal feature detectors and descriptors are introduced, the machine learning models that are commonly utilized for crowd scene analysis will be reviewed in this section.

2.3.1 k-Nearest Neighbours

k-Nearest Neighbours (k-NN) (Larose, 2005) is a non-parametric and data-driven algorithm used for multi-class classification. Given a testing data, k-NN seeks to find out its k nearest neighbours from the entire labelled training set with a pre-defined distance metric. The testing data is classified by a majority vote of its k nearest neighbours. In other words, the testing data is assigned to the dominant class among its k nearest neighbours. The special case is $k = 1$, where the testing data is simply assigned to the class of its nearest neighbour.

The advantages of k-NN are twofold. Firstly, it is easy to implement as in the training phase the only step is storing the training set and their class labels. Secondly, it is flexible to handle diverse data by using specific distance metrics. However, it is computationally intensive when training set is very large because it has to identify the k nearest neighbour for each testing data.

2.3.2 Sparse coding

Recently, sparse coding (Lee *et al.*, 2006; Mairal *et al.*, 2010) has been applied to a considerable amount of multi-disciplinary research (Mairal *et al.*, 2014). Generally speaking, given a set of n training data $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, sparse coding aims to represent $\mathbf{x}_i \in \mathbb{R}^m$ approximately as the product of a sparse weight vector $\boldsymbol{\alpha}_i$ and a learnt dictionary $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k]$, i.e., $\mathbf{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i$.

To acquire the optimal $\mathbf{A} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n]$ and $\mathbf{D}$, one may solve the following optimization problem:

$$\begin{aligned} &\min_{\mathbf{A},\mathbf{D}} : \sum_{i=1}^{n} \frac{1}{2} ||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda\psi(\boldsymbol{\alpha}_i) \\ &s.t. : \mathbf{d}_j^T \mathbf{d}_j \leq 1, \forall j = 1, \ldots, k \end{aligned} \tag{2.10}$$

where $\psi(\cdot)$ is a sparsity function, defined by a user.

There are two steps to optimize $\mathbf{D}$ and $\mathbf{A}$ alternatively. The first step is learning dictionary $\mathbf{D}$ while fixing $\mathbf{A}$. The second step is reconstructing $\mathbf{x}_i$ given a learnt dictionary $\mathbf{D}$. In other words, learning $\mathbf{A}$ while fixing $\mathbf{D}$. The optimization of Eq. (2.10) is equivalent to iteratively undertaking the two steps.

2.3.3 Probabilistic Graphical Models

Probabilistic Graphical Models (PGMs) (Koller and Friedman, 2009) compactly encode a complex distribution over a high-dimensional space by a graph model, which is composed of nodes and edges between the nodes. Each node corresponds to a variable, and a edge between two nodes means a direct probabilistic interaction between them. The PGMs are generally divided into two families: directed graph models and undirected graph models.

One of the most important directed graph models is Bayesian network, in which a directed edge from node X to node Y is represented as the conditional probability $P(Y|X)$. One notable example of the Bayesian network is Latent Dirichlet Allocation (LDA) (Blei *et al.*, 2003). Another directed graph model is Hidden Markov Model (HMM) (Baum and Petrie, 1966), which has been commonly used in temporal pattern recognition (Rabiner, 1989).

Because the influence between two nodes are symmetric, the undirected graph model, also known as a Markov Random Field (MRF) (Geman and Graffigne, 1986; Chen and Tang, 2007), uses an undirected edge to model the interaction between them. One notable variant of MRF is Conditional Random Field (CRF) (Lafferty *et al.*, 2001; He *et al.*, 2004).

PGMs are the mainstream of machine learning, and their applications include speech recognition, computer vision, etc. However, most of existing PGMs are required to label the dataset manually before training, which is highly manpower intensive. Moreover, it remains a challenge to learn a PGM online even though there are some attempts (Hoffman *et al.*, 2010).

2.3.4 Manifold learning

Because of the curse of dimensionality (Bellman and Corporation, 1957), a broad spectrum of dimension reduction algorithms (Hotelling, 1933; Fisher, 1936; Schölkopf *et al.*, 1998; Tenenbaum *et al.*, 2000; Roweis and Saul, 2000; Belkin and Niyogi, 2003; Niyogi, 2004) have been employed to generate low-dimensional manifolds for visualization and preprocessing purposes.

The earliest exploration in manifold learning linearly projects high-dimensional data into a low-dimensional subspace. Principal Component Analysis (PCA) (Hotelling, 1933) is the mainstay of the linear dimension reduction algorithms. Given a set of data with m dimensions, PCA aims to find the most l ($l \ll m$) meaningful bases, called principal components, to represent the data set. By solving an eigenvalue problem, namely, computing eigenvector and eigenvalues of the sample covariance matrix, PCA finds a linear low-dimensional embedding of data that best preserves their variance. Another example is the Fisher's Linear Discriminant (FLD) (Fisher, 1936). It introduces two matrices, a within-class scatter matrix S_w and a between-class scatter matrix S_b, to preserve as much as of the class discrimination information as possible. This is implemented by maximizing the between-class measure and minimizing within-class measure, i.e. maximizing the ratio of $\det|S_b|/\det|S_w|$.

Methods that generate non-linear manifolds have also been considered. Tenenbaum *et al.* (2000) propose Isometric feature mapping (Isomap). It seeks to preserve the intrinsic geometry of the data by constructing neighbourhood graph and computing geodesic distance between all pairs of data, thus has the capability to discover the non-linear degrees of freedom that underlie complex natural observations.

Belkin and Niyogi (2003) propose Laplacian Eigenmaps (LE) to preserve local neighbourhood information. Given a set of $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ in $\mathbb{R}^m$, the LE tries to find a set of points $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ in $\mathbb{R}^l$, where $l \ll m$ such $\mathbf{y}_i$ "represents" $\mathbf{x}_i$. The algorithmic procedure is composed of three steps. Firstly, an adjacency graph is constructed using either ϵ-neighbourhoods or k-NN. Specifically, let G be a graph of n nodes, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other, we put an edge between node i and j. Secondly, a weight matrix W is constructed on the basis of the adjacency graph. If there is an edge between node i and j, we put a weight on W_{ij} either Heat kernel $W_{ij} = e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||}{\sigma}}$, where σ determines the width of the kernel, or simple-minded, i.e., $W_{ij} = 1$ if node i and j is connected, otherwise is 0. Finally, eigenvectors and eigenvalues are computed

for the following generalized eigenvector problem (Golub and van Loan, 1996):

$$L\mathbf{v} = \lambda D\mathbf{v}, \tag{2.11}$$

where D is a diagonal matrix whose entries are column (or row) sums of W, i.e., $d_{ii} = \sum_j w_{ij}$; $L = D - W$ is the Laplacian matrix. Let $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{n-1}$ are the solution of Eq. (2.11) with ascending order of their corresponding eigenvalues $0 = \lambda_0 < \cdots < \lambda_{n-1}$, the l low-dimensional data representation $\mathbf{y}_i$ equals to $(\mathbf{v}_1(i), \ldots, \mathbf{v}_l(i))$.

While LE explores the non-linear data representation, Locality Preserving Projections (LPP) (Niyogi, 2004) extends the similar idea to a linear embedding subspace. As a linear approximation of LE, LPP aims to find the transformation matrix V such that $\mathbf{y}_i = V\mathbf{x}_i$. The major difference of LPP lies computing a variant generalized eigenvector problem:

$$XLX^T\mathbf{v} = \lambda XDX^T\mathbf{v}, \tag{2.12}$$

where D and L have the same definitions in Eq. (2.11), and X denotes the data matrix whose i-th column corresponds to $\mathbf{x}_i$. As a result, the transformation matrix V is represented as $(\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{l-1})$, where $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{l-1}$ are the solution of Eq. (2.12) with ascending order of their corresponding eigenvalues $\lambda_0 < \cdots < \lambda_{l-1}$.

2.3.5 Support Vector Machines

Support Vector Machines (SVM) (Cortes and Vapnik, 1995; Schölkopf *et al.*, 2000; Chang and Lin, 2011) is a supervised learning model, which has been widely employed in a number of real-world applications, e.g., text categorization (Joachims, 1998), image categorization (Csurka *et al.*, 2004).

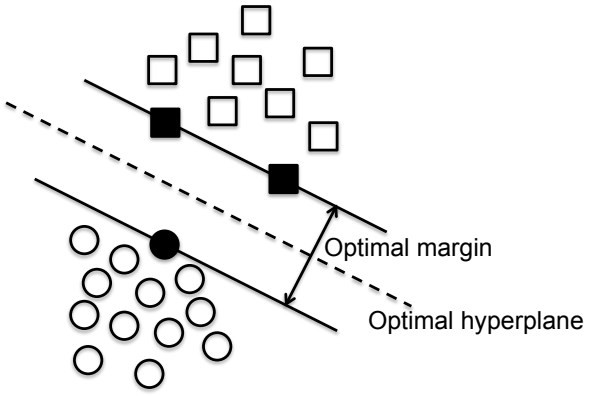


Figure 2.2: A simple example of SVM, where the support vectors, located on the margin, define the optimal hyperplane.

SVM originates from the two-class classification problem. Given a set of labelled two class training data $\{(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)\}, \forall y_i \in \{-1, 1\}$, SVM seeks to find an optimal hyperplane $\mathbf{w} \cdot \mathbf{x}_i + b$ which maximizes the margin between the two classes - Figure 2.2. To acquire the optimal parameters $\mathbf{w}$ and b, one can solve the following quadratic programming problem:

$$\begin{aligned} &\min_{\mathbf{w},\xi,b} : \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i \\ &s.t. : y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0, \end{aligned} \tag{2.13}$$

where ξ_i are slack variables, and C is a regularization parameter, controlling the trade-off between structure and empirical risk.

Using Lagrange multipliers, we can solve the following dual problem instead:

$$\begin{aligned} &\min_{\alpha} : \frac{1}{2}\sum_{ij} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_i \alpha_i \\ &s.t. : 0 \leq \alpha_i \leq C,\ \sum_i \alpha_i y_i = 0. \end{aligned} \tag{2.14}$$

One significant property of SVM is the kernel method (Scholkopf and Smola, 2001). That is, replacing $\mathbf{x}_i^T \mathbf{x}_j$ in Eq. 2.14 with a kernel function $k(\mathbf{x}, \mathbf{x}_j)$, classification can be conducted in a non-linear feature space.

Following the similar idea, SVM has also been extended to one-class problem, namely OCSVM (Schölkopf *et al.*, 2001; Tax and Duin, 2004), and multi-class problem (Hsu and Lin, 2002). Because one of our work is to extend OCSVM for online learning, we will introduce the concept of OCSVM and its batch mode solution in Chapter 5, and its online learning solution and improved variant will be discussed in Chapter 5 and Chapter 6 respectively.

2.4 Crowd scene analysis

Existing work of crowd scenes analysis can be categorized into two classes: CER (Garate *et al.*, 2009; Chan *et al.*, 2009; Thida *et al.*, 2012; Su *et al.*, 2012, 2013) and AED (Boiman and Irani, 2007; Adam *et al.*, 2008; Mehran *et al.*, 2009; Benezeth *et al.*, 2009; Wu *et al.*, 2010; Mahadevan *et al.*, 2010; Cui *et al.*, 2011; Cong *et al.*, 2011; Shi *et al.*, 2010; Zhao *et al.*, 2011; Saligrama and Chen, 2012; Roshtkhari and Levine, 2013b,a; Dutta and Banerjee, 2015; Yuan *et al.*, 2015; Antić and Ommer, 2015).

The CER is a multi-class classification problem. One possible solution is to train all the crowd event class in a single model, e.g., LDA. Another possible solution is to

train a number of two-class classifiers, where each classifier models a crowd event type separately, the classification is based on a one-against-all scheme (Csurka *et al.*, 2004).

The AED is somewhat different from CER, where only normal crowd events are trained for detection. In other words, it is an one-class classification problem (Khan and Madden, 2009).

2.4.1 Crowd event recognition

Garate *et al.* (2009) propose a novel tracker, based on the HOG descriptor, to track the feature points that are detected in video clips. The crowd events are recognized by a pre-defined model considering both speed and direction of these tracked feature points. Although the trajectory-based approach is simple, it is however impractical for crowd scenes analysis under extremely crowded environments with occlusions.

Chan *et al.* (2009) adopt a holistic representation to capture the variability of the motion field. A video sequence is divided into a set of video clips, where each video clip is modelled by a dynamic texture. The classification is performed by either the nearest neighbour approach measured by the Kullback-Leibler (KL) divergence (Chan and Vasconcelos, 2005b) or the SVM with the KL kernel (Moreno *et al.*, 2003).

Su *et al.* (2013) model crowd motion patterns as a Spatio-Temporal Viscous Fluid Field (STVFF), in which the interaction forces between pedestrians are estimated and designed by a Bag-of-Words (BoW) (Sivic and Zisserman, 2009) model. Crowd events are recognized by utilizing a LDA model.

To address the problem of curse of dimensionality, Thida *et al.* (2010, 2012) project the video frames into an embedded low-dimensional subspace. Specifically, a weighted adjacency matrix is constructed, in which an element measures spatial similarity using histogram of optical flow and temporal similarity between two frames. By solving an eigenvalue problem with the notion of Laplacian, the first k eigenvectors with the k smallest non-zero eigenvalues are their corresponding low-dimensional representation. Crowd scene analysis is carried out on the low-dimensional trajectories with machine learning models.

2.4.2 Abnormal event detection

AED in video surveillance (Sodemann *et al.*, 2012) is attracting more attention along with the growing importance of public security and safety. We divide abnormal event into global abnormal event and local abnormal event on the basis of scale. Global

abnormal events are those abnormal events that occur in the whole frame, and local abnormal events are those that occur in the local part of a frame.

In Wu *et al.* (2010), the largest Lyapunov exponents and correlation dimensions of the particle trajectories are extracted for chaotic dynamics analysis and anomaly detection in crowd scenes. An interaction energy potential function is adopted in Cui *et al.* (2011) to capture the typical behaviors of pedestrians, whereas abnormal activities are indicated as unusual energy-velocity patterns.

Adam *et al.* (2008) propose a real-time approach to detect unusual event using multiple fixed-location monitors, where the monitors are deployed evenly in the spatial location of a frame. In each monitor, the Sum of Square Differences (SSD) (Shechtman and Irani, 2007) error matrix corresponding to discrete shifts in a certain window surrounding the monitor pixel is computed and transform into a probability distribution. An event is normal only if the distribution difference between the new observed event and the pre-set most likely event is less than a threshold, otherwise is abnormal. This algorithm is especially effective in detecting abrupt change of speed or direction of pedestrian. However, it is not adaptive and cannot detect more complicated abnormal events without considering temporal context.

Mehran *et al.* (2009) use the social force model (Helbing and Molnar, 1995) to estimate the interaction force between pedestrians. Specifically, a grid of particles over a image is deployed, where the average optical flow is computed in each particle. The particles are moved with the flow field they overlay, an approach termed as particle advection (Ali and Shah, 2007). The interaction force is estimated as the difference between the desired velocity of a particle and its actual velocity. After the interaction force has been extracted in every particle, BoW is built to represent a video clip, LDA is used for AED. While the social force model is very effective in detecting a global abnormal event, i.e., abnormal event in the whole frame, it is still challenging to detect a local abnormal event.

PGMs have also widely deployed for AED. Boiman and Irani (2007) try to compose a new video clip by means of chunks of video data that are extracted from training data. Regions in the new video that can be composed by large contiguous chunks of video data in training set are considered as normal, otherwise are suspicious/abnormal. This approach, which is termed as "inference by composition", implements by a Bayesian network to incorporate both the appearance and spatial location of chunks of video data. It shows a good performance in detecting suspicious behaviour, nevertheless, the huge size of the example database results in high computational cost for spatial-

temporal patch searching. A similar idea is found in Roshtkhari and Levine (2013b) as well. By constructing a hierarchical codebook model, the approach speeds up the process significantly.

Another Bayesian network model, entitled "Video parsing", is proposed in Antić and Ommer (2011). Given a frame, the foreground probability for every pixel is computed using a background subtraction approach. A linear SVM classifier is trained on the foreground probability map to distinguish background from anything else, where a set of hypotheses are detected. As a result, a hypothesis is denoted as location, scale, velocity, and appearance descriptor. Similar to the 'inference by composition', a hypothesis which has a high matching cost with any hypotheses in training data is regarded as abnormal, otherwise is normal. Furthermore, foreground probability map improves detection results by removing background pixels. It demonstrates promising performance in anomaly detection. Nevertheless, because non-parametric and data-driven, high computational cost is required to search data.

While most of existing approaches regard the motion features extracted from each location independent and identically distributed, some approaches also model the spatial/spatial-temporal dependencies between them. For example, Kratz and Nishino (2009) capture the spatial relationships between local spatial-temporal motion patterns, represented as statistical spatial-temporal gradient in Eq. (2.8), by HMM. Another example is in Kim and Grauman (2009), where the Mixture of Probabilistic Principal Component Analyzers (MPPCA) (Tipping and Bishop, 1999) models the local activity patterns and a spatial-temporal MRF model detects abnormal events in video by considering spatial and temporal dependencies.

Sparse coding is another hot research topic for AED. The basic concept (Wright *et al.*, 2009) is based on the idea that normal events are more easily to be reconstructed from a learnt dictionary than abnormal events. In other words, the reconstruction cost for abnormal events is much higher than that for normal events. In Cong *et al.* (2011); Zhu *et al.* (2014), motion patterns are represented as Multi-scale Histogram of Optical Flow (MHOF), where a dictionary $\mathbf{D}$ is learnt, with the $L_{2,1}$ norm as the sparsity function. The major difference of Zhu *et al.* (2014) from Cong *et al.* (2011) lies in that replacing Euclidean distance with wavelet Earth Movers Distance (EMD) as the objective function. A similar idea is proposed in Zhao *et al.* (2011), where motion features are detected through Cuboid (Dollár *et al.*, 2005); and the sparsity function is defined as the L_1 norm. Sparse Combination Learning (SCL) is proposed in Lu *et al.* (2013) to speed up the learning process. While the processing cost is reduced

significantly by SCL as well as resizing the frame resolution to a really small scale (i.e., 120×160), SCL is still a batch mode learning algorithm, thus it cannot handle concept drift which is a common occurrence in real-time video surveillance.

In Mahadevan *et al.* (2010); Li *et al.* (2014), Mixture of Dynamic Textures (MDT) (Chan and Vasconcelos, 2005a) is adopted to detect both spatial anomaly and temporal anomaly. To improve the detection performance, Li *et al.* (2014) propose a Hierarchical Mixture of Dynamic Texture (H-MDT) to detect temporal anomaly. The spatial location of a frame is divided into various subregions with multi-scale. A MDT is learnt from the spatial-temporal patches extracted from each subregion. As a result, multi-scale temporal anomaly maps are generated by the H-MDT. Inspired by the concept of saliency detection (Itti *et al.*, 1998; Hou and Zhang, 2007; Mahadevan and Vasconcelos, 2010) that comparing the difference between the feature of a location and its surrounding features, spatial anomalies are the locations whose dynamic textures are different from those of their surrounding. The hierarchy of spatial anomaly maps are generated by employing the various surrounding window size. Finally, the abnormal events are detected using an online CRF to fuse anomaly maps that across time, space, and scale. It demonstrates that combining H-MDT with CRF improves the performance significantly compared with Mahadevan *et al.* (2010). However, the learning of the CRF filter requires training data annotated manually beforehand, making it impractical for real-time application.

Tziakos *et al.* (2009, 2010) employ a manifold learning algorithm to detect abnormal events. Motion descriptors are extracted from a region of interest, where the LE algorithm is adopted to project those features into a embedding space, the frames in a testing video clip are labelled as abnormal if their Mahalanobis distances to the training samples are greater than a threshold.

2.5 Summary

This chapter reveals some basic but popular feature detectors and descriptors in video content analysis, manifold learning algorithms in computer vision together with some machine learning techniques that are used for crowd scenes analysis. Overall, there are still some limitations need to be addressed.

Firstly, each feature detector/descriptor has its own strengths and weaknesses, and may be applied to a specific scenario. However, most of them are not applicable to crowd scene analysis due to the following reasons. Firstly, while video surveillance in

crowd scene displays high density of pedestrians, the feature detectors, e.g., Harris3D, usually generate sparse interest points, which may result in some missing detection. As a result, dense sampling combined with some preprocessing technique such as background subtraction to filter out noise is the most effective feature detecting scheme. Secondly, most of the aforementioned feature descriptors exhibit discriminative power while increasing computational cost, which cannot meet the real-time response requirement in video surveillance.

Secondly, while LE has been employed for crowd scene analysis, it is a non-linear manifold learning algorithm which operates in a batch mode. In other words, it is unclear how to embed new coming frames into the subspace unless re-training the model. This is much more important as in video surveillance, the video data is usually very large and arrives in a stream fashion. As a result, there is demand for replacing the non-linear algorithm with a linear dimension reduction algorithm.

Finally, although the existing work for crowd scene analysis has achieved satisfactory results, they still cannot meet all the challenges in crowd scene analysis, namely, online learning as well as real-time response. Therefore, we aim to address the aforementioned limitations in the remaining chapters.

Chapter 3

Research Methodology

3.1 General approach

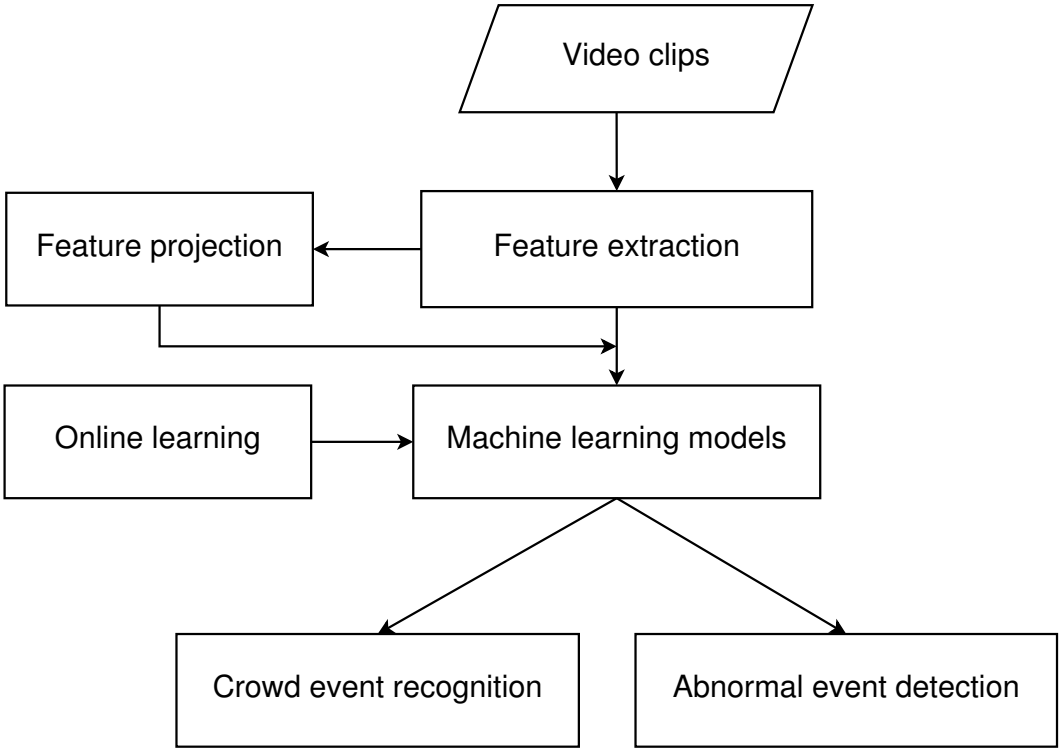


Figure 3.1: The flowchart of a general approach in the thesis.

A general approach proposed in the thesis is displayed in Figure 3.1. Compared with the conventional approach in Figure 1.1, there are two modifications. Firstly, in Chapter 4, considering the temporal correlation between frames in video clips as well as the curse of dimensionality resulted from the high-dimensional features, we propose a dimension reduction algorithm to compute the low-dimensional feature representations,

a step called "feature projection". Those low-dimensional feature representations are utilized for CER and AED with machine learning models. Secondly, as video data is usually very large and arrive sequentially, we propose an online learning approach of OCSVM classifier for AED. The details of the "online learning" step are discussed in Chapter 5 and 6.

3.2 Performance evaluation

Various criteria that evaluate the performance for different tasks have been considered in this thesis.

As the CER is a multi-class classification problem, we adopt accuracy to measure the performance, i.e.,

$$\text{accuracy} = \frac{\text{correct recognized events}}{\text{total number of events}}. \tag{3.1}$$

For one-class classification problem, namely AED, the performance is evaluated by drawing the Receiver Operating Characteristic (ROC) curve and computing the area under the ROC curve (AUC) (Fawcett, 2006) - Figure 3.2(a). More specifically, we tune the threshold on the detection result and draw the ROC curve, where x and y axis are False Positive Rate (FPR) and True Positive Rate (TPR), defined as:

$$\text{FPR} = \frac{\text{false positive}}{\text{false positive} + \text{true negative}},$$

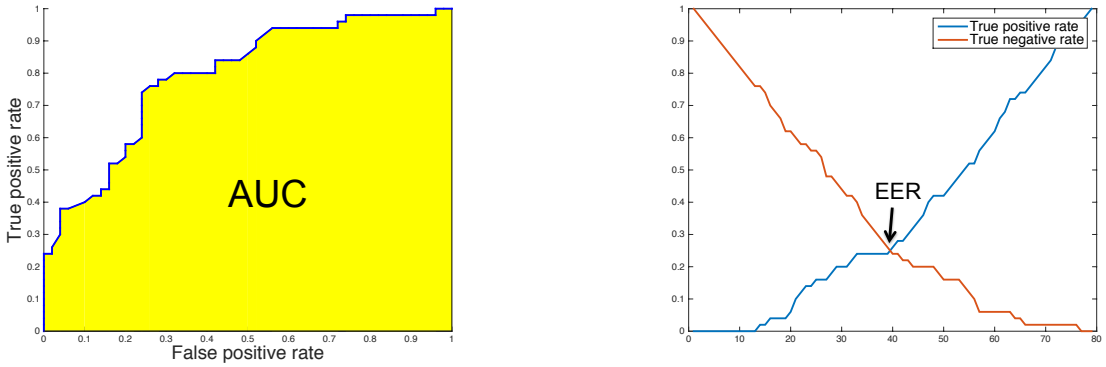
$$\text{TPR} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}},$$

where "true positive" and "true negative" correspond to the numbers of correctly detected normal and abnormal frames respectively, "false negative" is the number of false alarm frames, and "false positive" is the number of missed abnormal frames.

On the basis of ROC curve, AUC and/or Rate of Detection (RD) are employed to compare with other approaches. The latter is defined as RD $= 1 -$ EER, where EER stands for the Equal Error Rate (EER), corresponds to the point on the ROC curve where true positive rate equals to true negative rate (i.e., 1 - false positive rate) - Figure 3.2(b). A good performance is supposed to have a low EER and hence a high RD.

For comparison purposes, three criteria are selected to evaluate the Shot Boundary Detection (SBD) performance, i.e., recall, precision, and F_1, given as:

$$\text{recall} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}},$$



(a) (b)

Figure 3.2: (a) A ROC curve example, where the yellow area corresponds to its AUC. (b) The EER corresponds to the intersection point of true positive rate curve and true negative rate curve.

$$\text{precision} = \frac{\text{true positive}}{\text{true positive} + \text{false negative}},$$

$$F_1 = 2 \cdot \frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}},$$

where "true positive" and "false positive" correspond to the numbers of correctly and falsely detected shot boundaries respectively, and "false negative" is the number of missed shot boundaries. Following the TRECVID practice, a SBD outcome is considered "correct" if there is at least a one frame overlap between the detected transition and the annotated transition.

3.3 Datasets

To evaluate the performance of the proposed approaches, extensive experiments have been conducted on five different datasets throughout this thesis. Among them, PETS 2009 dataset[1] is used for CER. Meanwhile, UMN dataset [2], Subway Exit dataset (Adam *et al.*, 2008), and UCSD anomaly dataset[3] are used for AED. In addition, we adopt TRECVID 2007 SBD data[4] for SBD.

We adopt the five datasets for several reasons. Firstly, they have been widely used for evaluation in their individual topic. Secondly, it is very convenient to compare with

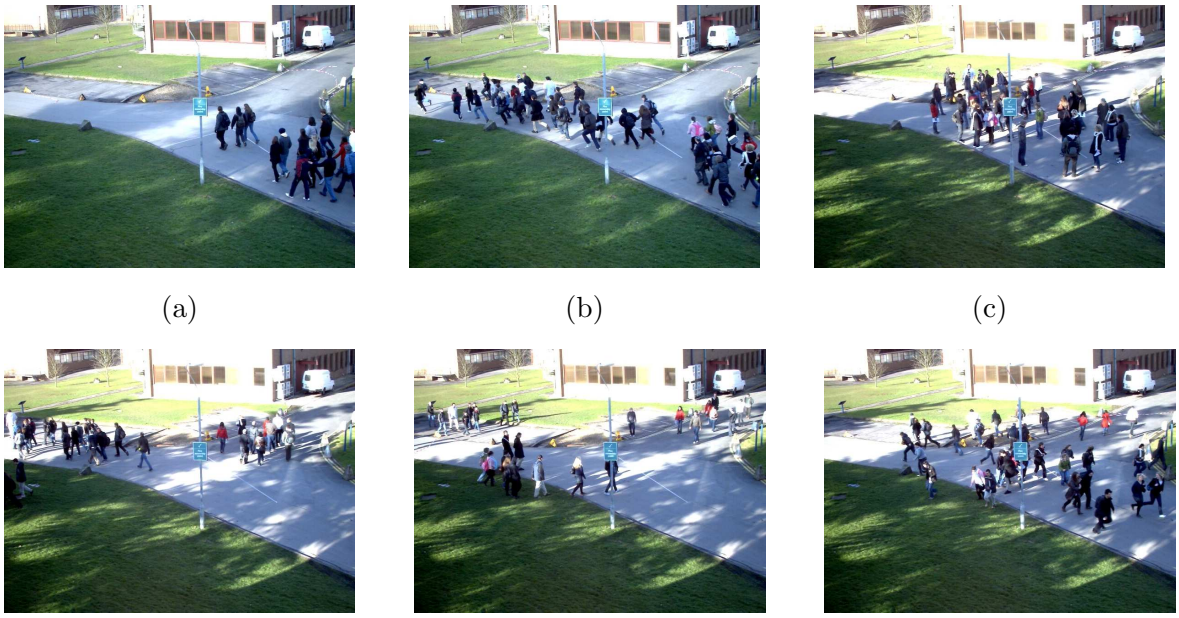
[1]http://www.cvg.rdg.ac.uk
[2]http://mha.cs.umn.edu
[3]http://www.svcl.ucsd.edu/projects/anomaly/dataset.html
[4]http://trecvid.nist.gov/trecvid.data.html

Table 3.1: Ground truth for CER in the PETS dataset

Classes	**Timestamp** [frames]
Walking	14-16 [0-36, 108-161] 14-31[0-50]
Running	14-16 [37-107, 162-223]
Local dispersion	14-16 [0-184, 280-333]
Local movement	14-33 [197-339] 14-27 [185-279]
Crowd splitting	14-31 [51-130]
Crowd formation	14-33 [0-196]
Evacuation	14-33 [340-377]

the state-of-the-art approaches on them as most of them have complete annotations. Details of these datasets are introduced as follows.

3.3.1 PETS 2009 dataset



(a) (b) (c)

(d) (e) (f)

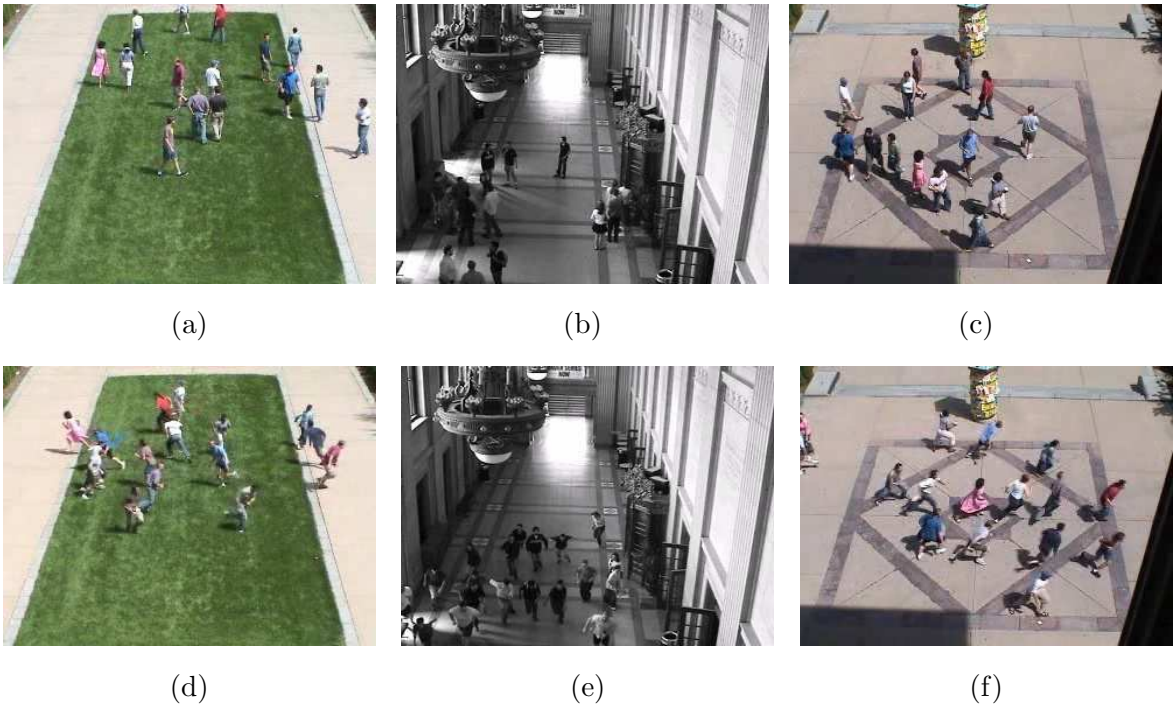
Figure 3.3: Sample frames in the PETS dataset. (a) Walking. (b) Running. (c) Local dispersion. (d) Splitting. (e) Formation. (f) Evacuation.

The PETS 2009 dataset contains four video clips (1066 frames in total) of four views with timestamps 14-16, 14-27, 14-31 and 14-33 (here we use view "001"). Crowd events vary over time as follows: walking, running, local movement, local dispersion,

splitting, formation and evacuation. Figure 3.3 illustrates some sample frames of these categories. For comparison purpose, we use the same ground truth as in Thida *et al.* (2012), as shown in Table 3.1. However, one drawback of PETS is that it is synthetic for data collection. In other words, all the shots are staged.

3.3.2 UMN dataset

The UMN dataset contains eleven video clips of three different scenes. More precisely, two clips of Scene 1 (outdoor), six clips of Scene 2 (indoor), and three clips of Scene 3 (outdoor). The total length of the UMN dataset is $7,739$ frames with a resolution of 320×240 pixels. Each clip starts with normal activities and ends with some anomalies – see Figure 3.4. Motion patterns in different clips vary.



(a) (b) (c)

(d) (e) (f)

Figure 3.4: Sample frames in the UMN dataset. The top row represents normal frames in the dataset, and the bottom row corresponds abnormal frames. (a)(d): Scene 1; (b)(e) Scene 2; (c)(f) Scene 3.

However, as mentioned in Li *et al.* (2014), the UMN shows some limitations. Firstly, the same as PETS, all shots are staged. Secondly, the abnormal event happens in the whole frame and produce salient motion changes, which is easy to detect and thus achieve promising performance.

3.3.3 Subway Exit dataset

(a) (b) (c)

Figure 3.5: Examples of abnormal events in Subway Exit dataset, where red boxes correspond to abnormal events.

The length of Subway Exit video is about 43 minutes of 64,901 frames in total. To make a qualitative comparison with other state-of-the-art methods, we define the same three types of abnormal events: wrong direction, loitering, and miscellaneous (including cleaning the wall, etc.). Although the above definitions of abnormal events are rather subjective, we have 19 abnormal events in total as ground-truth - Table 3.2. Some abnormal events are illustrated in Figure 3.5.

Following Zhao *et al.* (2011), anomaly detection is considered as correct if at least one abnormal event is detected in an annotated frame range, whereas detection is deemed as a false alarm if at least one anomaly event is triggered outside the annotated range.

Compared with the UMN dataset, this dataset is much more natural. However, the main limitations are twofold. Firstly, most of frames in video sequence is redundant as no pedestrian appear in them. Secondly, the evaluation metric is too coarse because lack of accurate groundtruth annotation.

Table 3.2: Groundtruth of Subway Exit dataset.

	Wrong direction	Loitering	Misc	Total
Ground truth	9	3	7	19

3.3.4 UCSD anomaly detection dataset

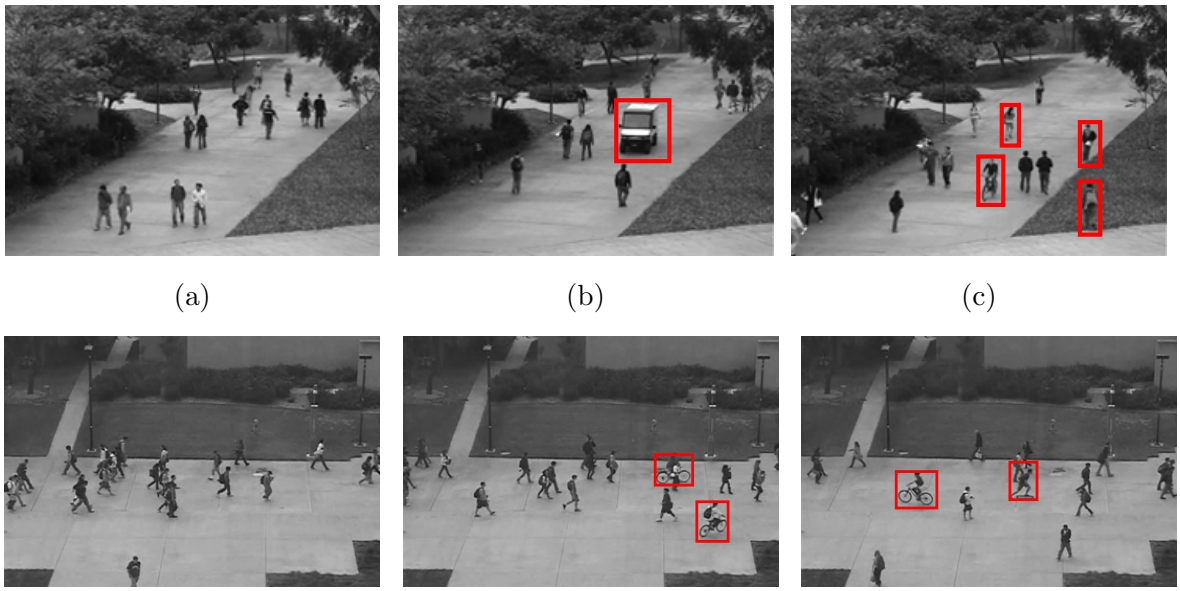
The UCSD anomaly detection dataset contains two subsets, acquired by a stationary camera from two different pedestrian walkways - Figure 3.6. The first set, denoted as

"Ped1", has 34 training clips and (6,800 frames) and 36 testing clips (7,200 frames), each with 158 × 238 resolution and a fixed length of 200 frames, while the second, denoted as "Ped2", has 16 training clips (2,550 frames) and 12 testing clips (2,010 frames), each with a resolution of 240 × 360, and the length varying from 120 to 180 frames. In general, Ped1 is more challenging than Ped2 as the angle of camera results in larger perspective distortion and more significant scale variation of motion objects. Furthermore, abnormal events in Ped1 include not only motion anomalies caused by bikers, skateboarders and small carts etc., but also contextual anomalies, e.g., pedestrian walking across the lawn. The composition of abnormal events in each set is illustrated in Table 3.3.

Table 3.3: Composition of abnormal events in the UCSD dataset.

	Bike	Skater	Cart	Walk across	Other	Total
Ped1	28	13	6	4	3	54
Ped2	19	3	1	0	0	23

The same as the UMN dataset, AUC and RD are employed for evaluation. However, Mahadevan *et al.* (2010); Li *et al.* (2014) propose to evaluate on two levels: frame level and pixel level. For the former an anomaly detection is counted correct if a frame is related to an abnormal event regardless of the anomaly location within the frame, while for the latter the anomaly detection is counted correct only if at least 40% of the truly anomalous pixels are detected. As frame-level can only measure temporal localization accuracy, the pixel-level evaluation is much more accurate as it rules out some arbitrary anomaly detection, i.e., a random location in an anomaly frame is detected as an anomaly.



(a) (b) (c)

(d) (e) (f)

Figure 3.6: Sample frames in the UCSD dataset, where top row correspond to Ped1 data, and bottom row correspond to Ped2 data. Red boxes correspond to annotated abnormal events.

The Ped1 dataset published by (Mahadevan *et al.*, 2010) has only 10 clips (out of 36 clips) annotated with pixel-level ground truth. To enhance the utilization of this dataset, more annotations were given by (Antić and Ommer, 2011) for the rest of the clips[5]. We denote the former as "partial annotation", while the latter as "full annotation".

Compared with UMN data and Subway Exit data, UCSD data is much more challenging due to a broad spectrum of abnormal event types. Meanwhile, due to the complete annotation, it is convenient to compare the evaluation results with the state-of-the-arts approaches. Therefore, the experiments for AED are mainly carried on this dataset.

3.3.5 TRECVID 2007 SBD dataset

The TREC Video Retrieval Evaluation (TRECVID) is an annual worldwide benchmarking activity, whose goal is to encourage research on content-based information retrieval in digital video. SBD task is one of the many evaluation tasks between 2001 and 2007 (Smeaton *et al.*, 2010). The TRECVID 2007 SBD dataset contains 17 video

[5]http://hciweb.iwr.uni-heidelberg.de/compvis/research/parsing

sequences of 637,805 frames, where 2,463 shots are annotated - Table 3.4. Among them, 90% shots are hard cuts, and the rest are gradual transitions. Contents of the dataset are diverse, covering a wide range from news broadcast clips to archived grayscale videos, which is different from previous years, where only news report videos are included.

Table 3.4: Shot composition in TRECVID 2007 SBD dataset.

Video ID	Number of frames	Hard cut	Gradual transition	Total
BG_2408	35,892	101	20	121
BG_9401	50,049	89	3	92
BG_11362	16,416	104	4	108
BG_14213	83,115	106	61	167
BG_34901	34,389	224	16	240
BG_35050	36,999	98	4	102
BG_35187	29,025	135	23	158
BG_36028	44,991	87	0	87
BG_36182	29,610	95	14	109
BG_36506	15,210	77	6	83
BG_36537	50,004	259	30	289
BG_36628	56,564	192	10	202
BG_37359	28,908	164	6	170
BG_37417	23,004	76	12	88
BG_37822	21,960	119	10	129
BG_37879	29,019	95	4	99
BG_38150	52,650	215	4	219
Total	637,805	2236	227	2463

Chapter 4

Crowd Scene Analysis using adaptive quantization and manifold learning

4.1 Overview

Recently, manifold learning in computer vision has become a popular research topic (Turk and Pentland, 1991; Tenenbaum *et al.*, 2000; He *et al.*, 2005; Wang and Suter, 2007, 2008; Elgammal and Lee, 2009; Tziakos *et al.*, 2010; Torki and Elgammal, 2010b; Thida *et al.*, 2010; Saghafi and Rajan, 2012). Earlier work concentrated on dimension reduction algorithms for face recognition (Turk and Pentland, 1991; He *et al.*, 2005). Originally Turk and Pentland (1991) adopt PCA to embed high-dimensional face images into a linear low-dimensional subspace. These significant features in subspace they termed as "eigenfaces". Using the eigenfaces, a new face image is embedded into the subspace initially and determined if it is a face image by checking if the image is sufficiently close to "face space". If it is a face image, it is classified either a known person or not. Kim *et al.* (2002) extends the same idea for face recognition from PCA to kernel PCA (Schölkopf *et al.*, 1998). To address the limitations of large variation in lighting direction and facial expression, Belhumeur *et al.* (1997) propose fisherfaces on the basis of FLD. Compared with eigenfaces, fisherfaces achieve greater between-class scatter, thus simplify face recognition problem. More than a decade later, He *et al.* (2005) build on the "eigenfaces" concept and propose Laplacianfaces based on the LPP algorithm. Unlike analysing the sample covariance matrix in eigenfaces, Laplacianfaces is aiming at preserving local information and obtaining a face subspace that best detects the

essential face manifold structure.

Extending face recognition to other computer vision topics, Isomap is used to represent a set of images as a set of points in a low dimensional space (Pless, 2003). In this way a video can be considered to have two parts: a space of possible images and a trajectory through the image space. This gives rise to a new tool for video analysis by examining the video trajectory, from which events can be segmented and anomalies can be detected. Following a similar idea, explicit representations for dynamic shape manifolds of moving humans can also be learnt through LPP, which is used for action recognition (Wang and Suter, 2007). Other attempts that apply LE to object categorization and feature-spatial matching have been proposed in Torki and Elgammal (2010a,b). To maintain the discrimination of local features while keeping their spatial arrangement information, an adjacency graph that preserves intra-image spatial structure and inter-image feature affinity is constructed. Focusing on the analysis of crowd scenes, Tziakos *et al.* (2009, 2010) detect unusual events in highway by combining LE with temporal information. In Thida *et al.* (2010, 2012), a framework was proposed for event detection using optical flow and Spatio-Temporal Laplacian Eigenmaps (ST-LE). However, the major drawback of LE is that it is a non-linear algorithm which operates in a batch mode. In other words, the generalized eigenvalue problem of LE has to be re-computed when new frames are arriving, which is computationally intensive and unsuitable for video analysis.

To automatically recognize different type of crowd events and detect anomalous crowd events, in this chapter, we propose a framework for crowd scene analysis in video surveillance by learning video manifolds. A new feature code designed by applying adaptive quantization and binarization is adopted so as to increase the dissimilarity between motion patterns. We propose a novel linear dimensionality reduction algorithm that considers both spatial and temporal similarities between frames when generating the video manifold. Experimental results demonstrate that our new method outperforms the state-of-the-art methods.

4.2 The computational framework

Our computational framework is shown in Figure 4.1. Given a video clip, a Histogram of Optical Flow (HOF) descriptor is first extracted between two successive frames. On the basis of HOF, we generate a Quantized Binary Code (QBC) for every frame. Then all the QBCs are embedded into a low-dimensional manifold using a novel algorithm

called "Spatial-Temporal Locality Preserving Projections" (STLPP). CER and AED are eventually conducted on the low-dimensional manifolds using 1-Nearest Neighbour (1-NN) and OCSVM respectively. Compared with the conventional crowd scene analysis approach, we use STLPP to embed high-dimensional features into low-dimensional subspaces while incorporating both spatial and temporal similarities. Details of these algorithmic steps are explained as follows.

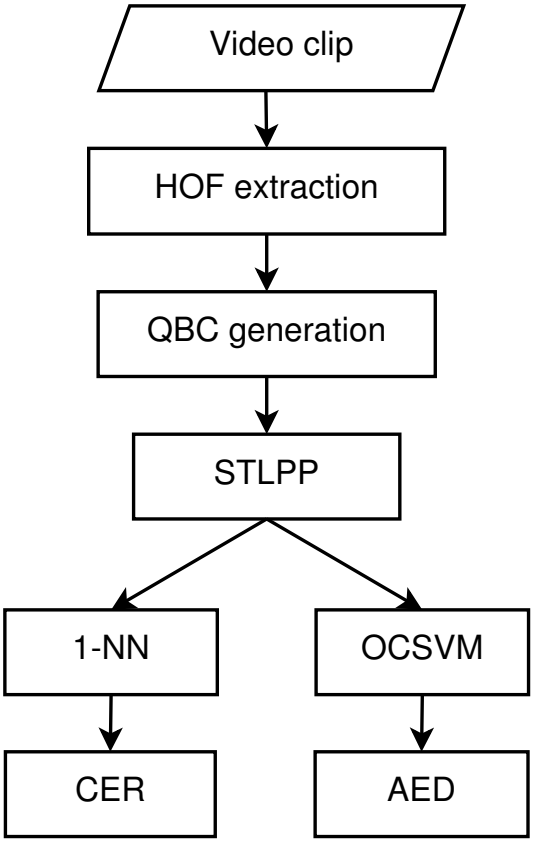
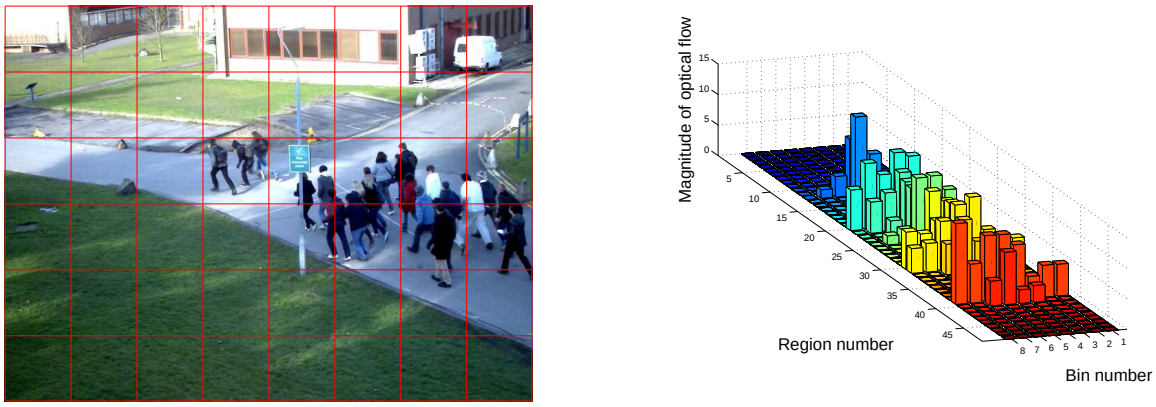


Figure 4.1: The computational framework for crowd scenes analysis.

4.2.1 HOF extraction

We adopt a matrix of size $W \times H \times L$ for a video clip, where W and H are the width and height of the video frame respectively, and L is the total number of frames of the clip. Optical flow between successive frames is estimated according to Horn and Schunck (1981). The optical flow vector of the frame at time t is denoted as: $\{(f_{x,t}, f_{y,t})\}$. Then, insignificant values due to camera motion or noise are removed based on a predefined threshold.



(a) (b)

Figure 4.2: The splitting of frames into 6 by 8 regions (a) and the corresponding generation of HOF (b).

The orientation α of a nonzero optical flow vector is determined as:

$$\alpha = \begin{cases} \tan^{-1}(\frac{f_y}{f_x}) & f_x > 0 \\ \tan^{-1}(\frac{f_y}{f_x}) + \pi & f_y \geq 0, f_x < 0 \\ \tan^{-1}(\frac{f_y}{f_x}) - \pi & f_y < 0, f_x < 0 \\ +\frac{\pi}{2} & f_y > 0, f_x = 0 \\ -\frac{\pi}{2} & f_y < 0, f_x = 0 \end{cases} \tag{4.1}$$

with $-\pi < \alpha < \pi$. Following the same idea of previous work (Dalal and Triggs, 2005; Chaudhry *et al.*, 2009), we then construct a 8-bin histogram for α.

We compute the average magnitude of optical flows in each bin, and a histogram stacking average flows within $m \times n$ regions is formed. Figure 4.2 shows an example of the feature extraction process. We partition a frame into 48 (i.e., 6×8) regions - Figure 4.2(a), and every region has 8 bins. This results in a 2-D histogram in Figure 4.2(b). The corresponding direction with higher magnitudes in the histogram indicates the main directions of crowd motions, and the region index indicates the position of the crowd in the frame.

4.2.2 QBC generation

Although HOF features were successfully applied for the purpose of CER and AED (Thida *et al.*, 2010, 2012), there are questions around the stability and performance

of this encoding scheme under different degrees of temporal similarity. Furthermore, as our overall objective is to move towards real-time analysis of on-line video stream data, we are also interested in how we can create more compact representations for video manifold learning with a view to reducing the computational complexity whilst, maintaining or improving the discriminant ability of crowd motion patterns.

Given this motivation and based on the 2-D optical flow histogram, we next propose a novel feature scheme called *Quantized Binary Code* (QBC). Rather than using the optical flow histogram vectors directly for further computational procedures, we transform the histogram code through quantization and binarization. Our intention is to make the transformed feature code more discriminative for different motion types.

For each region we have a 8-dimension flow vector, f, after feature extraction. QBC scheme first discretizes the continuous flow value into n discrete intervals $\{(0, \theta_1], (\theta_2, \theta_3], \ldots, (\theta_{n-1}, +\infty)\}$, where n is the quantization level and $\theta_i (i = 1, \cdots, n-1)$ are threshold vectors. The flow vector is then expanded into a QBC vector as a n-tuple $\mathbf{t} = \{\mathbf{c}_i | i = 1, \ldots, n\}$, where the length of each tuple $\mathbf{c}_i$ is the same as the number of dimension of flow vector, i.e., 8. The binary element values in each tuple are assigned with the help of threshold vector θ_i:

$$\begin{aligned} &\mathrm{c}_{1j} = 1, && \text{IF } 0 < f_j \leqslant \theta_1; \\ &\mathrm{c}_{n-1j} = 1, && \text{IF } \theta_{n-2} < f_j \leqslant \theta_{n-1}; \\ &\mathrm{c}_{nj} = 1, && \text{IF } f_j > \theta_{n-1}; \end{aligned} \tag{4.2}$$

where $j = 1, 2, \ldots, 8$. Otherwise, these elements will remain 0 as initialized.

Figure 4.3 shows an example of QBC generation. The QBC has 16 binary elements (i.e., $n = 2$), the first 8 of them corresponding to $\mathbf{c}_1$ whether the magnitude of optical flow on 8 directions is less than the threshold θ respectively, and the next 8 elements corresponding to $\mathbf{c}_2$ whether the magnitude of optical flow is equal to or greater than θ_n. Threshold θ_1 is indicated by the red dotted line in the diagram.

As can be seen in this figure, similar activities in different regions generate about the same scale of flow values setting bits 1–8 of the QBC feature to 1 and leaving bits 9–16 at 0. Where the activities are different, this causes the magnitude of optical flow to become greater than the threshold, θ_i, therefore in the instances where this occurs, specific elements of the latter 8 bits of the QBC feature are set to 1. In the case of Figure 4.3 as bins 1, 5, and 8 exceed the threshold, θ_i, then bits 9, 13, and 16 of the QBC feature are set to 1 and bits 1, 5, and 8, of the same feature remain, as initialized, at 0.

Note that when an optical flows amplitude is too low or zero, it will result in 0

among both the 1-8 bits and the 9-16 bits. Therefore the two parts here in QBC are not complementary: A 0 in a lower bit does not necessarily mean a 1 in the corresponding higher bit, and a 0 in a higher bit does not imply a 1 in the lower bit either.

Despite the quantization and binarization operations, the QBC feature not only still preserves both the directional and magnitude information of local motions, but also is more discriminative for different motion types than the optical flow histogram. When $n > 2$, more quantization levels are used, resulting in a QBC code of higher dimensionality, which may contribute to potentially better discriminant ability. The effect of n on performance will be examined in Section 4.3.1.

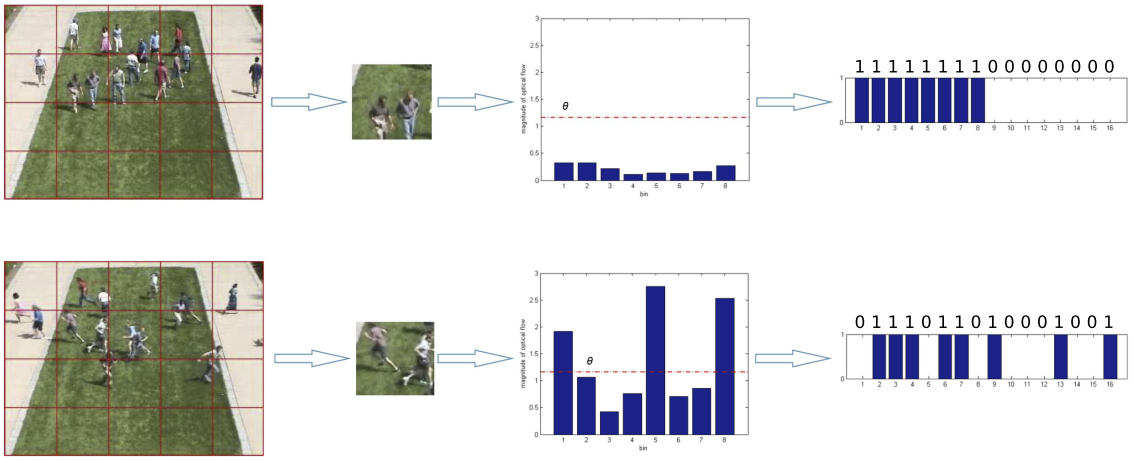


Figure 4.3: Generating QBC from HOF.

4.2.3 STLPP

Here we extend the original LPP algorithm (Niyogi, 2004) by considering both spatial and temporal similarities of events in crowd scenes. LPP provides a good foundation for our work as it has a number of interesting characteristics which can be subsequently leveraged for real-time video analysis. Firstly, LPP differs from other dimensionality reduction algorithms in that it is a linear transformation which makes it less computationally expensive. Secondly, it has reported good performance in terms of discriminating power. This is an important feature as we require a representation to facilitate clearer separation between "normal" and "abnormal" video frames. And thirdly, the most advantageous feature of LPP is that not only can it be defined on the training points, it can also accommodate any new incoming data points locating them in the reduced representation space (Niyogi, 2004); this is however is a weakness of other di-

mensionality reductions methods such as LE. Such desirable feature is important since we are dealing with a stream of video data so we need to consider what computationally efficient dimensionality reduction methods, which can map all data points, could be used for incremental learning algorithms such as online OCSVM (Wang *et al.*, 2013) as this is our ultimate goal.

Unlike the original LPP algorithm, however, STLPP utilizes both spatial and temporal information for manifold learning. We argue that the inclusion of temporal information should result in improved recognition performance. Although we could have adopted ST-LE (Thida *et al.*, 2010, 2012) to incorporate this temporal information, LPP has a number of advantages over LE which have been previously described. Furthermore we argue that ST-LE amplifies the effect of temporal information to such a degree that it reduces the contribution of the spatial component as discussed in experimental results, therefore an alternative method for preserving local structure needs to be considered. The algorithm is presented as follows.

Constructing the weight matrix

Suppose there are m frames in a clip, and each frame has its QBC vector $\mathbf{x}_i$ extracted, $i = 1, \ldots, m$. Let W be a symmetric $m \times m$ matrix, with the weight between frame i and frame j given as:

$$w_{ij} = \mathrm{SS}_{ij} \times \mathrm{TS}_{ij}, \tag{4.3}$$

where SS_{ij} denotes the spatial similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$, i.e., the QBC vectors of frame i and frame j respectively:

$$\mathrm{SS}_{ij} = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\parallel \mathbf{x}_i \parallel \parallel \mathbf{x}_j \parallel}, \tag{4.4}$$

and TS_{ij} denotes the temporal similarity between the two frames:

$$\mathrm{TS}_{ij} = e^{-\frac{(i-j)^2}{\sigma^2}}, \tag{4.5}$$

where σ is a parameter controlling the effective scope for temporal similarity.

Generating eigenmaps

Having obtained the similarity matrix W, we deal with the following generalized eigenanalysis problem:

$$XLX^T\mathbf{v} = \lambda XDX^T\mathbf{v}, \tag{4.6}$$

where D is a diagonal matrix whose entries are column (or row) sums of W, i.e., $d_{ii} = \sum_j w_{ij}$; $L = D - W$ is the Laplacian matrix; and X denotes the data matrix whose i-th column $\mathbf{x}_i$ corresponds to the QBC vector of frame i.

Let column vectors $\mathbf{v}_0, \ldots, \mathbf{v}_{l-1}$ be the solutions of Eq. (4.6), with the corresponding eigenvalues in ascending order: $\lambda_0 <, \ldots, < \lambda_{l-1}$. Then, the l-dimensional embedding vector $\mathbf{y}_i$ corresponding to $\mathbf{x}_i$ is estimated by:

$$\mathbf{y}_i = V^T \mathbf{x}_i, \tag{4.7}$$

where $V = (\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_{l-1})$. One significant advantage of STLPP is that it is a linear dimensionality reduction algorithm. For a new arriving clip, we could simply embed it into low dimensional subspace using Eq. (4.7).

4.2.4 Machine learning models

Now the high-dimensional video frames are embedded into a low-dimensional manifold. A frame in a video clip is represented as a data point in the embedding space. A trajectory $\mathbf{s}_i$ is constructed as $\mathbf{s}_i = \{\mathbf{y}_1, \ldots, \mathbf{y}_T\}$, where T denotes the temporal window size, and $\mathbf{y}_t$ is the l-dimensional embedding vector obtained in Eq. (4.7). We employ learning models on the embedded manifolds for CER and AED.

1-NN for CER

To recognize crowd events, we measure the distance between the reference trajectory and the test trajectory in the low-dimensional embedding space. Specifically, given two trajectories $\mathbf{s}_1$ and $\mathbf{s}_2$, Hausdorff Distance (Huttenlocher *et al.*, 1993) is used to compute their distance:

$$H(\mathbf{s}_1, \mathbf{s}_2) = \max(h(\mathbf{s}_1, \mathbf{s}_2), h(\mathbf{s}_2, \mathbf{s}_1)), \tag{4.8}$$

where

$$h(\mathbf{s}_1, \mathbf{s}_2) = \max_{\mathbf{y}_i^1 \in \mathbf{s}_1} \min_{\mathbf{y}_j^2 \in \mathbf{s}_2} \|\mathbf{y}_i^1 - \mathbf{y}_j^2\|, \tag{4.9}$$

and $\|\cdot\|$ is the Euclidean norm.

Crowd events are classified by the 1-NN algorithm. Denote the training set of trajectories as TR. For a test trajectory $\mathbf{s}_{te}$, it is classified as Class c if it finds the nearest match in TR with a class label c:

$$c = \arg\min_c H(\mathbf{s}_{te}, \mathbf{s}^c), \ \forall\ \mathbf{s}^c \in \text{TR}, \text{class}(\mathbf{s}^c) == c. \tag{4.10}$$

Note that a more sophisticated classifier could be employed, but here we concentrate on evaluating the 1-NN discriminative ability of QBC and STLPP.

OCSVM for AED

We adopt the OCSVM for AED, which has already been applied for the purposes of document classification (Manevitz and Yousef, 2002), time-series novelty detection (Ma and Perkins, 2003), and handwritten signature verification (Guerbai *et al.*, 2014).

Given a sample of normal trajectories $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$, OCSVM aims to find an optimal hyperplane that separates the data points from the origin in the feature space:

$$f(\mathbf{s}) = \mathbf{w} \cdot \Phi(\mathbf{s}) - \rho, \tag{4.11}$$

where Φ is a feature map that transforms $\mathbf{s}$ from the input space $\mathcal{S}$ to the feature space $\mathcal{F}$.

To acquire the optimal parameters $\mathbf{w}$ and ρ, one can solve the following quadratic programming problem:

$$\begin{aligned} &\min_{\mathbf{w},\xi,\rho} : \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i - \rho \\ &s.t. : \mathbf{w} \cdot \Phi(\mathbf{s}_i) \geq \rho - \xi_i,\ \xi_i \geq 0, \end{aligned}$$

where ξ_i are slack variables, and C is a regularization parameter, controlling the trade-off between structure and empirical risks.

Through analysis employing Lagrangians, the problem can be transformed to the following dual problem:

$$\begin{aligned} &\min_{\alpha} : \frac{1}{2}\sum_{ij} \alpha_i \alpha_j k(\mathbf{s}_i, \mathbf{s}_j) \\ &s.t. : 0 \leq \alpha_i \leq C,\ \sum_i \alpha_i = 1, \end{aligned} \tag{4.12}$$

where α_i represents the *weight* on vector $\mathbf{s}_i$, and $k(\mathbf{s}_i, \mathbf{s}_j) = \Phi(\mathbf{s}_i) \cdot \Phi(\mathbf{s}_j)$ is a kernel function that measures the similarity between the two trajectories $\mathbf{s}_i$ and $\mathbf{s}_j$, with a maximum of 1 and a minimum of 0.

4.3 Experimental result

To evaluate the performance of the proposed STLPP framework, we conduct experiments using two benchmark datasets: the PETS 2009 dataset and the UMN dataset.

Table 4.1: The quanta matrix for a histogram bin.

Class	Interval $(0, \theta_1] \dots (\theta_{j-1}, \theta_j] \dots (\theta_{n-1}, +\infty)$	Sum of class
C_1	$a_{11} \dots a_{1j} \dots a_{1n}$	A_{1+}
$\vdots$	$\vdots \dots \vdots \dots \vdots$	$\vdots$
C_i	$a_{i1} \dots a_{ij} \dots a_{in}$	A_{i+}
$\vdots$	$\vdots \dots \vdots \dots \vdots$	$\vdots$
C_m	$a_{m1} \dots a_{mj} \dots a_{mn}$	A_{m+}
Sum of intervals	$A_{+1} \dots A_{+j} \dots A_{+n}$	A

In general we adopt the same or similar experiment settings to the relevant literature so as to make the results comparable. Details of experimental settings and results are presented here, for CER and AED respectively.

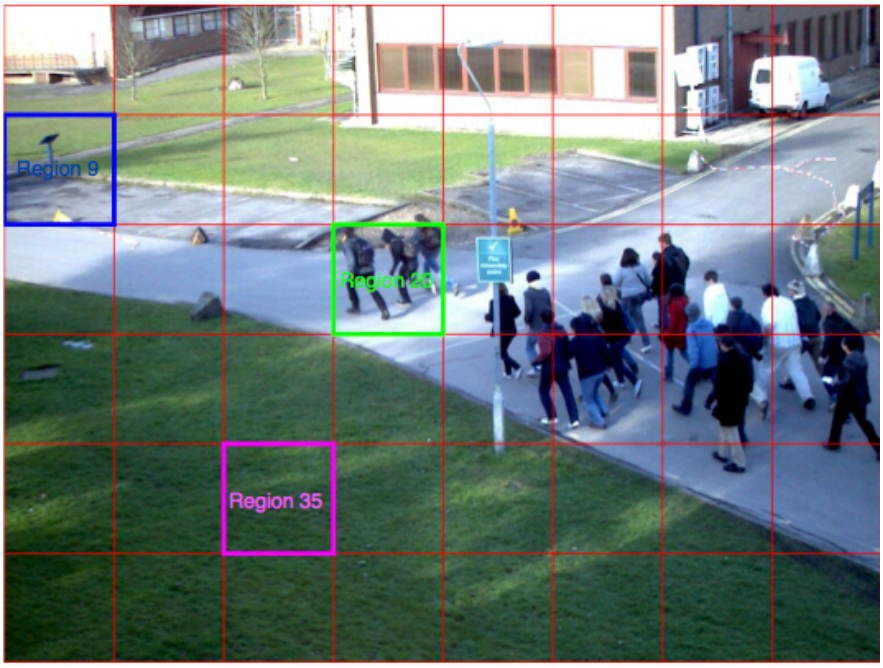
4.3.1 CER result

Evaluation of QBC

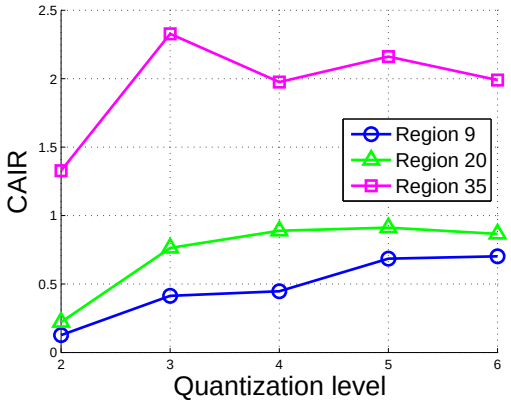
Our first experiment is to investigate the effects of QBC, compared with that of the plain optical flow histograms as in previous work. In our implementation, the binarization threshold is derived individually for each region, based on which the QBC for each frame is obtained.

We assess the quality of QBC using two measurements: the Class-Attribute Interdependence Redundancy (CAIR) (Ching *et al.*, 1995; Tsai *et al.*, 2008) and the Information Gain (IG) (Kullback, 1968). Both measurements are widely used for feature selection, and for evaluating the quality of discretization schemes as they can effectively represent the interdependency between the target class and discretized attributes.

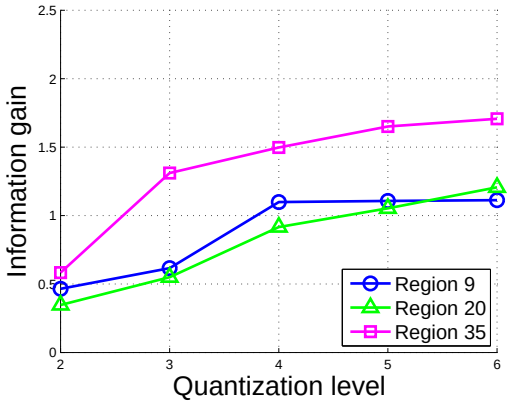
Assume for a histogram bin b, the $n-1$ discretization thresholds, $\theta_1, \cdots, \theta_{n-1}$ are used to create n intervals for a region. The intervals, and the occurrences of the feature value for each class form a so-called "quanta matrix", as shown in Table 4.1. The occurrence accumulation for each class is $A_{i+} = \sum_j a_{ij}$, and the accumulation per interval is calculated as $A_{+j} = \sum_i A_{ij}$, $i = 1, \cdots, m$, $j = 1, \cdots, n$. The total accumulation is denoted as A. Hence the probabilities for Class i regarding the j-th interval is then $p_{ij} = \frac{a_{ij}}{A}$, the probability for Class i is $p_{i+} = \frac{A_{i+}}{A}$, and the probability for Interval j is $p_{+j} = \frac{A_{+j}}{A}$.



(a)



(b)



(c)

Figure 4.4: The CAIR (b) and IG (c) performance comparision in terms of quantization level n from three randomly selected regions (a).

Using the above definitions, the CAIR and the IG for the current bin are defined as:

$$\text{CAIR} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}\log_2\frac{p_{ij}}{p_{i+}p_{+j}}}{\sum_{i=1}^{m}\sum_{j=1}^{n} p_{ij}\log_2\frac{1}{p_{ij}}} \tag{4.13}$$

$$\begin{aligned} \text{IG} = &\ H(C) - H(C|A) \\ = &\ -\sum_{i=1}^{m} p_{i+}\log_2 p_{i+} + \sum_{i=1}^{m}\sum_{j=1}^{n} p_{+j}p_{ij}\log_2 p_{ij} \end{aligned} \tag{4.14}$$

Eventually we obtain the average CAIR and IG across all histogram bins.

Figure 4.4 illustrates the CAIR and the IG performance with different quantization levels ($n = 2, \ldots, 6$) from three random selected regions (Region 9, 20 and 35). It is shown that even though their performance in each region is different from each other, it achieves better performance with $n = 3$ (Figure 4.4(b)) or 4 (Figure 4.4(c)). Moreover, the experiment shows that the higher quantization level does not lead to a significant improvement in performance.

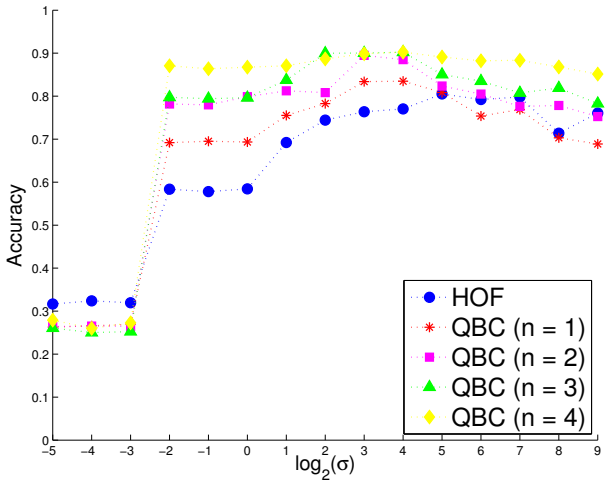
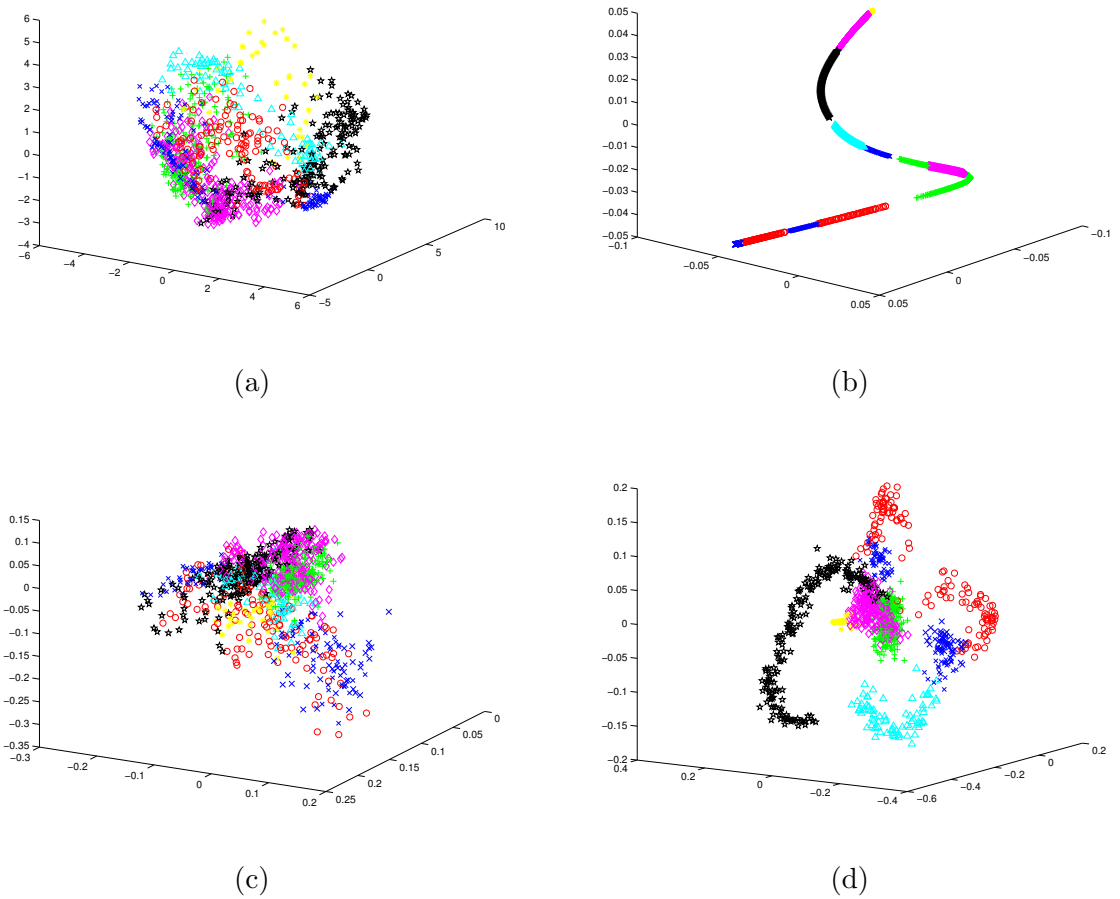


Figure 4.5: QBC vs HOF. Best viewed in colour.

Next, we set the embedding dimension l and the temporal window size T both to 3, obtaining trajectories of 9 dimensions; then a 4-fold cross-validation is employed on these trajectories to compute recognition accuracy with σ increasing exponentially ($\sigma = 2^{-5}, 2^{-4}, \ldots, 2^{9}$), and the scale with the best cross-validation accuracy is chosen. Meanwhile, the QBC ($n = 1, \ldots, 4$) and the HOF are tested respectively to verify the effectiveness of the QBC. The outcome is illustrated in Figure 4.5. It shows that, over a large range of σ values (from $\frac{1}{4}$ to 64), the performance of the QBC is more robust and better. Meanwhile, the performance of the QBC reaches its peaks when σ is set as 8 and 16 (i.e., 2^3 and 2^4), where the quantization level of $n = 3$ is sufficient to give competitive outcome.



(a) (b) (c) (d)

Figure 4.6: 3-D manifolds of PETS crowd events generated by different algorithms. (a) PCA. (b) ST-LE. (c) LPP. (d) STLPP. (Event keys: blue $\times$ - walking, red $\circ$ - running, green $+$ - local dispersion, magenta $\diamond$ - local movement, cyan $\triangle$ - splitting, black $\star$ - formation, yellow $*$ - evacuation)

Evaluation of STLPP

Furthermore, we evaluate the capability of STLPP when QBC is employed. Figure 4.6 shows the 3-D manifolds of four dimension reduction algorithms: LPP, PCA, ST-LE, and STLPP. For STLPP, σ is set to 16 (i.e., 2^4), and the quantization level n is set to 3. It can be seen that STLPP and ST-LE show better visual clustering effects. However, the effect of temporal information in ST-LE is too strong, shaping the generated video manifold almost into a line structure over time. For STLPP - Figure 4.6(d), similar motion patterns still cluster well even though temporal similarity is integrated. Note that the same event have different clusters, e.g., walking event (blue $\times$). By looking

through the dataset, this is because the two walking events have different direction, as depicted in Figure 4.7.

(a) (b)

Figure 4.7: Same events generate different video manifolds due to different motion directions. (a) Walking event from right to left. (b) Walking event from left to right.

To evaluate the classification performance, a training/testing split ratio is set to 1/2 for the dataset (the same as in (Thida *et al.*, 2012)). The average results from 10 randomized runs are reported.

We first evaluate the effect of temporal window size T and embedding dimensions l. Embedding dimension and temporal window size are both tuned from 1 to 10 in the experiment. The average error rate is reported in Figure 4.8. Clearly, the performance has been improved significantly when temporal window size and embedding dimension are adjusted from 1 to 3. Larger embedding dimension settings seem to further reduce the error. Nevertheless, long temporal window size (i.e., from 7 to 10) results in deterioration performance. In the following, considering both computational cost and performance, embedding dimension and temporal window size are both set to 5.

Table 4.2: Confusion matrix for event recognition in the PETS dataset ($l = 5, T = 5$).

Walking (W)	1.00	0	0	0	0	0	0
Running (R)	0.03	0.97	0	0	0	0	0
Local dispersion (LD)	0	0	0.95	0.05	0	0	0
Local movement (LM)	0	0	0.07	0.92	0	0.01	0
Splitting (S)	0.03	0	0	0	0.97	0	0
Formation (F)	0	0	0	0.02	0	0.98	0
Evacuation (E)	0	0	0	0.01	0	0	0.99
Classified as →	W	R	LD	LM	S	F	E

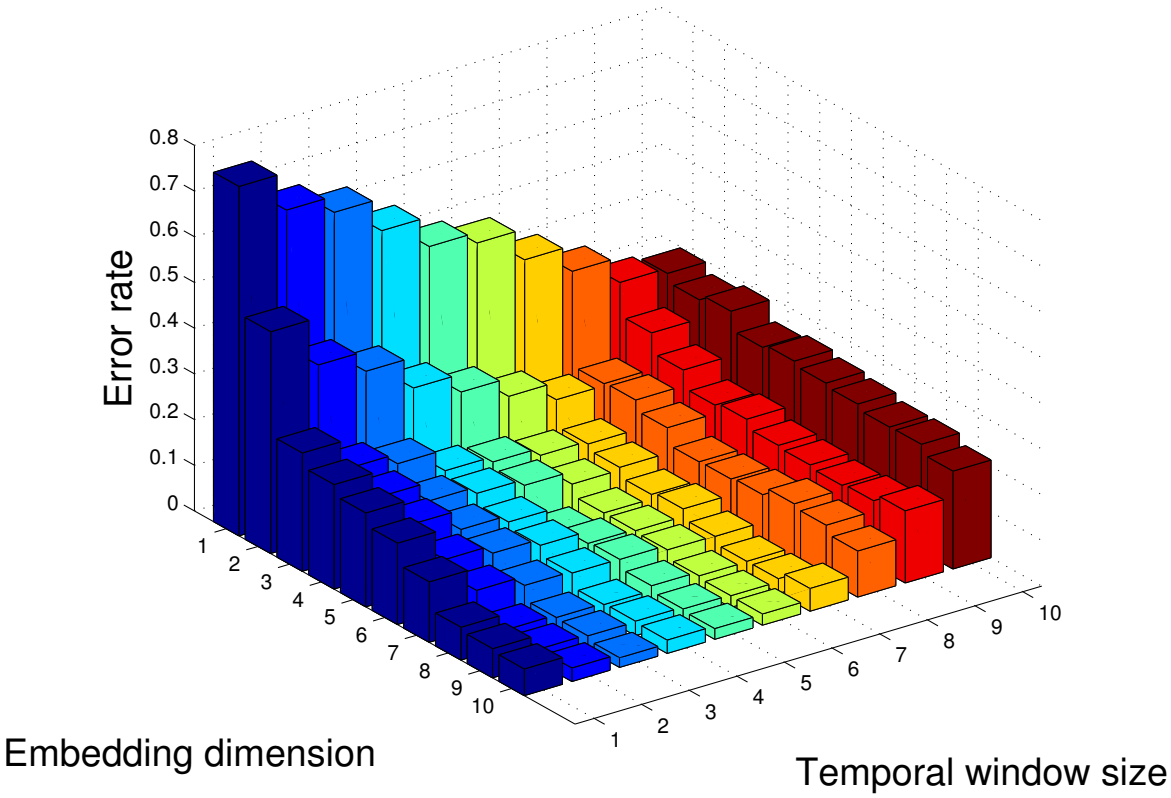


Figure 4.8: Error rate in terms of embedding dimension and temporal window size.

The confusion matrix corresponding to the highest recognition accuracy, is shown in Table 4.2.

Table 4.3: Comparison of event recognition accuracy on the PETS dataset.

Method	Accuracy (%)
HOG Tracker (Garate *et al.*, 2009)	82.83
Dynamic Textures (Chan *et al.*, 2009)	84.33
ST-LE (Thida *et al.*, 2012)	89.42
STVFF (Su *et al.*, 2012)	90.20
STLPP with HOF	93.70
STLPP with QBC ($n = 1$)	93.64
STLPP with QBC ($n = 2$)	96.79
STLPP with QBC ($n = 3$)	**96.85**
STLPP with QBC ($n = 4$)	96.33

Clearly most crowd events have very good recognition performance. The high similarity between Local Movement (LM) and Local Dispersion (LD) contributes to a significant confusion, which is however consistent with the manifold shown in Figure 4.6(a), where the overlap of the two event classes in the manifold is obvious. Compared with the state-of-the-art methods, including HOG Tracker (Garate *et al.*, 2009), Dynamic Texture (Chan *et al.*, 2009), ST-LE (Thida *et al.*, 2012), and STVFF (Su *et al.*, 2012), STLPP has increased the recognition accuracy by around 7%, as shown in Table 4.3.

4.3.2 AED result

To compare with the state-of-the-art approaches, we follow the same settings as in other works: 3/4 normal trajectories are selected randomly for training and the rest (normal and abnormal) are used for testing. The results reported are averaged from 10 randomized runs. In this experiment, we split each frame (240×320) into 20 (4×5) regions. Radial Basis Function (RBF) kernel (Chang and Lin, 2011) is adopted to measure the similarity between two trajectories.

Similar to previous experiments, we first fix the embedding dimension and temporal window size (both to 3) and compute the best detection accuracy while increasing σ exponentially. The corresponding results are shown in Figure 4.9. The σ with the best detection accuracy in three scenes are 2, 2^{-2} and 2^{-2} respectively. Their corresponding 3-dimension manifolds demonstrate that our proposed method separate normal and abnormal events clearly in the low-dimensional embedding space.

Table 4.4: Comparison of average AUC on the UMN dataset.

Method	**AUC**
Chaotic Dynamics (Wu *et al.*, 2010)	0.99
Social Force (Mehran *et al.*, 2009)	0.96
SRC (Cong *et al.*, 2011)	0.99
ST-LE (Thida *et al.*, 2012)	0.97
Phase Correlation (Shi *et al.*, 2010)	0.89
STLPP	**0.99**

Eventually, we compare our method using QBC and OCSVM with six other state-of-the-art methods in Table 5.1, where the AUC values are reported. The performance of our method is competitive, being the same as (Wu *et al.*, 2010; Cong *et al.*, 2011), and outperforming others (Mehran *et al.*, 2009; Thida *et al.*, 2012; Shi *et al.*, 2010).

4.4 Summary

In this chapter, we have proposed a novel method for crowd scene analysis in video surveillance through video manifold learning. Based on using regional optical flow histograms, we adopt the adaptive quantization and binarization coding scheme which gives better event discriminant ability as revealed by increased information gain and class-dependent redundancy indices. Furthermore, the improved feature codes are embedded in a low-dimensional space using a manifold projection algorithm that integrates both spatial and temporal similarities among frames. The proposed approach is able to generate manifolds with well-shaped motion pattern clusters. Experimental results obtained from two benchmark sets have verified that our proposed method gives competitive performance for CER and AED.

Despite the competitive results and the linear property, the proposed framework still works in batch-mode, hence cannot satisfy the demand of applications where data are received incrementally from online video streams (Lu and He, 2005), which is much more important for video surveillance application. To this end, we propose an online adaptive OCSVM to detect abnormal events in the next chapter.

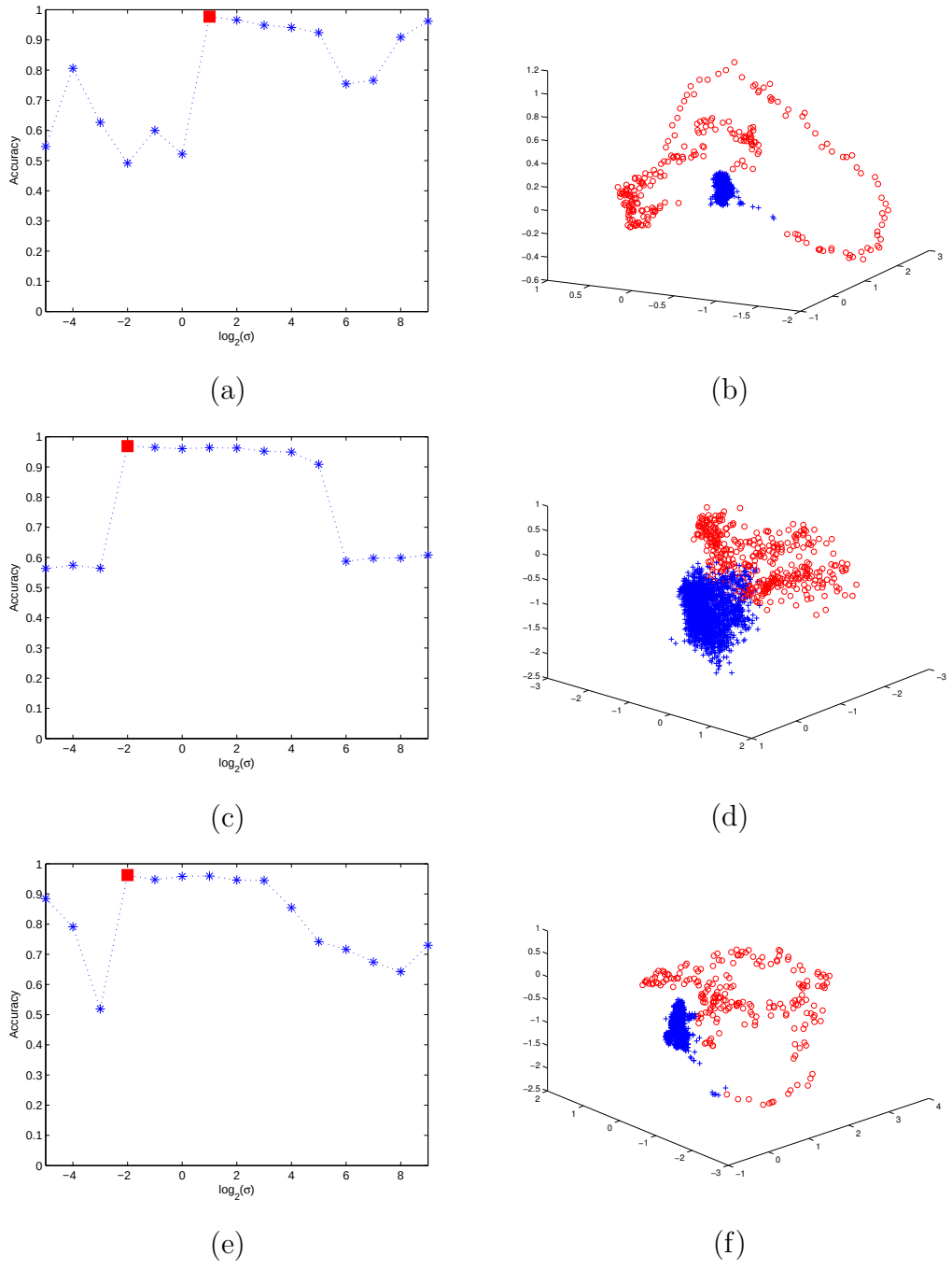


Figure 4.9: STLPP manifolds for the UMN scenes (shown on the right) generated from optimal σ values (indicated by red square markers shown on the left).(a)(b) Scene 1; (c)(d) Scene 2; (e)(f) Scene 3. In (b)(d)(f), blue + indicates normal frames, and red $\circ$ indicates abnomal frames.

Chapter 5

Online Adaptive OCSVM for AED

5.1 Overview

Anomalies are patterns in data that do not conform to a well-defined notion of normal behaviour (Chandola *et al.*, 2009). Correspondingly, anomaly detection refers to identifying those anomalous patterns given the normal patterns. Recently, in the context of increased awareness of national security and public safety, Abnormal Event Detection (AED), i.e., anomaly detection in crowd scenes, has attracted more and more attention for the development of intelligent video surveillance systems.

Most existing machine learning algorithms for anomaly detection (Breunig *et al.*, 2000; Roth, 2006) operate by nature in batch mode, which cannot be directly applied to AED due to the large volume and the stream fashion of the video data. To this end, we propose an online adaptive One-Class Support Vector Machines (OCSVM) framework to detect anomalies in crowd scenes, inspired by the work of (Cauwenberghs and Poggio, 2001; Laskov *et al.*, 2006). To the best of our knowledge, our work is the first attempt to apply incremental and decremental framework to AED. The main contributions are as follows:

- We extend the incremental and decremental framework from binary classification scenario to one-class scenario. By solving a linear system and recursively constructing new exact solutions to keep the Karush-Kuhn-Tucker conditions (Boyd and Vandenberghe, 2004) satisfied for the updated data, this approach is capable of processing video streams in an online fashion with a reasonable degree of efficiency.

- Using a sliding buffer technique to keep the recent data, our framework is able

to handle concept drift.

- We provide a unified framework to detect both global and local anomalies.

5.2 Introduction to OCSVM

During the last few years, OCSVM has been widely applied to a number of studies, including signature classification (Guerbai *et al.*, 2014), anomaly detection (Manevitz and Yousef, 2002; Ma and Perkins, 2003), and clustering (Ben-Hur *et al.*, 2002) etc.

Here we briefly outline the OCSVM and introduce the notations to be used later for presenting our framework. Given a set of unlabeled training data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, OCSVM aims to find an optimal hyperplane that separates the data points from the origin in the feature space:

$$f(\mathbf{x}) = \mathbf{w} \cdot \Phi(\mathbf{x}) - \rho, \tag{5.1}$$

where Φ is a feature map that transforms $\mathbf{x}$ from the input space $\mathcal{X}$ to the feature space $\mathcal{F}$.

To acquire the optimal parameters $\mathbf{w}$ and ρ, one can solve the following quadratic programming problem:

$$\begin{aligned} &\min_{\mathbf{w},\xi,\rho} : \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i - \rho \\ &s.t. : \mathbf{w} \cdot \Phi(\mathbf{x}_i) \geq \rho - \xi_i,\ \xi_i \geq 0, \end{aligned}$$

where ξ_i are slack variables, and C is a regularization parameter, controlling the trade-off between structure and empirical risks.

By introducing Lagrange multipliers $\alpha_i, \beta_i \geq 0$, the corresponding Lagrangian is formulated as:

$$L = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \xi_i - \rho - \sum_i \alpha_i(\mathbf{w} \cdot \Phi(\mathbf{x}_i) - \rho + \xi_i) - \sum_i \beta_i \xi_i. \tag{5.2}$$

To minimize L, the derivatives of Eq. (5.2) with respect to the variables $\mathbf{w}$, ξ, and ρ are set to zero, giving

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} &= \sum_i \alpha_i \Phi(\mathbf{x}_i), \\ \frac{\partial L}{\partial \xi} = 0 \Rightarrow \alpha_i &= C - \beta_i, \\ \frac{\partial L}{\partial \rho} = 0 \Rightarrow \sum_i \alpha_i &= 1. \end{aligned} \tag{5.3}$$

Substituting the above three equations into Eq. (5.2), we solve the dual problem instead:

$$\begin{aligned} &\min_{\alpha} : \frac{1}{2}\sum_{ij}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) \\ &s.t. : 0 \le \alpha_i \le C,\ \sum_i \alpha_i = 1, \end{aligned} \tag{5.4}$$

where $k(\mathbf{x}_i,\mathbf{x}_j) = \Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$ is a kernel function that measures the similarity between the two examples $\mathbf{x}_i$ and $\mathbf{x}_j$, with a maximum of 1 and a minimum of 0.

The utilization of the kernel function $k(\mathbf{x}_i,\mathbf{x}_j)$ means that the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ can be measured without exploiting the feature map Φ and the high-dimensional space explicitly. Combining Eq. (5.3) and Eq. (5.1), the separating function is rewritten as:

$$f(\mathbf{x}) = \sum_j \alpha_j k(\mathbf{x},\mathbf{x}_j) - \rho. \tag{5.5}$$

The success of OCSVM stems from the kernel method as well as margin maximization properties, both inherited from SVM. However, the conventional OCSVM model is limited to batch mode, resulting in two difficulties for data stream based applications. Firstly, the computational complexity of standard batch mode numeric techniques increases quadratically with the size of the training data, which is unsuitable to video data stream as the training set is usually very large (Thompson *et al.*, 2013). Secondly, although some more efficient approaches, e.g., (Platt, 1999; Keerthi *et al.*, 2001; Sonnenburg *et al.*, 2006), have been proposed to tackle SVM training for large-scale data, the streaming nature of the data, i.e., video frames arriving sequentially rather than all at once, demands an OCSVM classifier that can learn in an online mode.

To address this limitation, researchers have proposed a series of online OCSVM algorithms (Kivinen *et al.*, 2004; Desobry *et al.*, 2005; Laskov *et al.*, 2006; Gómez-Verdejo *et al.*, 2011; Wang, 2013; Kolev *et al.*, 2015). A straightforward method is to apply the batch algorithm by using a sliding window (Desobry *et al.*, 2005), but this incurs a high computational cost for each batch computation. Kivinen *et al.* (2004) adopt a stochastic gradient descent technique to develop a computationally efficient algorithm while giving only approximate results. An analogous adaptive OCSVM is proposed in Gómez-Verdejo *et al.* (2011) based on the solution of iterated re-weighted least squares, where a forgetting factor parameter reflects the time impact on data. To detect abnormal event in surveillance, Wang (2013) proposes both online Least Squares One-Class Support Vector Machines (LS-OCSVM) and sparse online LS-OCSVM to model covariance descriptors of frame behavior. Based on the work of incremental

and decremental SVM (Cauwenberghs and Poggio, 2001), Laskov *et al.* (2006) propose an incremental OCSVM for online learning. Following the same idea of Laskov *et al.* (2006), Kolev *et al.* (2015) apply the incremental OCSVM to flight data analysis by considering both false positive and false negative separately. The incremental OCSVM, however, poses some weaknesses as follows. Firstly, its computational cost rockets up as the dataset grows, which is unsuitable for real-time demanding system. Secondly, it lacks a decremental procedure to abandon obsolete data. While analogous to the incremental OCSVM, our online adaptive OCSVM overcomes the above drawbacks, where the incremental and decremental scheme as well as the sliding buffer technique cut down the processing time while handling concept drift (Gama *et al.*, 2014) or slow distribution changes effectively.

5.3 The computational framework

The flowchart of our proposed computational framework is illustrated in Figure 5.1. Given a training set of video segments, each segment is divided into a set of video events. Non-overlapping spatial-temporal patches are densely sampled from each event. For each small patch, a Histogram of Optical Flow (HOF) descriptor vector is computed. We then form a visual vocabulary by performing k-means clustering of a random subset of descriptors extract from the training set. By assigning each descriptor to its closest vocabulary word, the video events are represented as histograms and used to train an OCSVM classifier. In the testing process, when a new test event arrives, its histogram representation is verified by the learnt OCSVM model. If the test event is not detected as an anomaly but satisfies the update criterion, the OCSVM model will be updated for further detection.

5.3.1 Event representation and feature extraction

The proposed procedure of event representation and feature extraction is a Bag of Words (BoW) approach, which has recently demonstrated impressive performance on document classification (Hofmann, 2001; Blei *et al.*, 2003), image categorization (Csurka *et al.*, 2004; Fei-Fei and Perona, 2005; Sivic *et al.*, 2005; Moosmann *et al.*, 2008; van Gemert *et al.*, 2010; Van De Sande *et al.*, 2010) and action recognition (Klaser *et al.*, 2008; Niebles *et al.*, 2008; Willems *et al.*, 2008; Wang *et al.*, 2009). More specifically, a document, image or video clip is represented as the bag of its words, local

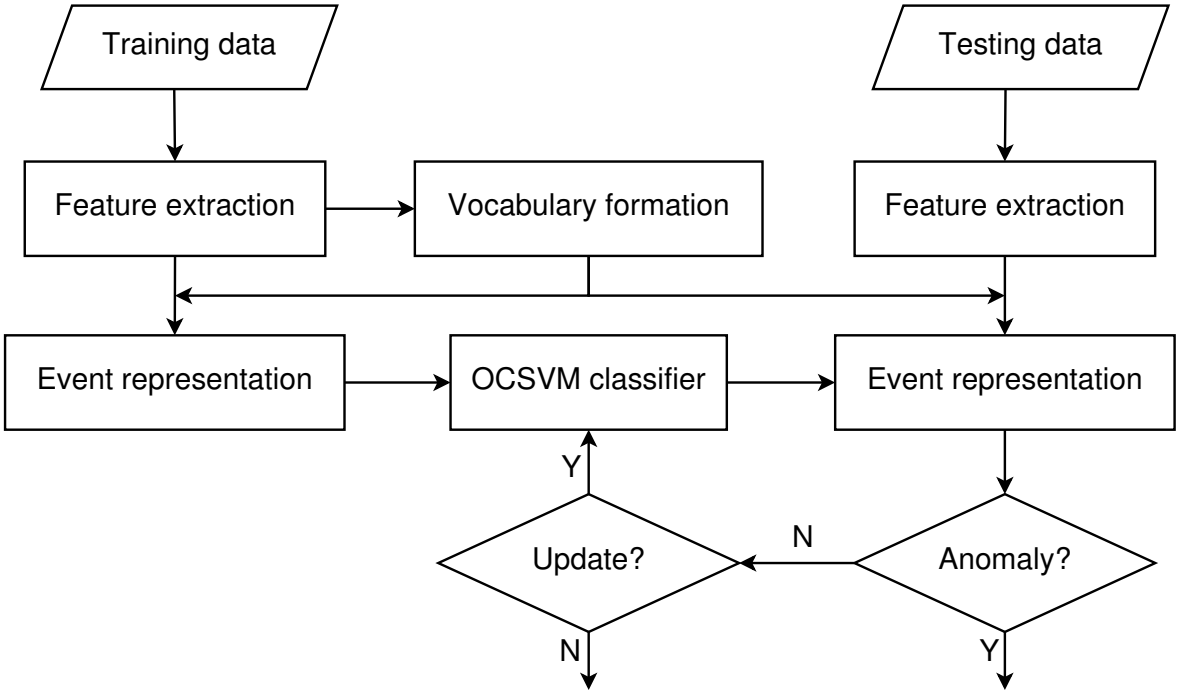


Figure 5.1: The flowchart of online adaptive OCSVM for AED.

image features or local video features, discarding any information about their position and structure.

However, due to the fact that locations and structure information are of vital importance in image categorization and action recognition, some variant BoW approaches (Lazebnik *et al.*, 2006; Yang *et al.*, 2009; Choi *et al.*, 2013) are proposed to incorporate the position information.

Event representation

Following Cong *et al.* (2011), we classify the anomaly in crowd scenes into two classes on the basis of anomaly scale: Global Anomaly (GA) and Local Anomaly (LA). The GA is defined for anomaly on the whole-scene level, while the LA refers to an individual's behaviour is different from that of historical and neighbouring individuals. To deal with the two anomaly categories, we propose two separate representations.

As shown in Figure 5.2, video segments are firstly obtained using a sliding window. We then extract the non-overlapping spatial-temporal patches and compute their HOF descriptors to form a codebook. These video segments are divided into spatial-temporal events or temporal events. Specifically, the video segment is partitioned into $2^l \times 2^l$ cells (l is called pyramid level, $l = 0, 1, 2, 3$). In each cell we compute its histogram separately, and the spatial-temporal event is represented as a concatenated histogram.

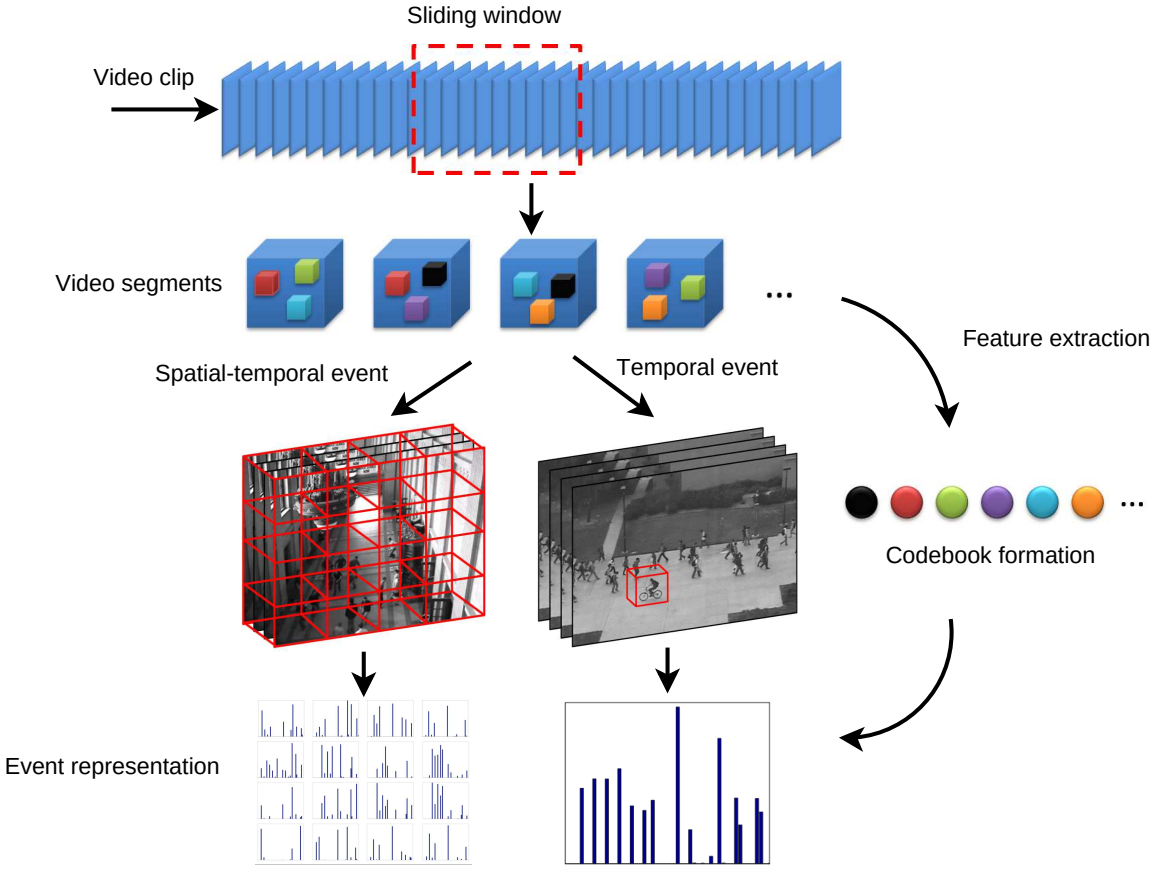


Figure 5.2: The framework of event representation.

The approach of spatial-temporal event representation, called Spatial-Temporal Pyramid Matching (STPM) (Lin *et al.*, 2014), has been proved effective for crowd event recognition. As for LA, it usually happens in a rather compact region, so we ignore the position information and only one histogram is computed.

Histogram of Optical Flow

Using a sliding window to get the video segments, we then compute optical flow using Horn and Schunck's method (Horn and Schunck, 1981). The flow vector containing two directional components is denoted as $\{f_x, f_y\}$. The orientation α and magnitude f of the flow vectors are calculated as:

$$\begin{aligned} \alpha &= \arctan(\frac{f_y}{f_x}), \\ f &= \sqrt{f_x^2 + f_y^2}, \end{aligned} \tag{5.6}$$

where the range of α is $(-\frac{\pi}{2}, \frac{\pi}{2})$, and thus mapped to $[0, 2\pi)$.

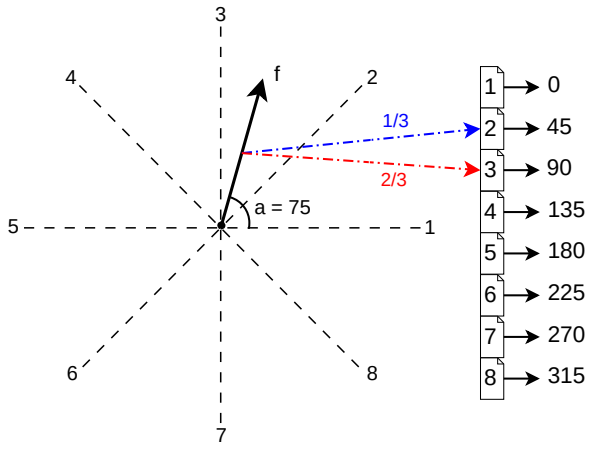


Figure 5.3: Illustration of interpolating a flow vector into neighboring bins ($N = 8$).

Next, in each spatial-temporal patch, all flow vectors are quantized into N bins by using a soft-assignment scheme (van Gemert *et al.*, 2010). Figure 5.3 shows an example with $N = 8$ bins. Each bin covers $360/8 = 45$ degrees, and the dotted lines correspond to bin centres. For a flow vector with orientation $\alpha = 75°$, its deviation from Bin 2 centre (i.e., 45°) and Bin 3 centre (i.e., 90°) are 30° and 15° respectively. Thus, the weights assigned to Bin 2 and Bin 3 are 1/3 (i.e. 15/45) and 2/3 (i.e. 30/45) respectively. Likewise, each flow vector is assigned to two neighbouring bins with relevant, non-zero weights unless they align exactly with one of the dotted lines (in which case the weight to the corresponding bin will be 1, and all others 0). A N-dimensional HOF descriptor is then formed counting all flow vectors' soft contribution. Obviously, our approach is not restricted to the simple descriptor, which can be easily replaced by some more complicated spatial-temporal feature descriptors, e.g., 3D HOG (Klaser *et al.*, 2008), or 3D SIFT (Scovanner *et al.*, 2007).

5.3.2 Online adaptive OCSVM

Figure 5.4 illustrates the procedure of online adaptive OCSVM algorithm. From time t to $t + 1$, we add the new arriving data (red circle) to the sliding buffer using the incremental procedure, while removing the obsolete data (blue circle) from the sliding buffer through the decremental procedure. The incremental procedure of the algorithm, inspired by Cauwenberghs and Poggio (2001), is also the same as Laskov *et al.* (2006). Nevertheless, the additional decremental procedure and sliding buffer techniques in our approach reduce the computational cost significantly and discard obsolete data patterns that no longer reflect the distribution of normal patterns.

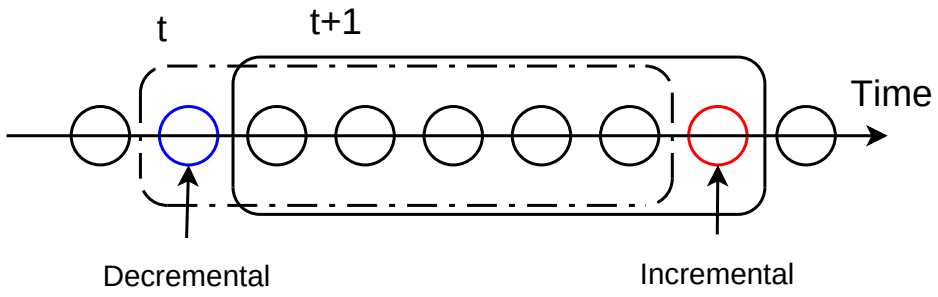


Figure 5.4: The framework of online adaptive OCSVM.

Karush-Kuhn-Tucker conditions

To elaborate our approach, we rewrite the dual problem Eq. (5.4) as a saddle-point formulation:

$$\max_{\rho} \min_{0 \le \alpha_i \le C} : W = \frac{1}{2} \sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \rho(\sum_i \alpha_i - 1).$$

The first-order conditions on W reduce to the Karush-Kuhn-Tucker (KKT) conditions:

$$g_i = \frac{\partial W}{\partial \alpha_i} = \sum_j \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \rho$$

$$\Longrightarrow f(\mathbf{x}_i) \begin{cases} \geq 0, & \text{if } \alpha_i = 0, \\ = 0, & \text{if } 0 < \alpha_i < C, \\ \leq 0, & \text{if } \alpha_i = C, \end{cases} \tag{5.7}$$

$$\frac{\partial W}{\partial \rho} = \sum_i \alpha_i - 1 = 0. \tag{5.8}$$

On the basis of Eq. (5.7), the indexes of training set $I_{\mathcal{X}}$ are defined as three sets:

$$\begin{aligned} I_{\mathcal{X}} &= I_{\mathcal{S}} \cup I_{\mathcal{E}} \cup I_{\mathcal{O}}, \\ I_{\mathcal{S}} &= \{i : \mathbf{x}_i \in \mathcal{X}, 0 < \alpha_i < C\}, \\ I_{\mathcal{E}} &= \{i : \mathbf{x}_i \in \mathcal{X}, \alpha_i = C\}, \\ I_{\mathcal{O}} &= \{i : \mathbf{x}_i \in \mathcal{X}, \alpha_i = 0\}. \end{aligned}$$

In parallel, the training data $\mathcal{X}$ is divided into three subsets:

$$\begin{aligned} \mathcal{X} &= \mathcal{S} \cup \mathcal{E} \cup \mathcal{O}, \\ \mathcal{S} &= \{\mathbf{x}_i : \forall i \in I_{\mathcal{S}}\}, \\ \mathcal{E} &= \{\mathbf{x}_i : \forall i \in I_{\mathcal{E}}\}, \\ \mathcal{O} &= \{\mathbf{x}_i : \forall i \in I_{\mathcal{O}}\}, \end{aligned}$$

where $\mathcal{S}$, $\mathcal{E}$, and $\mathcal{O}$ are referred to margin support vector set, error support vector set, and the remaining set. Figure 5.5 illustrates the three subsets in OCSVM model. From now on, we will abbreviate $k(\mathbf{x}_i, \mathbf{x}_j)$ to k_{ij}. For any two subsets denoted as $\mathcal{X}$ and $\mathcal{Y}$, $k_{\mathcal{XY}}$ is the kernel matrix whose rows are indexed by $\mathcal{X}$, and the columns are indexed by $\mathcal{Y}$.

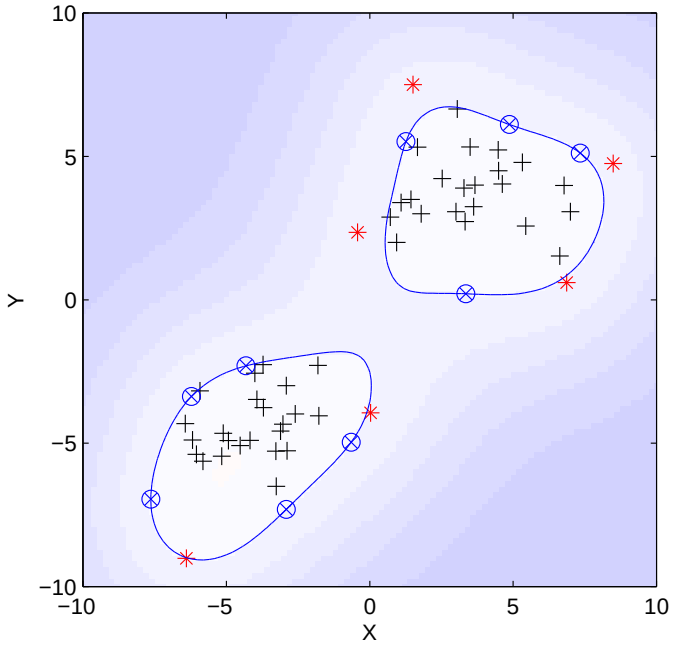


Figure 5.5: In OCSVM, set $\mathcal{S}$ corresponds to data on the boundary (blue $\otimes$), set $\mathcal{E}$ corresponds to the data outside boundary (red $*$), and set $\mathcal{O}$ is the data in the boundary (black $+$).

Derivation

To add or remove a data $\mathbf{x}_c$ to/from the trained OCSVM model, if we have $g_c > 0$, we add or remove $\mathbf{x}_c$ straightforwardly and wait for processing the new incoming data because it has no impact on the model. If we have $g_c \leq 0$, we update the coefficients of margin support vectors $\mathcal{S}$ and ρ to keep the KKT conditions satisfied for the enlarged or reduced data set:

$$\Delta g_i = k_{ic}\Delta\alpha_c + \sum_{j \in I_{\mathcal{S}}} k_{ij}\Delta\alpha_j + \Delta\rho, \quad \forall i \in I_{\mathcal{X}} \cup c, \tag{5.9}$$

$$0 = \Delta\alpha_c + \sum_{j \in I_{\mathcal{S}}} \Delta\alpha_j. \tag{5.10}$$

For all margin support vectors set $\mathcal{S}$, $g_i \equiv 0, \forall i \in I_{\mathcal{S}}$. Eq. (5.9) and (5.10) can be rewritten as matrix notation:

$$\underbrace{\begin{bmatrix} 0 & 1 \\ 1 & k_{\mathcal{S}\mathcal{S}} \end{bmatrix}}_{K} \begin{bmatrix} \Delta\rho \\ \Delta\alpha_{\mathcal{S}} \end{bmatrix} = - \begin{bmatrix} 1 \\ k_{\mathcal{S}c} \end{bmatrix} \Delta\alpha_c,$$

$$\Longrightarrow \begin{bmatrix} \Delta\rho \\ \Delta\alpha_{\mathcal{S}} \end{bmatrix} = \begin{bmatrix} \beta \\ \beta_{\mathcal{S}} \end{bmatrix} \Delta\alpha_c, \tag{5.11}$$

with coefficient sensitivities given by

$$\begin{bmatrix} \beta \\ \beta_{\mathcal{S}} \end{bmatrix} = -Q \begin{bmatrix} 1 \\ k_{\mathcal{S}c} \end{bmatrix}, \tag{5.12}$$

where $Q = K^{-1}$. Note that we assume the kernel matrix K is nonsingular, namely invertible.

Substitute Eq. (5.11) in Eq. (5.9):

$$\Delta g_i = \gamma_i \Delta\alpha_c, \tag{5.13}$$

with margin sensitivities:

$$\gamma_i = k_{ic} + k_{i\mathcal{S}}\beta_{\mathcal{S}} + \beta, \ \forall i \notin I_{\mathcal{S}}. \tag{5.14}$$

Incremental learning

Nevertheless, we cannot obtain the new OCSVM state directly as in Eq. (5.11) and (5.13) the composition of the sets $\mathcal{S}$, $\mathcal{E}$ and $\mathcal{O}$ changes relative to the change of $\Delta\alpha_c$ and Δg_i.

In the incremental learning procedure, we have identified the following five conditions that are most likely to occur (Cauwenberghs and Poggio, 2001; Laskov *et al.*, 2006):

1. g_c becomes zero, namely $\mathbf{x}_c$ joins to $\mathcal{S}$. The corresponding increment is computed as $\Delta\alpha_c^g = \frac{-g_c}{\gamma_c}$.

2. α_c reaches C, corresponding to $\mathbf{x}_c$ becomes an error support vector. The equivalent incremental step is computed as $\Delta\alpha_c^\alpha = C - \alpha_c$.

3. One g_i in $\mathcal{E}$ becomes zero, equivalent to $\mathbf{x}_i$ transferring from $\mathcal{E}$ to $\mathcal{S}$. The most likely occurred constraint $\Delta\alpha_c^{\mathcal{E}}$ equals to finding the minimal increment:

$$\Delta\alpha_c^{\mathcal{E}} = \min \frac{-g_i}{\gamma_i}, \ \forall i \in I_{\mathcal{E}} \cap \gamma_i > 0.$$

4. One g_i in $\mathcal{O}$ becomes zero, equivalent to $\mathbf{x}_i$ transferring from $\mathcal{O}$ to $\mathcal{S}$. The most likely occurred step is computed as:

$$\Delta\alpha_c^{\mathcal{O}} = \min \frac{-g_i}{\gamma_i},\ \forall i \in I_{\mathcal{O}} \cap \gamma_i < 0.$$

5. $\mathbf{x}_i$ in $\mathcal{S}$ reaches a bound, α_i with equality 0 is equivalent to transferring $\mathbf{x}_i$ from $\mathcal{S}$ to $\mathcal{O}$, and equality C from $\mathcal{S}$ to $\mathcal{E}$. The most likely increment equals:

$$\Delta\alpha_c^{\mathcal{S}} = \min \frac{\Delta\alpha_i^{\mathcal{S}}}{\beta_i},\ \forall i \in I_{\mathcal{S}},$$

where

$$\Delta\alpha_i^{\mathcal{S}} = \begin{cases} C - \alpha_i, & \text{if } \beta_i > 0; \\ -\alpha_i, & \text{if } \beta_i < 0. \end{cases}$$

Finally, $\Delta\alpha_c$ is determined by finding the minimal value of the above conditions:

$$\Delta\alpha_c = \min(\Delta\alpha_c^g, \Delta\alpha_c^\alpha, \Delta\alpha_c^{\mathcal{E}}, \Delta\alpha_c^{\mathcal{O}}, \Delta\alpha_c^{\mathcal{S}}). \tag{5.15}$$

Once obtaining $\Delta\alpha_c$, we can update ρ, α_i, g_i through Eq. (5.11) and (5.13). The process repeats until the coefficient α_c becomes C or g_c reaches zero, i.e., the enlarged data set satisfies the KKT condition. Algorithm 1 details the incremental learning procedure.

Decremental learning

The single incremental procedure is reversible, when we remove a data $\mathbf{x}_c$ (with $g_c \leq 0$) from the trained OCSVM model, we decrease its coefficient a_c from its orginal value to 0 while checking the following four conditions that are most likely to happen:

1. α_c reaches 0, namely $\mathbf{x}_c$ transfers from $\mathcal{S}$ or $\mathcal{E}$ to 0. The equivalent decrement is computed as $\Delta\alpha_c^\alpha = -\alpha_c$.

2. One g_i in $\mathcal{E}$ becomes zero, equivalent to $\mathbf{x}_i$ transferring from $\mathcal{E}$ to $\mathcal{S}$. The most likely occurred decrease $\Delta\alpha_c^{\mathcal{E}}$ is computed as:

$$\Delta\alpha_c^{\mathcal{E}} = \max \frac{-g_i}{\gamma_i},\ \forall i \in I_{\mathcal{E}} \cap \gamma_i < 0.$$

3. One g_i in $\mathcal{O}$ becomes zero, equivalent to $\mathbf{x}_i$ transferring from $\mathcal{O}$ to $\mathcal{S}$. The corresponding decrement is computed as:

$$\Delta\alpha_c^{\mathcal{O}} = \max \frac{-g_i}{\gamma_i},\ \forall i \in I_{\mathcal{O}} \cap \gamma_i > 0.$$

Algorithm 1 Incremental learning

Input: $\mathcal{S}$, $\mathcal{E}$, $\mathcal{O}$, $\{\alpha_i\}_{i=1}^{n}$, $\{g_i\}_{i=1}^{n}$, ρ, Q, $\mathbf{x}_c$

Output: $\mathcal{S}$, $\mathcal{E}$, $\mathcal{O}$, $\{\alpha_i\}_{i=1}^{n+1}$, $\{g_i\}_{i=1}^{n+1}$, ρ, Q

```
Initialize α_c ← 0.
Compute g_c using Eq. (5.7).
if g_c > 0 then
    O ← O ∪ x_c
    return
else
    while true do
        Compute β, β_S and γ_i in terms of Eq. (5.12) and Eq. (5.14).
        Compute Δα_c according to Eq. (5.15).
        α_c ← α_c + Δα_c
        α_S ← α_S + β_S Δα_c
        g_i ← g_i + γ_i Δα_c, ∀i ∈ I_E ∪ I_O ∪ c
        ρ ← ρ − β Δα_c
        Update S, E, O according to occurred condition.
        if S has changed then
            Update Q according to Eq. (5.19) or Eq. (5.20).
        end if
        if Δα_c = Δα_c^g then
            S ← S ∪ x_c
            return
        end if
        if Δα_c = Δα_c^α then
            E ← E ∪ x_c
            return
        end if
    end while
end if
```

Algorithm 2 Decremental learning

Input: $\mathcal{S}$, $\mathcal{E}$, $\mathcal{O}$, $\{\alpha_i\}_{i=1}^{n}$, $\{g_i\}_{i=1}^{n}$, ρ, $c \in I_{\mathcal{X}}$

Output: $\mathcal{S}$, $\mathcal{E}$, $\mathcal{O}$, $\{\alpha_i\}_{i=1}^{n-1}$, $\{g_i\}_{i=1}^{n-1}$, ρ, Q

```
if x_c ∈ O then
    O ← O \ x_c
    return
end if
if x_c ∈ S then
    S ← S \ x_c
    Update Q according to Eq. (5.19) or Eq. (5.20).
else
    E ← E \ x_c
end if
while true do
    Compute β, β_S and γ_i in terms of Eq. (5.12) and (5.14).
    Compute Δα_c according to Eq. (5.16).
    α_c ← α_c + Δα_c
    α_S ← α_S + β_S Δα_c
    g_i ← g_i + γ_i Δα_c ∀i ∈ I_E ∪ I_O
    ρ ← ρ − β Δα_c
    Update S, E, O according to occurred condition.
    if S has changed then
        Update Q according to Eq. (5.19) or Eq. (5.20).
    end if
    if Δα_c = Δα_c^α then
        return
    end if
end while
```

4. $\mathbf{x}_i$ in $\mathcal{S}$ reaches a bound, α_i with equality 0 is equivalent to transferring $\mathbf{x}_i$ from $\mathcal{S}$ to $\mathcal{O}$, and equality C from $\mathcal{S}$ to $\mathcal{E}$. The largest possible decrement is computed as:

$$\Delta\alpha_c^{\mathcal{S}} = \max \frac{\Delta\alpha_i^{\mathcal{S}}}{\beta_i}, \ \forall i \in I_{\mathcal{S}},$$

where

$$\Delta\alpha_i^{\mathcal{S}} = \begin{cases} C - \alpha_i, & \text{if } \beta_i < 0; \\ -\alpha_i, & \text{if } \beta_i > 0. \end{cases}$$

On the contrary, $\Delta\alpha_c$ is determined by finding the maximal value of the above conditions:

$$\Delta\alpha_c = \max(\Delta\alpha_c^{\alpha}, \Delta\alpha_c^{\mathcal{E}}, \Delta\alpha_c^{\mathcal{O}}, \Delta\alpha_c^{\mathcal{S}}). \tag{5.16}$$

The same as the single incremental procedure, it repeats until α_c becomes zero. Algorithm 2 details the decremental learning procedure.

Recursive update of Q

It is time-consuming if we compute the inverse matrix Q whenever the set $\mathcal{S}$ has changed. Fortunately, by applying the Sherman-Morrison-Woodbury formula (Golub and van Loan, 1996) for block matrix inversion, we can update the matrix Q in an efficient way. More specifically, let $\hat{Q}$ be the enlarged inverse matrix, when a data $\mathbf{x}_i$ transfers to $\mathcal{S}$, the inversed kernel matrix expands to:

$$\hat{Q} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & k_{\mathcal{SS}} & k_{i\mathcal{S}}^T \\ 1 & k_{i\mathcal{S}} & k_{ii} \end{bmatrix}^{-1} = \begin{bmatrix} K & V^T \\ V & k_{ii} \end{bmatrix}^{-1}, \tag{5.17}$$

where

$$V = \begin{bmatrix} 1 & k_{i\mathcal{S}} \end{bmatrix}.$$

Using Sherman-Morrison-Woodbury formula, the expanded inversed kernel matrix is rewritten as:

$$\begin{bmatrix} K & V^T \\ V & k_{ii} \end{bmatrix}^{-1} = \begin{bmatrix} K^{-1} & 0 \\ 0 & 0 \end{bmatrix} + (k_{ii} - VK^{-1}V^T)^{-1} \begin{bmatrix} -K^{-1}V^T \\ 1 \end{bmatrix} \begin{bmatrix} -VK^{-1} & 1 \end{bmatrix}. \tag{5.18}$$

To sum up, the update rule from Q to $\hat{Q}$ is computed as:

$$\hat{Q} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} + (1 - VQV^T)^{-1} \begin{bmatrix} -QV^T \\ 1 \end{bmatrix} \begin{bmatrix} -VQ & 1 \end{bmatrix}, \quad i \notin I_{\mathcal{S}}. \tag{5.19}$$

It is analogous to remove a data $\mathbf{x}_i$ from the set $\mathcal{S}$. To distinguish from the shrinking inverse matrix Q, let $\hat{Q}$ be the original inverse matrix. Combining with Eq. (5.18), we can write:

$$\hat{Q} = \begin{bmatrix} \hat{Q}_{\mathcal{S}\mathcal{S}} & \hat{Q}_{i\mathcal{S}}^T \\ \hat{Q}_{i\mathcal{S}} & \hat{Q}_{ii} \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{\zeta} \begin{bmatrix} QV^TVQ & -QV^T \\ -VQ & 1 \end{bmatrix},$$

where $\zeta = 1 - VQV^T$.

After matrix manipulation, the update rule is written as:

$$Q = \hat{Q}_{\mathcal{S}\mathcal{S}} - \hat{Q}_{ii}^{-1}\hat{Q}_{i\mathcal{S}}^T\hat{Q}_{i\mathcal{S}}, \quad i \notin I_{\mathcal{S}}. \tag{5.20}$$

5.4 Experimental result

We have conducted experiments on three benchmark datasets: 'UMN', 'UCSD Ped2', and 'Subway Exit'. The UMN dataset is used to verify the effectiveness of spatial-temporal events for GA, while the UCSD Ped2 and Subway Exit datasets are used to test temporal events for LA. In the following presentation, unless otherwise specified, Radial Basis Function (RBF) kernel, i.e., $k(x_i, x_j) = \exp(-\frac{||x_i - x_j||^2}{\sigma})$ are adopted in the experiments, and $50,000$ HOF descriptors are selected randomly from the training set to form a codebook through the k-means algorithm. All experiments are repeated in 10 runs by selecting training and testing sets randomly, and the average performance is reported.

The size of events and spatial-temporal patch is chosen considering both efficiency and effectiveness. If the size of spatial-temporal patch is too small, more subtle motion information will be detected while increasing computational cost, whereas enlarging the size reduces the computation but may ignore some important motion information. Likewise, the width and height for temporal event for LA correspond to the scale of the scene, so that each event does not contain too many objects which may interfere with each other. Furthermore, the selection of temporal size provides the trade-off between the ability to detect anomaly and timely response.

5.4.1 UMN dataset

In the dataset, the size of spatial-temporal patch is set as $10 \times 10 \times 3$ and the temporal window size is set as to 12 with no overlap. The parameters σ and C in OCSVM are set to 0.005 and 0.1 empirically. The update threshold θ is set as -0.05, which means for

a new incoming data $\mathbf{x}_c$, if $g_c \geq \theta$, the OCSVM model will be updated. The selection of θ reflects what degree of slow, tiny change will be tracked.

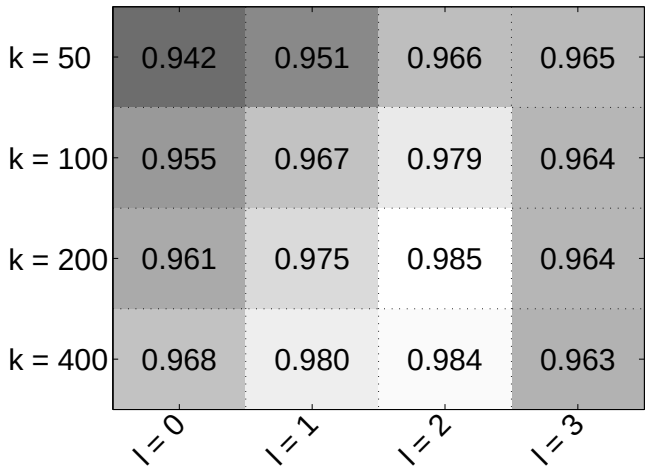


Figure 5.6: The influence of parameter settings of k and l on AUC performance of UMN dataset.

In the training process, we use the first 300 frames from each scene to train the initial parameters using a batch OCSVM algorithm (e.g. sequential minimal optimization (Platt, 1999)), and the rest of the frames for testing. To demonstrate the effectiveness of event representation for GA, we first tune the size of codebook k from 50 to 400 while adjusting pyramid level l from 0 to 3, where the cell sizes are correspondingly split from 1×1 (i.e., $2^0 \times 2^0$) to 8×8 (i.e., $2^3 \times 2^3$). Figure 5.6 displays how the parameter settings impact on the Area Under the Curve (AUC) performance. It shows that the performance rises by more than 2% when l increases from 0 to 2, but deteriorates when l reaches 3. We assume this is because the video segments is partitioned so finely that the GA representation is smoothed. On the other hand, increasing codebook size improves performance by less than 2% at lower pyramid levels (e.g, $l = 0, 1, 2$). The improvement, however, is eliminated when the codebook size increases from 200 to 400. Based on the experiment, all the following experimental results are reported with k of 200 and l of 2.

The quantitative results for Scene 1, Scene 2, and Scene 3 are shown in Figure 5.7, 5.8, and 5.9 respectively, where the top row in each scene illustrates sample frames in the dataset, and the green dots and red crosses in the bottom rows represent the normal video events and abnormal video events respectively. It shows that most abnormal events (red crosses) have lower decision values than normal events (green dots), which

is consistent with the ground truth.

Method	AUC
Optical flow (Mehran *et al.*, 2009)	0.840
Social force (Mehran *et al.*, 2009)	0.960
Chaotic invariant (Wu *et al.*, 2010)	0.994
SRC (Cong *et al.*, 2011)	0.996
H-MDT-CRF (Li *et al.*, 2014)	0.995
ST-LE (Thida *et al.*, 2012)	0.970
Phase Correlation (Shi *et al.*, 2010)	0.892
Online adaptive OCSVM	**0.985**

Table 5.1: Comparison of AUC on the UMN dataset.

We also compared our method with the state-of-the-art methods, as shown in Table 5.1. The overall Area Under the Curve (AUC) performance of our approach in three scenes are 0.9947, 0.9827, and 0.9856 (0.9853 overall), which is comparable to Chaotic invariant (Wu *et al.*, 2010), H-MDT-CRF (Li *et al.*, 2014) and SRC (Cong *et al.*, 2011), better than other methods.

5.4.2 UCSD Ped2 dataset

In the Ped2 dataset, we empirical apply a $40 \times 40 \times 15$ temporal event with $10 \times 10 \times 7$ overlap in the training set and $20 \times 20 \times 7$ overlap in the testing set. The parameters σ, C, and θ are given as 0.3, 0.1, and 0.001 through cross validation.

We use the conventional batch OCSVM on the events extracted from the first training clip (i.e., 120 frames) to obtain the initial parameters. Our online adaptive OCSVM approach is then applied to the rest of 15 training clips and 12 testing clips. Note that the update threshold θ is not triggered in the process of training so that our approach is able to learn the distribution of all the normal patterns. As in (Li *et al.*, 2014), two measurements are used to evaluate the performance of anomaly detection: frame-level and pixel-level. The former predicts which frames contain an anomaly regardless of the anomaly location within the frame, while the latter labels the event as an anomaly if at least 40% of the truly anomalous pixels are detected.

We report the ROC curve of the UCSD Ped2 dataset in Figure 5.10. Figure 5.11 shows some detected examples from UCSD Ped2 dataset. Note that the false alarm (red rectangle) that is detected by our approach in Figure 5.11(d). It is labelled as

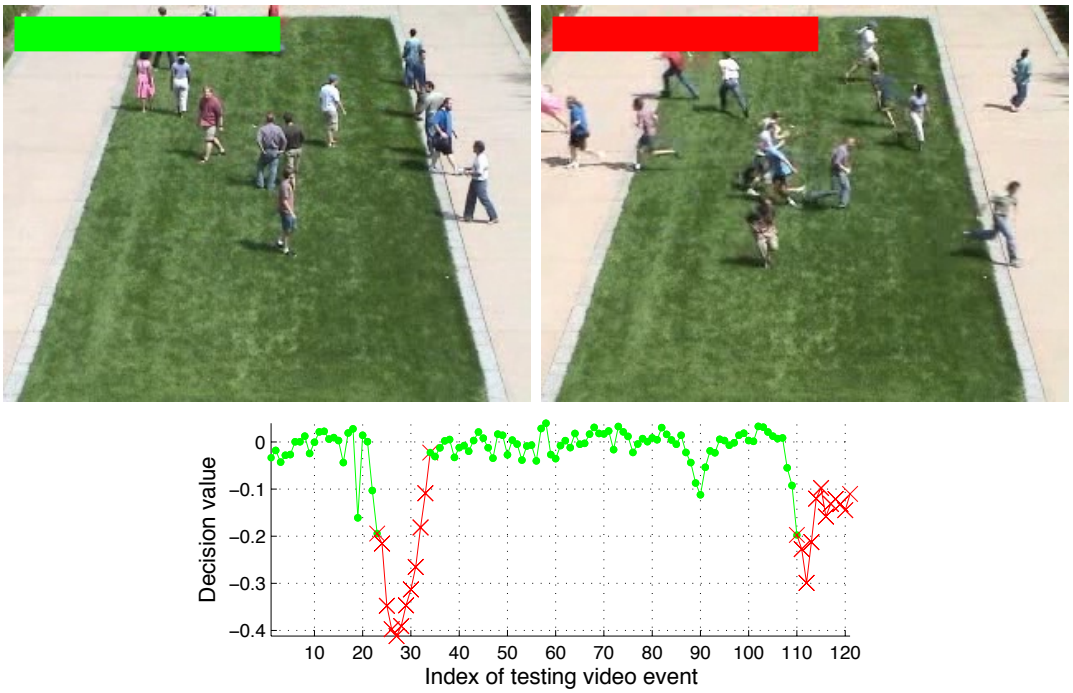


Figure 5.7: Experimental results of the online adaptive OCSVMs model on UMN dataset Scene 1.

normal event according to ground truth. However, our approach regards this event (pedestrian walks from bottom to left) as anomaly because it never happened in the training set, which is consistent with the definition of anomaly detection.

Compared with the state-of-the-art approaches in Table 5.2, the EER performance of our proposed algorithm is comparable to that of H-MDT-CRF (Li *et al.*, 2014) and LDA (Weinshall *et al.*, 2013) on the frame-level, but outperforms H-MDT-CRF on the pixel-level.

Table 5.2: Quantitative comparison of our method and the state-of-the-art approaches.

Method	Frame-level EER (%)	Pixel-level EER (%)
H-MDT-CRF (Li *et al.*, 2014)	19	30
LDA (Weinshall *et al.*, 2013)	16	–
Online adpative OCSVM	**18**	**23**

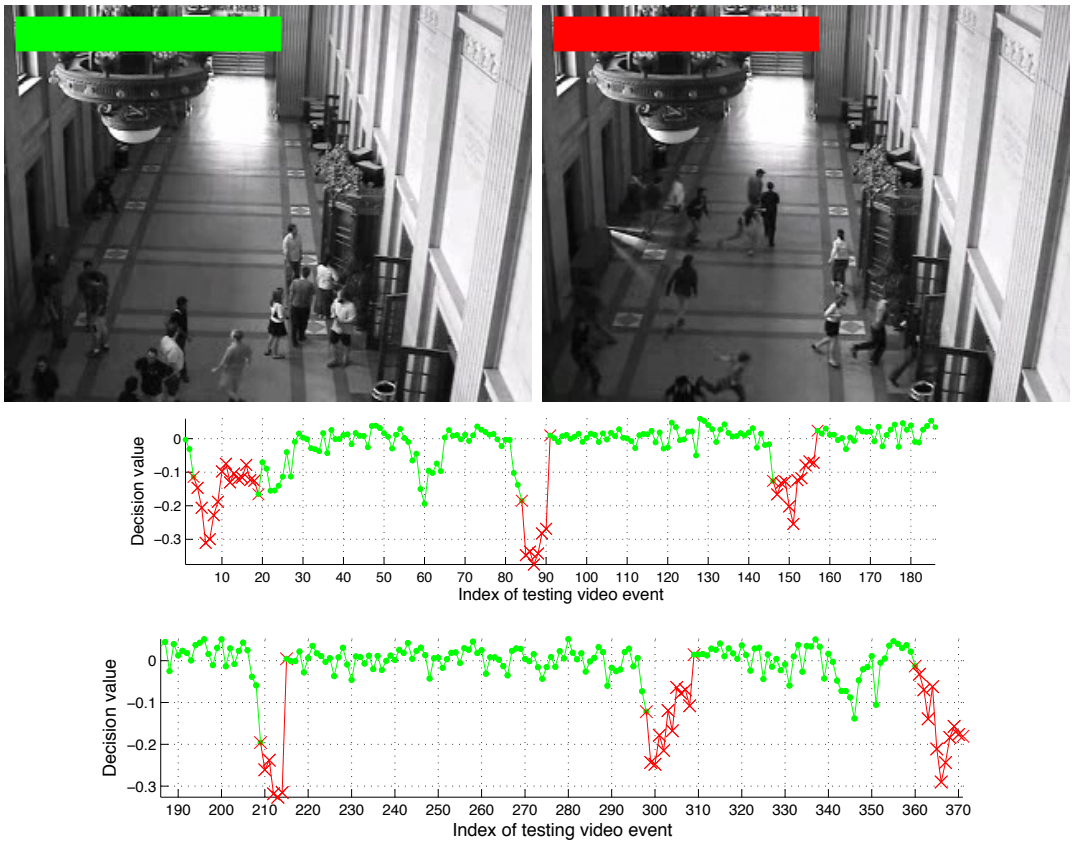


Figure 5.8: Experimental results of the online adaptive OCSVMs model on UMN dataset Scene 2.

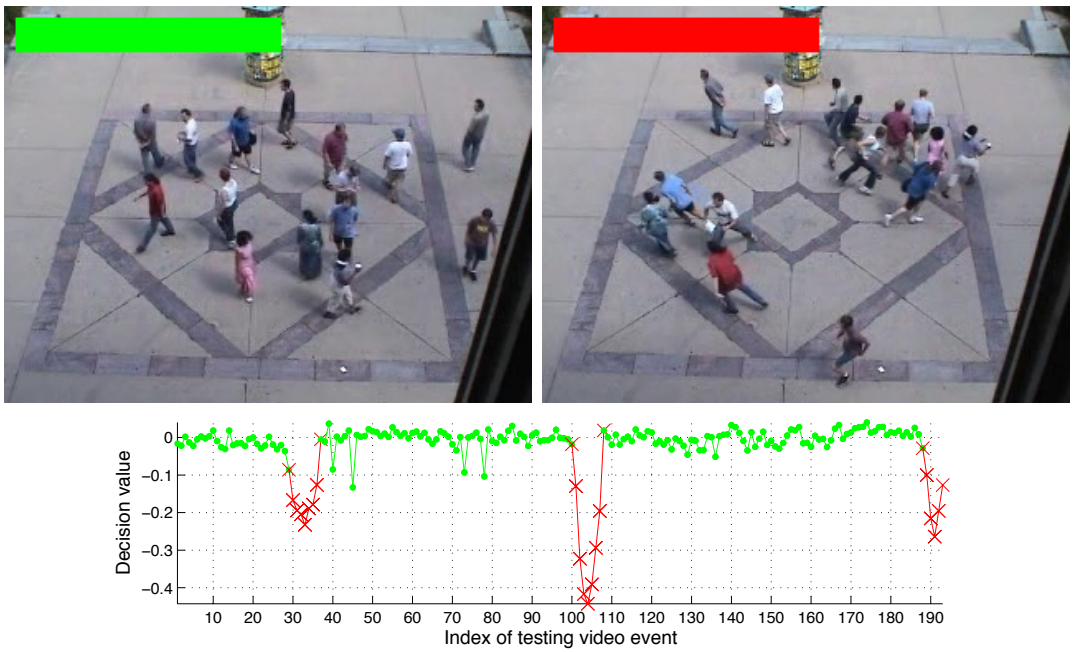


Figure 5.9: Experimental results of the online adaptive OCSVMs model on UMN dataset Scene 3.

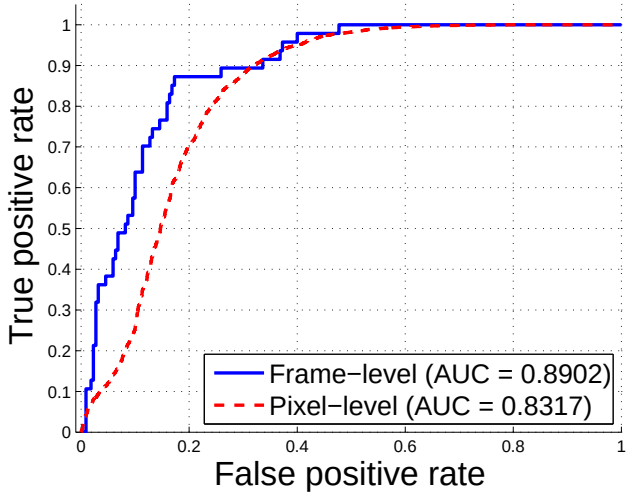


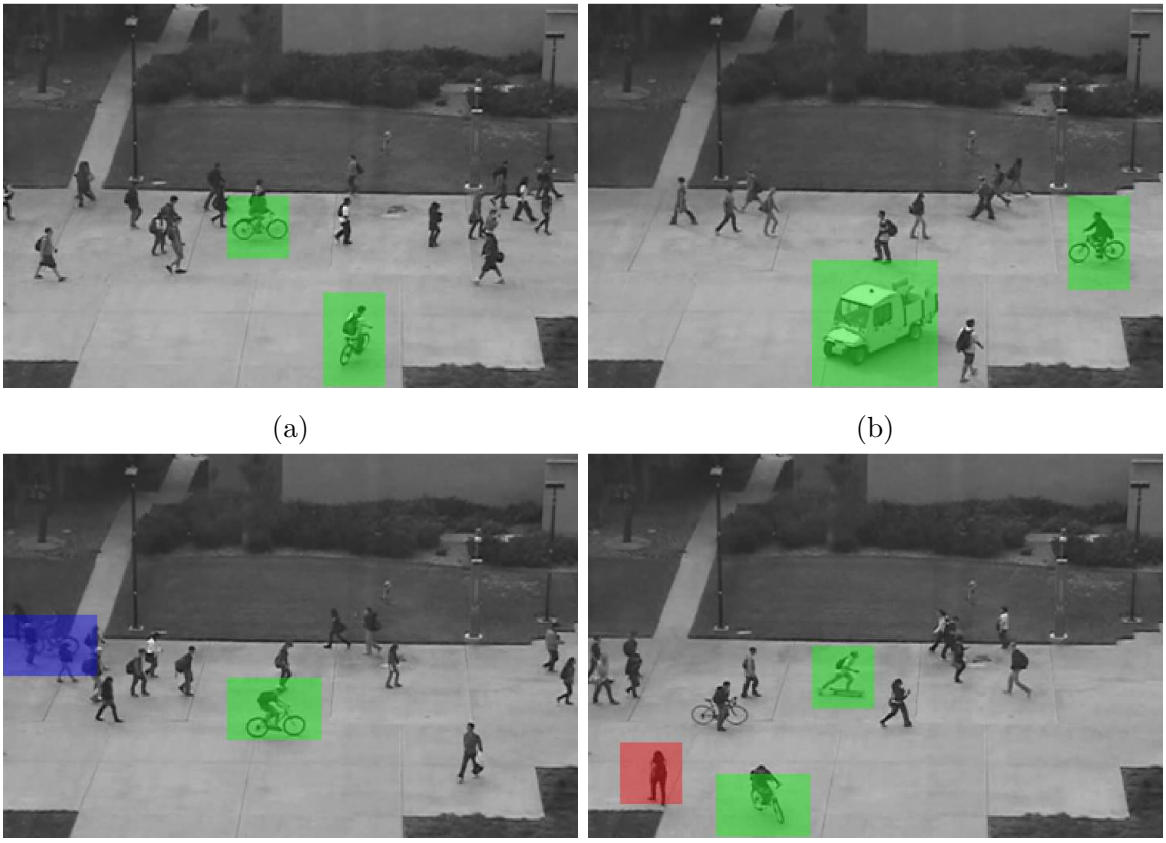
Figure 5.10: The ROC curves of the frame-level and the pixel-level on Ped2 dataset.

Table 5.3: Comparison of the results on the Subway Exit dataset.

	Wrong direction	Loitering	Misc	False alarm
Ground truth	9	3	7	0
Real-time (Adam *et al.*, 2008)	9	-	-	0
SRC (Cong *et al.*, 2011)	9	-	-	0
Sparse coding (Zhao *et al.*, 2011)	9	3	7	2
Online adpative OCSVM	**9**	**3**	**5**	**2**

5.4.3 Subway Exit dataset

To reduce computational cost, the original frames are first down sampled from 384×512 to 240×320, and we define the size of event as $50 \times 50 \times 15$ with $20 \times 20 \times 7$ overlap in both training and testing dataset. The online adaptive OCSVM model has the same settings as in UCSD Ped2 dataset. Following the practice in (Adam *et al.*, 2008), the first 6 minutes ($9,000$ frames) are used for training, and the rest for testing. Figure 5.12 illustrates some detection results from the Subway Exit dataset, where the green, red, and blue rectangles correspond to correct detection, false alarms and missing detections. It should be noted that the normal events in Figure 5.12(c)(d) are detected as an anomaly because the corresponding event type (pedestrians walking from right to left) was not learned in the training process. The comparative experimental results are shown in Table 5.3. It can be seen that our model has achieved a comparable result.

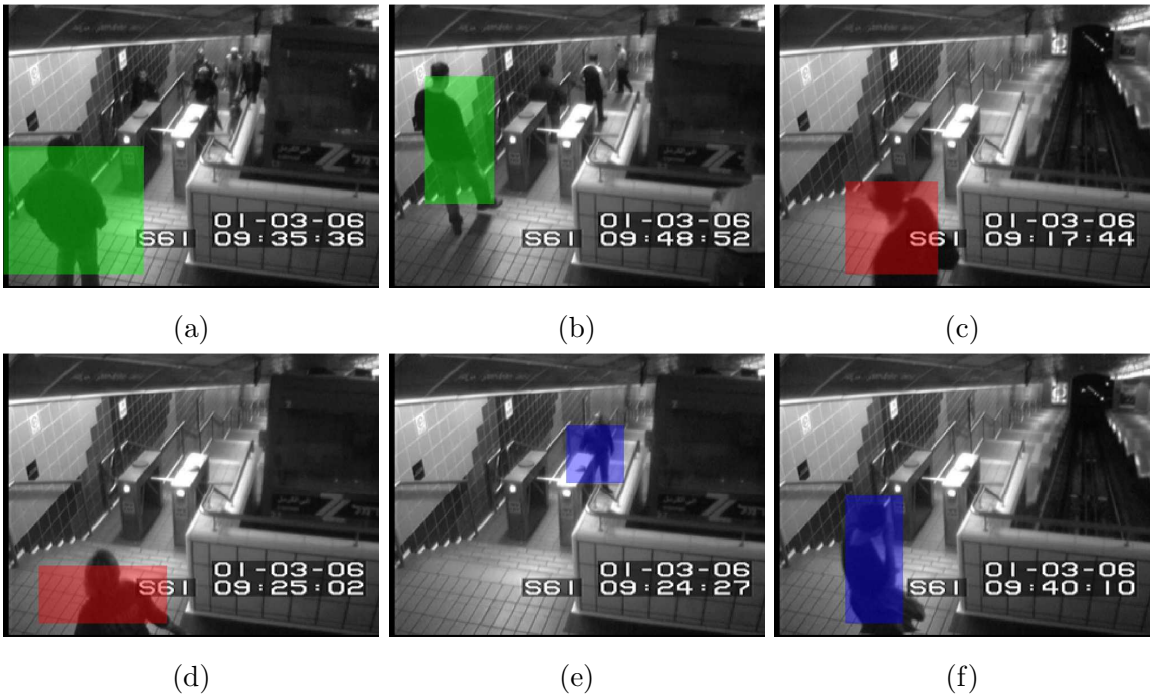


(a) (b)

(c) (d)

Figure 5.11: Examples of detected abnormal events, where the green rectangles are true negatives, the blue and red rectangle are false positive and false negative respectively.

Our algorithm is implemented using MATLAB R2014a on a 2.7GHZ Intel Core i5 with 8GB RAM. The average computation time is 0.11 second/frame for the UMN dataset, 0.19 second/frame for the UCSD Ped2 dataset, and 0.28 second/frame for the Subway Exit dataset.



(a) (b) (c)

(d) (e) (f)

Figure 5.12: LA detection results on Subway Exit dataset. (a)(b) Correct detections. (c)(d) False alarms. (e)(f) Missing detections.

5.5 Summary

This chapter presents a novel framework for AED on the basis of OCSVM. By keeping the KKT conditions satisfied for the changing data set, our approach effectively updates OCSVM classifier in an online fashion. The online algorithm along with the use of a sliding buffer can adapt to new patterns and forget obsolete patterns at the same time. Satisfactory performance is gained for the detection of both global and local anomalies using benchmark datasets.

The proposed framework, however, still poses some drawbacks. First, the classifier is trained and tested on whole video content exhaustively, which not only increases computational cost, but also deteriorates the detection performance due to including noise. Second, the motion features of same object from different locations may vary

due to perspective of camera, anomaly in distant view may be ignored due to small scale motion. One possible solution is to divide the spatial position of frame into grids, in each grid a classifier is trained. However, how to set the threshold for performance evaluation is still challenging. Last but not least, we may receive multiple data at each time for LA detection, which is time-consuming to update data once each time, namely, single update. In the next chapter, we will propose an improved framework to address these limitations.

Chapter 6

Multiple Incremental and Decremental Learning OCSVM for AED

6.1 Overview

In the previous chapter, we have proposed a novel online adaptive OCSVM model to detect abnormal events in crowd scene. The algorithm solves a linear system and constructs a new exact solution to keep the KKT conditions satisfied for the updated data, thus is able to add and remove data in an online fashion. Although it demonstrates satisfactory performance, more challenges have to be met.

On the one hand, the feature extraction scheme of previous approach is densely sampled, which requires high computational overhead and cannot meet the requirement of real-time constraint in video surveillance application. On the other hand, while the event representation scheme for GA detection is unable to localize the anomaly in a frame, we may receive many data at a time in LA detection, which also raises computational burden if we update one data each time.

Apart from the challenges of dealing with large amounts of streaming data, other factors may also significantly affect the performance of abnormal event detection. For instance, camera perspectives may result in scale variation of the extracted features. An event occurring in a close range may produce significant optic flow features, hence is likely to be recognized as anomaly, but due to perspective distortion, the same type of events occurring in a remote location may be deemed as normal. Furthermore, abnormal events in surveillance videos may not be confined to motion anomalies only.

For example, a "normal" event appearing at a location where it has never occurred before is supposed to be detected as an anomaly in certain scenarios, a case we will refer to as "contextual anomaly" from now on.

To tackle the problem of AED in video surveillance, we propose a Multiple Incremental and Decremental Learning with Kernel Fusion (MIDL-KF) framework. Compared with the computational framework in Chapter 5, the main improvements are as follows:

- We simplify the feature extraction scheme by extracting Region Of Interest (ROI) and proposing an Adaptive Multi-scale Histogram of Optical Flow (AMHOF) descriptor. The former decreases the computational cost and enhances robustness of the anomaly detector, while the latter adapts to different environments by adaptively deciding its scale binning using a clustering process.

- By combining motion information as well as spatial location information through kernel fusion, the MIDL-KF framework is able to detect motion anomalies as well as contextual anomalies even under perspective distortions.

- We extend the update scheme from single instance incremental and decremental learning to multiple Incremental and Decremental Learning (MIDL). Specifically, instead of updating one data after another, MIDL adopts a variable cache to store the received data entries in each frame and updates them together, which reduces computational cost effectively.

- The detection result has been improved further by a simplified Multiple Target Tracking (MTT) algorithm.

6.2 The computational framework

The MIDL-KF framework is illustrated in Figure 6.1. Given a training video clip, we first detect potential ROIs of each frame using a background subtraction approach. Then we extract the corresponding motion information (i.e., AMHOF) and spatial location from these ROIs. Combining the two sources of information using kernel fusion, MIDL is employed to update the OCSVM classifier continuously. The procedure is repeated until all the training data have been learnt. In the testing process, we have the same steps to detect potential ROIs and extract multiple sources of features, where the ROIs are detected as normal or abnormal by the learnt classifier.

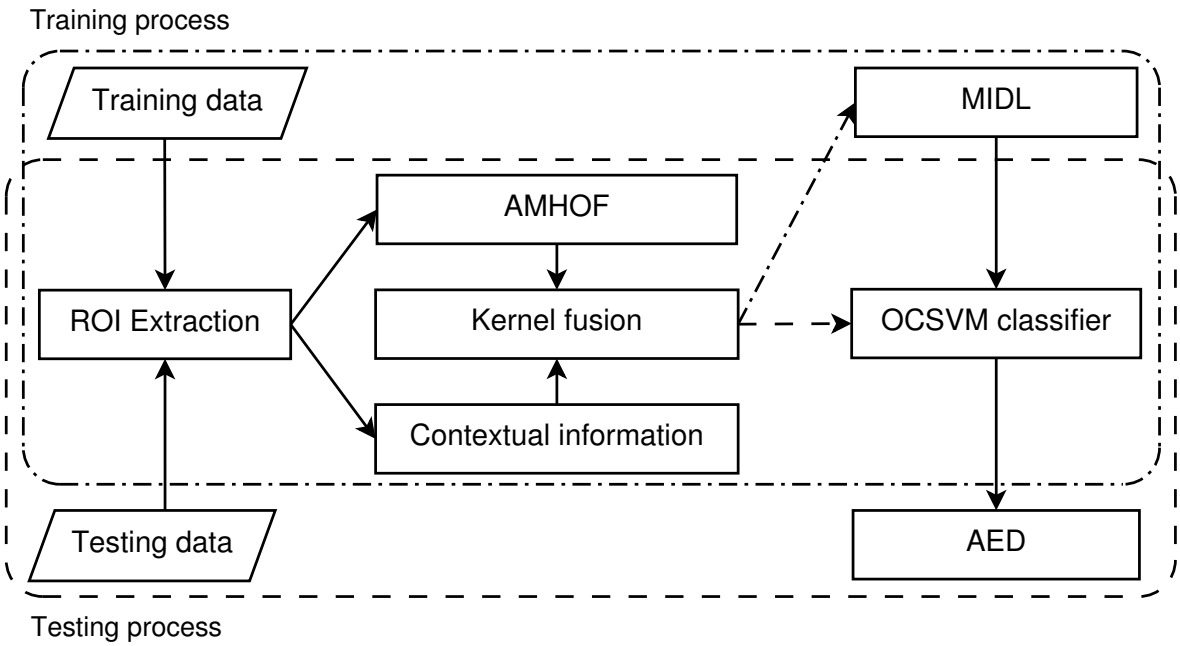


Figure 6.1: The MIDL-KF framework for abnormal event detection.

6.2.1 ROI extraction

(a) (b) (c)

Figure 6.2: ROI extraction based on background subtraction. (a) Incoming frame. (b) Foreground. (c) ROIs. Best viewed in colour.

A large proportion of visual content in video surveillance is background. Training a detector using whole frame content increases the computational costs and may compromise the detection performance due to excessive noise. Fortunately, for most surveillance applications stationary cameras are used, hence applying background subtraction to extract ROIs is feasible.

After modeling the background via the Gaussian Mixture Model (GMM) model (Stauffer and Grimson, 1999; Zivkovic, 2004), a sliding window is used to scan through the detected foreground of current frame and detect ROIs which may be related to

anomaly detection. Specifically, for all pixels inside the sliding window, if the proportions of foreground pixels is greater than a threshold (i.e., empirically set as 0.2 in our experiment), then the window is highlighted as a ROI.

Figure 6.2 displays the process of ROI extraction. As can been seen, this process filters most of the background out, reducing the computational cost as well as enhancing robustness of the anomaly detector. Obviously, background subtraction approach is not restricted to the model, which can be easily replaced by other ones, e.g., Kernel Density Estimation (KDE) (Elgammal *et al.*, 2000). This is however not the focus of the thesis.

6.2.2 Multiple feature extraction

After identifying the ROIs, we extract motion information and contextual information from them.

Due to the fact that an anomaly does not occur only in one frame, we employ a sliding window centred at time t – $\{t-n, \cdots, t-1, t, t+1, \cdots, t+n\}$ – to extract the AMHOF descriptor. Specifically, in each ROI optical flow of each pixel is calculated using Horn and Schunck's method (Horn and Schunck, 1981). The flow vector containing two directional components is denoted as $\{f_x, f_y\}$. The orientation α and magnitude f of the flow vectors are calculated as:

$$\alpha = \arctan(\frac{f_y}{f_x}),$$
$$f = \sqrt{f_x^2 + f_y^2},$$

where the range of α is $[-\pi/2, \pi/2]$, and thus mapped to $[0, 2\pi)$.

Next, all flow vectors are quantized into $M \times N$ bins, where M is number of directions, and N is the discretization level of motion intensities. The discretization centres of motion intensities are determined by performing k-means clustering ($k = N$) of a random subset of magnitude f of flow vectors from the training set. This step is a simplified BoW model, where each flow vector is treated as a word, thus reducing computations compared with our previous BoW model. Our AMHOF adapts to different scenarios by adjusting the centres automatically, which is different from the Multi-scale Histogram of Optical Flow (MHOF) in Cong *et al.* (2013), where the threshold of 2-level MHOF was empirically set to a fixed value.

Besides motion information, we also consider the contextual information, i.e., spatial location (i.e., x and y coordinates) of each ROI in the frame. Figure 6.3 shows

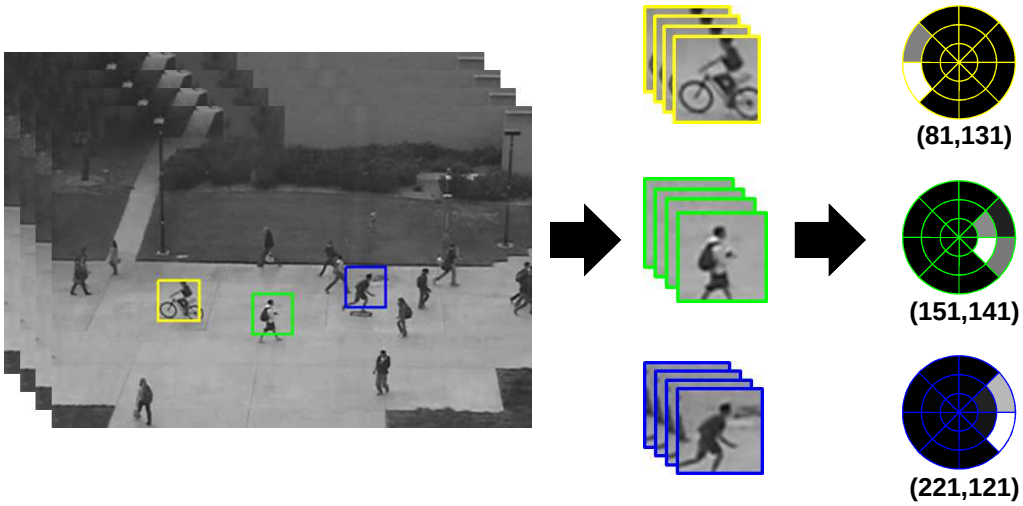


Figure 6.3: Multiple feature extraction from three different ROIs ($M = 8$, $N = 3$), where the colours Yellow, Green and Blue correspond to biker, pedestrian, and skateboarder, respectively. Best viewed in colour.

multiple features extracted from three ROIs, where the AMHOF discriminates different motion patterns by capturing both direction and motion intensity information.

6.2.3 Kernel Fusion

It has been demonstrated that using a kernel method to combine multiple sources of information improves classification performance (Joachims *et al.*, 2001; Camps-Valls *et al.*, 2006; Lanckriet *et al.*, 2004). It is also demonstrated in a case study (Das *et al.*, 2010) that the use of multiple kernels leads to a more powerful anomaly detection learning scheme. The basic idea of the kernel method is to define a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ to measure the similarity between data $\mathbf{x}_i$ and $\mathbf{x}_j$ in the feature space $\mathcal{F}$. Rather than selecting only one specific kernel function, Multiple Kernel Learning (MKL) (Rakotomamonjy *et al.*, 2008; Gönen and Alpaydın, 2011; Kembhavi *et al.*, 2009) defines a combination function to fuse multiple kernels:

$$k(\mathbf{x}_i, \mathbf{x}_j) = f(\{k_n(\mathbf{x}_i^n, \mathbf{x}_j^n)\}_{n=1}^{S}),$$

where $\mathbf{x}_i^n$ and $\mathbf{x}_j^n$ denote the relevant data from feature source n, and S is the number of multiple kernels.

We define the combination function as follows. Let $(\mathbf{d}_i, \mathbf{l}_i)$ and $(\mathbf{d}_j, \mathbf{l}_j)$ be the features extracted from ROIs $\mathbf{x}_i$ and $\mathbf{x}_j$, where $\mathbf{d}_i$ and $\mathbf{d}_j$ correspond to motion descriptors, namely AMHOF, and $\mathbf{l}_i$ and $\mathbf{l}_j$ correspond to the spatial locations. The similarity be-

tween $\mathbf{x}_i$ and $\mathbf{x}_j$ is given as:

$$k(\mathbf{x}_i, \mathbf{x}_j) = k_1(\mathbf{d}_i, \mathbf{d}_j) \cdot k_2(\mathbf{l}_i, \mathbf{l}_j), \tag{6.1}$$

where $k_1(\mathbf{d}_i, \mathbf{d}_j)$ measures motion similarity, and $k_2(\mathbf{l}_i, \mathbf{l}_j)$ measures spatial similarity.

We evaluate four kernel functions to measure the similarity between two AMHOF descriptors $\mathbf{d}_i$ and $\mathbf{d}_j$:

1. Linear kernel
$$k_1(\mathbf{d}_i, \mathbf{d}_j) = \mathbf{d}_i \cdot \mathbf{d}_j, \tag{6.2}$$

2. Radial Basis Function (RBF) kernel
$$k_1(\mathbf{d}_i, \mathbf{d}_j) = \exp\left(-\frac{\|\mathbf{d}_i - \mathbf{d}_j\|^2}{2\sigma_d^2}\right), \tag{6.3}$$

3. χ^2 kernel (Vedaldi and Zisserman, 2012)
$$k_1(\mathbf{d}_i, \mathbf{d}_j) = \sum_b \frac{2\mathbf{d}_i^b \mathbf{d}_j^b}{\mathbf{d}_i^b + \mathbf{d}_j^b}, \tag{6.4}$$

4. Histogram Intersection (HI) kernel (Swain and Ballard, 1991)
$$k_1(\mathbf{d}_i, \mathbf{d}_j) = \sum_b \min(\mathbf{d}_i^b, \mathbf{d}_j^b), \tag{6.5}$$

where b in Eq. (6.4) and (6.5) is the corresponding bin in $\mathbf{d}_i$ and $\mathbf{d}_j$.

To measure the contextual similarity, we use the RBF kernel:

$$k_2(\mathbf{l}_i, \mathbf{l}_j) = \exp\left(-\frac{\|\mathbf{l}_i - \mathbf{l}_j\|^2}{2\sigma_l^2}\right) \tag{6.6}$$

where σ_l is the span of the kernel.

To sum up, Eq. (6.1) indicates any two ROIs are similar only when they produce similar motion features *and* they are spatially close to each other. Experimental results verify that MIDL-KF not only overcomes the difficulty of perspective distortion, but also has the capability to detect spatial location anomalies.

6.2.4 MIDL framework

The framework of our MIDL algorithm is illustrated in Figure 6.4. An OCSVM classifier can be initialized by training it over a mini-batch of the data stream. Suppose we have trained the OCSVM classifier using the data in a variable buffer at time t.

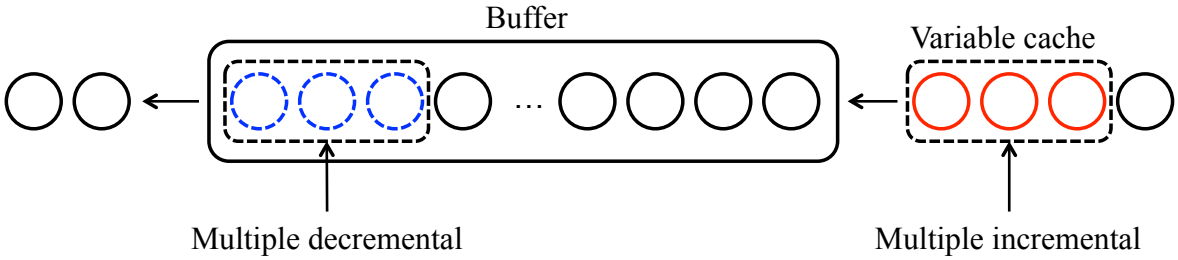


Figure 6.4: The MIDL framework. New arriving data instances to be added into the buffer are indicated by red solid circles, while the obsolete instances to be removed from the buffer are indicated by blue dashed circles.

From t to $t+1$, a variable cache is deployed to store incoming data. Then we add these incoming data into the buffer and remove those obsolete data while updating the classifier. Note that MIDL is flexible in the sense that we can choose to adopt multiple incremental learning with or without multiple decremental learning, depending on which is appropriate for the application scenario.

MIDL extends the work in Karasuyama and Takeuchi (2009) from a binary classification scenario to a one-class scenario. Compared with our previous work where each time the learning model can only update on a single data entry, our approach exhibits the ability to update multiple data simultaneously, thus effectively reducing the computational cost.

Derivation

For the multiple incremental and decremental learning process, suppose we add p new arriving data and remove q obsolete data simultaneously. Let $\mathcal{A} = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ and $\mathcal{R} = \{\mathbf{x}_1, \ldots, \mathbf{x}_q\} \subset \mathcal{X}$ be the sets of addition and removal, in parallel $I_\mathcal{A}$ and $I_\mathcal{R}$ denote index sets of $\mathcal{A}$ and $\mathcal{R}$.

We first initialize the coefficients of addition set $\mathcal{A}$ as zeros, i.e., $\alpha_i = 0, \forall i \in I_\mathcal{A}$, and remove $\mathcal{R}$ from $\mathcal{X}$, i.e., $\mathcal{X} \leftarrow \mathcal{X} \setminus \mathcal{R}$. For $g_i > 0, \forall i \in I_\mathcal{A}$, we append these data directly to $\mathcal{O}$ because they already satisfy the KKT conditions. Likewise, we discard the data in $\mathcal{R}$ with corresponding $\alpha_i = 0, \forall i \in I_\mathcal{R}$. For any $\mathbf{x}_i$ having $g_i \leq 0$ in $\mathcal{A}$ and $\mathcal{R}$, the KKT conditions are to be kept:

$$\Delta g_i = \sum_{j \in I_\mathcal{A}} k_{ij} \Delta\alpha_j + \sum_{j \in I_\mathcal{R}} k_{ij} \Delta\alpha_j + \sum_{j \in I_\mathcal{S}} k_{ij} \Delta\alpha_j + \Delta\rho, \tag{6.7}$$

$$\forall i \in I_\mathcal{X} \cup I_\mathcal{A},$$

$$0 = \sum_{j \in I_{\mathcal{A}}} \Delta\alpha_j + \sum_{j \in I_{\mathcal{R}}} \Delta\alpha_j + \sum_{j \in I_{\mathcal{S}}} \Delta\alpha_j.$$

For margin support vector set $\mathcal{S}$, $g_i \equiv 0, \forall i \in I_{\mathcal{S}}$. The above equations can be rewritten as the matrix notation:

$$\underbrace{\begin{bmatrix} 0 & 1 \\ 1 & k_{\mathcal{SS}} \end{bmatrix}}_{K} \begin{bmatrix} \Delta\rho \\ \Delta\alpha_{\mathcal{S}} \end{bmatrix} = - \begin{bmatrix} 1 & 1 \\ k_{\mathcal{SA}} & k_{\mathcal{SR}} \end{bmatrix} \begin{bmatrix} \Delta\alpha_{\mathcal{A}} \\ \Delta\alpha_{\mathcal{R}} \end{bmatrix}, \tag{6.8}$$

where $\Delta\alpha_{\mathcal{S}}$, $\Delta\alpha_{\mathcal{A}}$, and $\Delta\alpha_{\mathcal{R}}$ denote the changes of coefficients in sets $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{R}$.

The same as in Karasuyama and Takeuchi (2009), the change directions of $\Delta\alpha_A$ and $\Delta\alpha_R$ are given as:

$$\begin{aligned} \Delta\alpha_{\mathcal{A}} &= \eta(C\mathbf{1} - \alpha_{\mathcal{A}}), \\ \Delta\alpha_{\mathcal{R}} &= -\eta\alpha_{\mathcal{R}}, \end{aligned} \tag{6.9}$$

where η is a step length. Together with Eq. (6.8), we can write:

$$\begin{bmatrix} \Delta\rho \\ \Delta\alpha_{\mathcal{S}} \end{bmatrix} = \eta\Phi, \tag{6.10}$$

where

$$\Phi = \begin{bmatrix} \phi_\rho \\ \phi_{\mathcal{S}} \end{bmatrix} = -\underbrace{K^{-1}}_{Q} \begin{bmatrix} 1 & 1 \\ k_{\mathcal{SA}} & k_{\mathcal{SR}} \end{bmatrix} \begin{bmatrix} C\mathbf{1} - \alpha_{\mathcal{A}} \\ -\alpha_{\mathcal{R}} \end{bmatrix}. \tag{6.11}$$

Substituting Eq. (6.9) and Eq. (6.11) into Eq. (6.7):

$$\Delta g_i = \eta\Psi_i, \tag{6.12}$$

where

$$\Psi_i = k_{i\mathcal{A}}(C\mathbf{1} - \alpha_{\mathcal{A}}) - k_{i\mathcal{R}}\alpha_{\mathcal{R}} + \begin{bmatrix} 1 & k_{i\mathcal{S}} \end{bmatrix} \Phi, \quad \forall i \notin I_{\mathcal{S}}. \tag{6.13}$$

As shown in Eq. (6.10) and Eq. (6.12), the composition of the sets $\mathcal{S}$, $\mathcal{E}$ and $\mathcal{O}$ changes relative to the change of $\Delta\alpha_{\mathcal{S}}$ and Δg_i. To decide the learning step η, we consider the following cases:

1. g_i in $\mathcal{A}$ reaches zero, corresponding to $\mathbf{x}_i$ transfering from $\mathcal{A}$ to $\mathcal{S}$. The largest step is computed as:

$$\eta^{\mathcal{A}} = \min \frac{-g_i}{\Psi_i}, \quad \forall i \in I_{\mathcal{A}} \cap \Psi_i > 0.$$

2. g_i in $\mathcal{E}$ becomes zero, equivalent to $\mathbf{x}_i$ transferring from $\mathcal{E}$ to $\mathcal{S}$. The largest step $\eta^{\mathcal{E}}$ equals to finding the minimal increment:

$$\eta^{\mathcal{E}} = \min \frac{-g_i}{\Psi_i}, \quad \forall i \in I_{\mathcal{E}} \cap \Psi_i > 0.$$

3. g_i in $\mathcal{O}$ becomes zero, equivalent to $\mathbf{x}_i$ transferring from $\mathcal{O}$ to $\mathcal{S}$. The largest possible step is computed as:

$$\eta^{\mathcal{O}} = \min \frac{-g_i}{\Psi_i}, \quad \forall i \in I_{\mathcal{O}} \cap \Psi_i < 0.$$

4. $\mathbf{x}_i$ in $\mathcal{S}$ reaches a bound, α_i with equality 0 is equivalent to transferring $\mathbf{x}_i$ from $\mathcal{S}$ to $\mathcal{O}$, and equality C from $\mathcal{S}$ to $\mathcal{E}$. The largest possible increment is:

$$\eta^{\mathcal{S}} = \min \frac{\Delta\alpha_i^{\mathcal{S}}}{\phi_i}, \quad \forall i \in I_{\mathcal{S}},$$

where

$$\Delta\alpha_i^{\mathcal{S}} = \begin{cases} C - \alpha_i, & \text{if } \phi_i > 0; \\ -\alpha_i, & \text{if } \phi_i < 0. \end{cases}$$

The largest possible step length η is determined as:

$$\eta = \min(\eta^{\mathcal{A}}, \eta^{\mathcal{E}}, \eta^{\mathcal{O}}, \eta^{\mathcal{S}}, 1). \tag{6.14}$$

Once obtaining η, we can update ρ, α_i, and g_i through Eq. (6.9), (6.10) and (6.12). The procedures are repeated until η becomes 1, where all the data in $\mathcal{A}$, $\mathcal{R}$, and $\mathcal{X}$ satisfy the KKT conditions. Algorithm 3 details the learning process of MIDL.

6.3 Experimental result

The UCSD dataset and UMN dataset have been used for performance evaluation. As video clips in these datasets are fairly short in length, we employ incremental but no decremental learning to train the OCSVM. In the following presentation, unless otherwise specified, we empirically set parameter C in OCSVM as 0.1. Meanwhile, in the training process, we first use the conventional batch mode algorithm (Schölkopf *et al.*, 2001) on a small number events (i.e., $2,000$ in the following experiments) to obtain the initial parameters. Then incremental learning is deployed on the rest of the training data stream.

6.3.1 UCSD anomaly detection dataset

Setup

We set the orientation bin number M in AMHOF as 8 empirically. For the Ped1 dataset, we apply a 20×10 (i.e., height $\times$ width) sliding window with 1/2 overlap to

Algorithm 3 MIDL

Input: $\mathcal{S}$, $\mathcal{E}$, $\mathcal{O}$, $\{\alpha_i\}_{i=1}^{n}$, $\{g_i\}_{i=1}^{n}$, ρ, Q, $\mathcal{A}$, $I_{\mathcal{R}} \in I_{\mathcal{X}}$

Output: $\mathcal{S}$, $\mathcal{E}$, $\mathcal{O}$, $\{\alpha_i\}_{i=1}^{n+p-q}$, $\{g_i\}_{i=1}^{n+p-q}$, ρ, Q

Initialize $\alpha_{\mathcal{A}} \leftarrow 0$, compute $g_{\mathcal{A}}$ using Eq. (5.7).

$\mathcal{A} \leftarrow \mathcal{A} \setminus \mathbf{x}_i \quad \mathcal{O} \leftarrow \mathcal{O} \cup \mathbf{x}_i,\ \forall i \in I_{\mathcal{A}} \cap g_i > 0$

Initialize $\mathcal{R} \leftarrow \emptyset$

$\mathcal{R} \leftarrow \mathcal{R} \cup \mathbf{x}_i \quad \mathcal{S} \leftarrow \mathcal{S} \setminus \mathbf{x}_i \quad \mathcal{E} \leftarrow \mathcal{E} \setminus \mathbf{x}_i \quad \mathcal{O} \leftarrow \mathcal{O} \setminus \mathbf{x}_i,\ \forall i \in I_{\mathcal{R}}$

if $\mathcal{S}$ has changed **then**

Update Q according to Eq. (5.19) or Eq. (5.20).

end if

$\mathcal{R} \leftarrow \mathcal{R} \setminus \mathbf{x}_i,\ \forall i \in I_{\mathcal{R}} \cap \alpha_i = 0$

while true do

Compute Φ and Ψ_i using Eq. (6.11) and Eq. (6.13).

Compute the largest possible step η according to Eq. (6.14).

$\rho \leftarrow \rho - \eta\phi_\rho$

$\alpha_{\mathcal{S}} \leftarrow \alpha_{\mathcal{S}} + \eta\phi_{\mathcal{S}}$

$\alpha_{\mathcal{A}} \leftarrow \alpha_{\mathcal{A}} + \eta(C\mathbf{1} - \alpha_{\mathcal{A}})$

$\alpha_{\mathcal{R}} \leftarrow \alpha_{\mathcal{R}} - \eta\alpha_{\mathcal{R}}$

$g_i \leftarrow g_i + \eta\Psi_i,\ \forall i \notin I_{\mathcal{S}}$

Update $\mathcal{A}$, $\mathcal{S}$, $\mathcal{E}$, and $\mathcal{O}$ according to occurred condition.

if $\mathcal{S}$ has changed **then**

Update Q according to Eq. (5.19) or Eq. (5.20).

end if

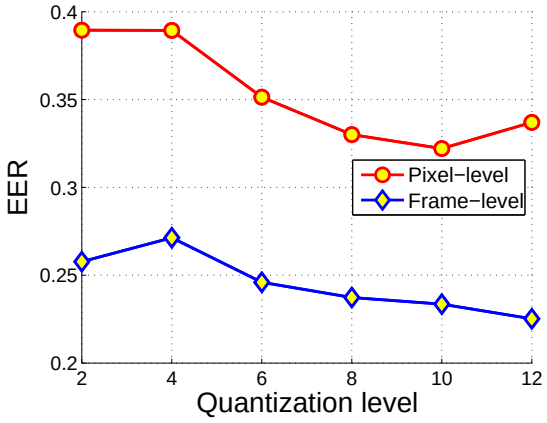
if $\eta = 1$ **then**

$\mathcal{E} \leftarrow \mathcal{E} \cup \mathcal{A} \quad \mathcal{R} \leftarrow \emptyset$

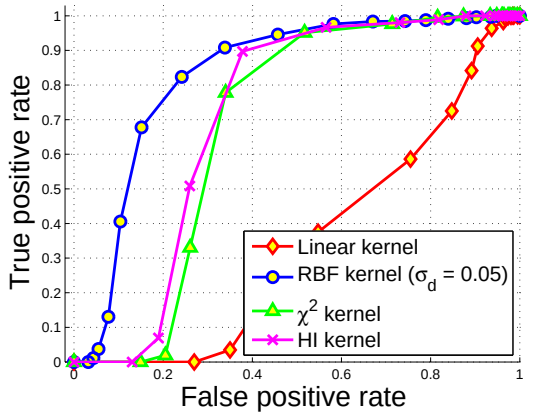
return

end if

end while



(a)



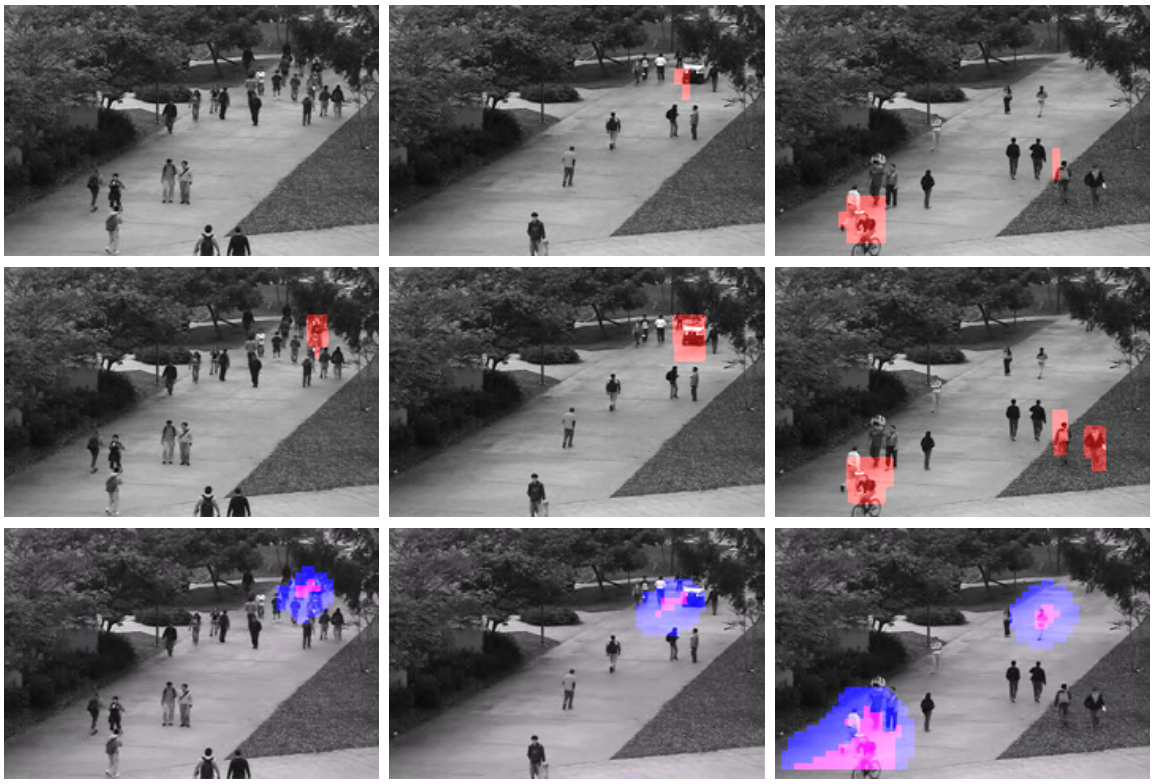
(b)

Figure 6.5: Parameter tuning for the Ped1 dataset. (a) EER variation over different quantization levels N. (b) The pixel-level (full annotation) ROC curves with various kernel metrics.

extract ROIs, and the temporal length of an event is set as 6. We resize the frame size to 160×240 so that the sliding window is able to scan whole frame. For Ped2, we apply a 30×30 sliding window with 2/3 overlap, while the temporal length is set to 8. The selection of the temporal lengths is aimed at the trade-off between detection accuracy and timely response.

UCSD Ped1 dataset

First, we need to find out the optimal parameter settings. There are three parameters to be considered, namely, the quantization level N of AMHOF, the kernel metric for motion similarity, and the bandwidth of the RBF kernel σ_l for contextual similarity



(a) (b) (c)

Figure 6.6: Detection results on Ped1. First row: MIDL. Second row: MIDL-KF. Third row: H-MDT-CRF, where the red masks are predicted by the CRF filter, and blue masks are predicted by simple thresholding. Best viewed in colour.

control. A simple approach is adopted by tuning one parameter while fixing the other parameters. We first change N from 2 to 12 while using motion feature only and the HI kernel for similarity measure. This simplified version without using kernel fusion is denoted as "MIDL" to differentiate it from MIDL-KF. As shown in Figure 6.5(a), the frame-level EER does not show any significant improvement as frame-level only measures temporal location accuracy, some arbitrary anomaly detection (i.e., a random location in an anomaly frame is detected as anomaly) may compensate for loss of performance. Instead, the full annotation pixel-level EER is more convincing. It decreases with N increasing from 2 to 10, and saturates when N enlarges from 10 to 12. With N equal to 10, Figure 6.5(b) illustrates the RBF kernel is superior to the other three kernels, where the bandwidth σ_d is acquired through cross validation. Using the same scheme, we obtain the optimal settings for the Ped1 dataset, i.e., $N = 10$, RBF kernel ($\sigma_d = 0.05$), and $\sigma_l = 40$.

Figure 6.6 illustrates the superiority of MIDL-KF over MIDL as well as H-MDT-

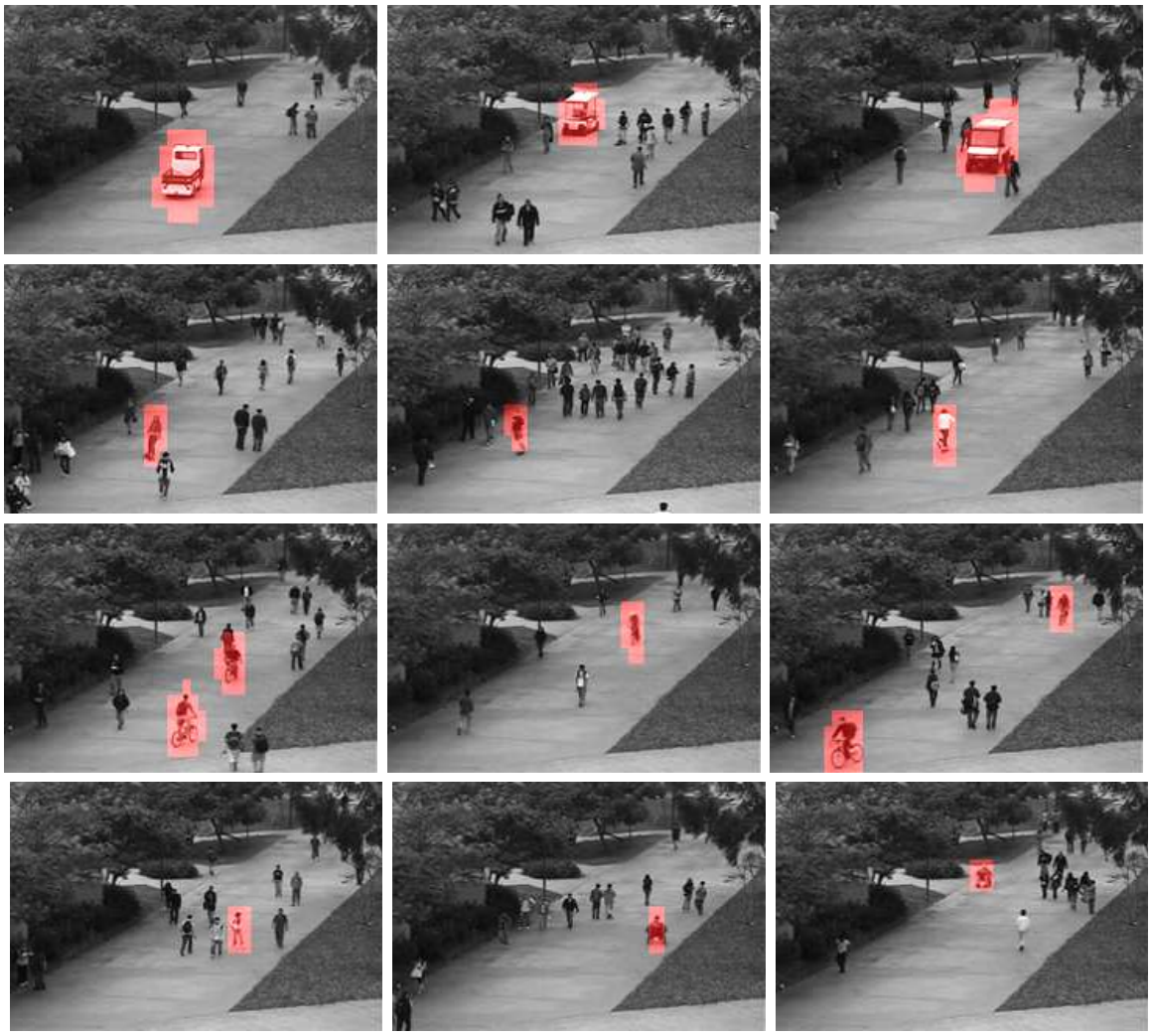
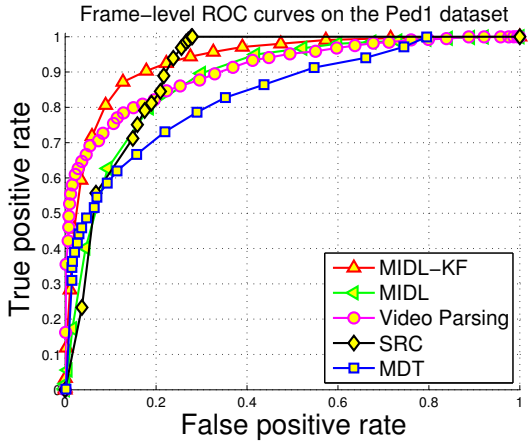


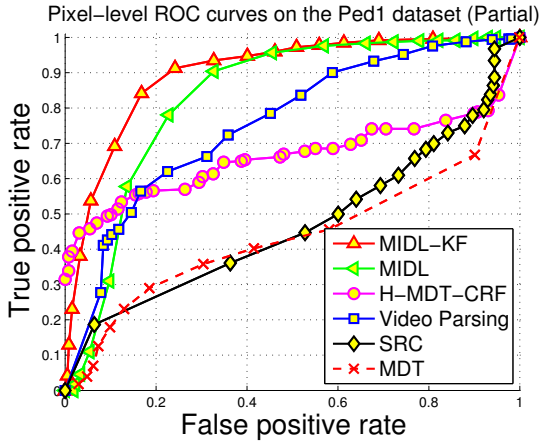
Figure 6.7: Detection results of MIDL-KF on Ped1. First row: cars. Second row: skateboarders. Third row: bikes. Fourth row: others. Best viewed in colour.

CRF (Li *et al.*, 2014). By comparison with detection results of MIDL, it demonstrates that MIDL-KF not only overcomes the shortcoming of perspective distortion, namely, detecting abnormal event in distant view (Figure 6.6(a)(b)), but also detects spatial location anomalies (Figure 6.6(c)). On the other hand, apart from detecting the contextual abnormal event (Figure 6.6(c)), MIDL-KF achieves more accurate abnormal event localization than H-MDT-CRF. More detection results are displayed in Figure 6.7.

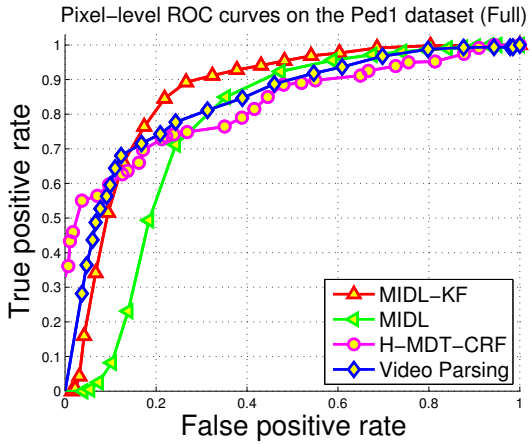
Figure 6.8 shows the ROC curves comparison of frame-level and pixel-level (partial annotation and full annotation) between MIDL-KF and the state-of-the-art approaches, including MDT (Mahadevan *et al.*, 2010), H-MDT-CRF (Li *et al.*, 2014), Video Parsing (Antić and Ommer, 2011), and Sparse Reconstruction Cost (SRC) (Cong *et al.*,



(a)



(b)



(c)

Figure 6.8: Comparison of ROC curves for Ped1: (a) frame-level, (b) pixel-level: partial annotation, and (c) pixel-level: full annotation.

Table 6.1: Performance comparison with the state-of-the-art on Ped1.

	Frame-level %		Pixel-level (%)			
			Partial		Full	
Method	**AUC**	**RD**	**AUC**	**RD**	**AUC**	**RD**
MDT (Mahadevan *et al.*, 2010)	83.8	75.6	44.0	49.4	-	-
H-MDT-CRF (Li *et al.*, 2014)	-	82.2	66.2	64.9	82.7	75.1
Video Parsing (Antić and Ommer, 2011)	90.5	81.9	75.6	67.6	83.6	76.7
SRC (Cong *et al.*, 2013)	90.2	80.8	47.1	46.0	-	-
SRC with EMD (Zhu *et al.*, 2014)	-	85.0	-	53.0	-	-
SCL (Lu *et al.*, 2013)	91.8	85.0	63.8	59.1	-	-
MIDL	87.9	80.7	83.0	76.1	76.5	73.4
MIDL-KF	**93.4**	**87.2**	**90.1**	**83.7**	**86.1**	**80.2**

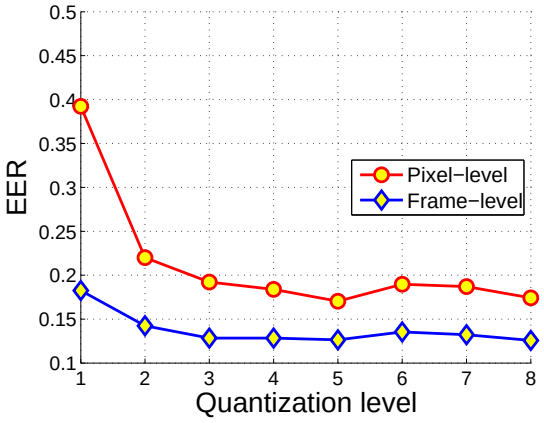
2013). In short, our approach achieves the best AUC both in frame-level and pixel-level evaluations.

For frame-level performance, MIDL-KF also compares favourably with the state of the art, including SCL (Lu *et al.*, 2013) and SRC with EMD (Zhu *et al.*, 2014), on the RD metric given in Table 6.1. As for pixel-level performance, we gain the RD of 83.7% in partial annotation and 80.2% in full annotation, both outperforming the other approaches.

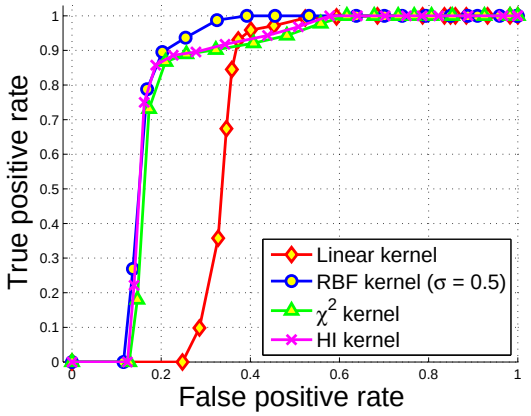
UCSD Ped2 dataset

Similar to the Ped1 dataset, we first obtain the optimal settings by tuning one parameter while fixing the other parameters. As depicted in Figure 6.9, we compare our results with the state-of-the-art approaches using the obtained optimal settings, i.e., $N = 5$, RBF kernel ($\sigma_d = 0.5$), and $\sigma_l = 100$.

Figure 6.10 displays both the frame-level and pixel-level ROC curves of MIDL-KF in comparison with the state-of-the-art. Compared with MIDL, there is only a slight improvement using MIDL-KF. By looking through the Ped2 dataset, we believe this is due to the fact that the anomaly is mostly related to fast motions only and the perspective distortion is insignificant, therefore using AMHOF alone is sufficient to detect the abnormal events. Figure 6.11 shows the detection results on a sample frame comparing with MIDL and H-MDT-CRF. More detection results are illustrated in Figure 6.12.



(a)



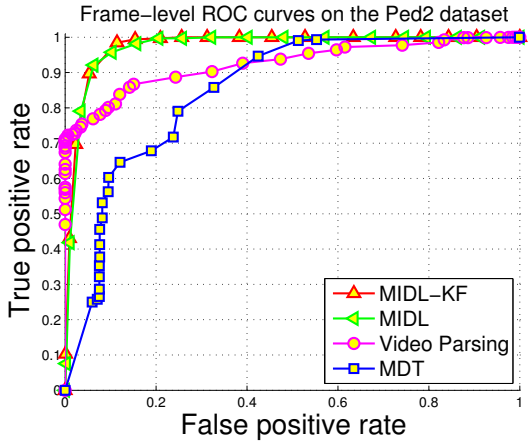
(b)

Figure 6.9: Parameter tuning for Ped2. (a) EER variation over different quantization levels N. (b) The pixel-level ROC curves with various kernel metrics.

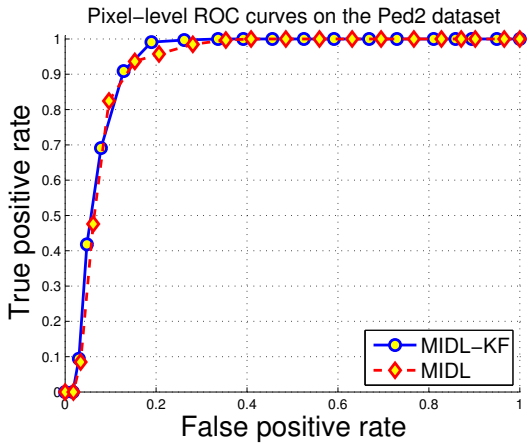
The comparison with the state-of-the-art approaches is presented in Table 6.2. Our frame-level RD has been improved by around 7.1% over Video Parsing (Antić and Ommer, 2011). We also improve the pixel-level RD by about 17.8% over H-MDT-CRF, attaining 87.9%.

6.3.2 UMN dataset

We empirically set the sliding window size as 12×16 with an overlap of 6×8, and the temporal length is set as 10. The direction information in the dataset is useless, so we set the direction bin M as 1, namely we only consider motion magnitude. The quantization level N is set as 10 to differentiate the motion intensities. The abnormal events happen in the whole scene and the perspective distortion is insignificant, so



(a)

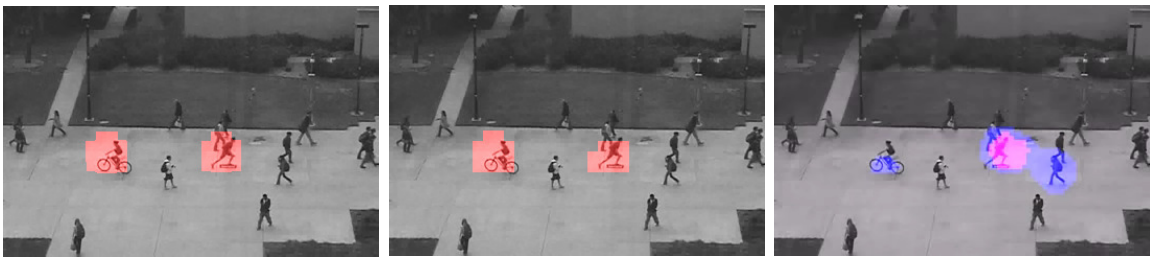


(b)

Figure 6.10: Comparison of ROC curves for Ped2: (a) frame-level, and (b) pixel-level.

here we only adopt MIDL for simplicity. For comparison purposes, we use the first 400 frames of each scene for training, and the rest for testing. Figure 6.14 displays detection results on some sample frames in the UMN dataset. Like (Li *et al.*, 2014), our approach detects and localizes anomalies simultaneously, which is different from our previous work, where the GA detection cannot localize the position of anomalies.

The UMN seems an easier dataset and the state of the art results are very good. Figure 6.13 presents the comparison between our MIDL and previous results in the literature. The AUC and RD results are given in Table 6.3. We gained an AUC of 99.5% and a RD of 96.5%, which are better than most of previous work and are comparable to Cong *et al.* (2013).



(a) (b) (c)

Figure 6.11: Detection results on Ped2: (a) MIDL, (b) MIDL-KF, and (c) H-MDT-CRF. Best viewed in colour.

Table 6.2: Performance comparison with the state-of-the-art on Ped2.

	Frame-level (%)		Pixel-level (%)	
Method	**AUC**	**RD**	**AUC**	**RD**
MDT (Mahadevan *et al.*, 2010)	84.8	74.0	-	45.0
H-MDT-CRF (Li *et al.*, 2014)	-	81.5	-	70.1
Video Parsing (Antić and Ommer, 2011)	91.0	85.8	-	-
LDA (Weinshall *et al.*, 2013)	-	16	-	-
MIDL	97.6	92.2	92.4	87.3
MIDL-KF	**97.7**	**92.9**	**93.3**	**87.9**

6.3.3 Computational efficiency analysis

Our framework is implemented using MATLAB R2014a and runs on a 2.7GHz Intel Core i5 with 8GB RAM.

To validate the computational efficiency of multiple incremental learning, we change the cache size from 1 to 150, and evaluate the detection performance as well as average processing time for each instance update. Note here the multiple incremental learning is equivalent to single instance incremental learning in Laskov *et al.* (2006); Lin *et al.* (2015) when cache size is equal to 1. The experimental results of the cache size effect are shown in Figure 6.15. As depicted in Figure 6.15(a)(b), MIDL has negligible effect on frame-level EER and pixel-level RD. On the other hand, although the average processing time for each dataset in Figure 6.15(c) is different from each other depending on the size of dataset, it constantly decreases with the cache size increasing from 1 to 20, and saturates from 30 to 150.

The overall computational time on UCSD and UMN dataset are shown in Table 6.4. In Ped1 dataset, the difference on testing time between MIDL and MIDL-KF are

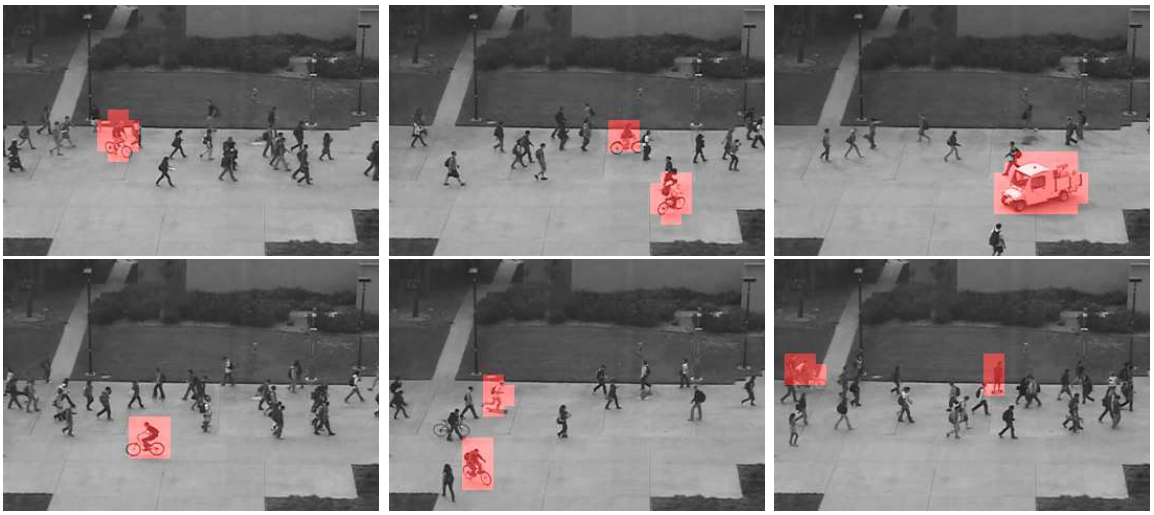


Figure 6.12: Detection results of MIDL-KF on Ped2. Best viewed in colour.

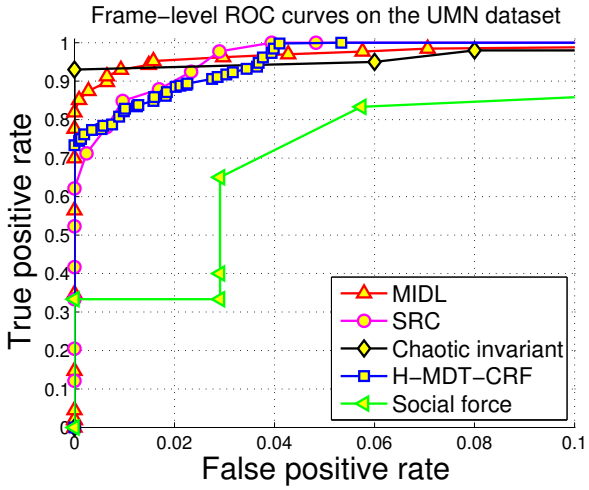


Figure 6.13: Frame-level ROC curves for UMN data.

negligible. However, the training time has increased from 0.06 second/frame without KF, to 0.25 second/frame with KF, which requires more iterations to converge. The average computation time of our approach is 0.15 second/frame for the UCSD Ped1 dataset, 0.08 second/frame for the UCSD Ped2 dataset, and 0.12 second/frame for the UMN dataset. Comparison with the computational time on UCSD Ped1 dataset, our approach is much faster than MDT (Mahadevan *et al.*, 2010), H-MDT-CRF (Li *et al.*, 2014), Video Parsing (Antić and Ommer, 2011), and SRC (Cong *et al.*, 2013), among which the best computational time is about 1.0 second/frame (H-MDT-CRF), with the exception of the extremely fast SCL (0.007 second/frame). However, note that SCL resizes the frame resolution from 158×238 to 120×160, which reduces computational cost significantly.

Table 6.3: Frame-level AUC and RD comparison for UMN data.

	Frame-level (%)	
Method	**AUC**	**RD**
Social force (Mehran *et al.*, 2009)	94.9	87.4
Chaotic invariant (Wu *et al.*, 2010)	99.4	94.7
SRC (Cong *et al.*, 2013)	99.6	97.2
H-MDT-CRF (Li *et al.*, 2014)	99.5	96.3
MIDL	**99.5**	**96.5**

(a) (b) (c)

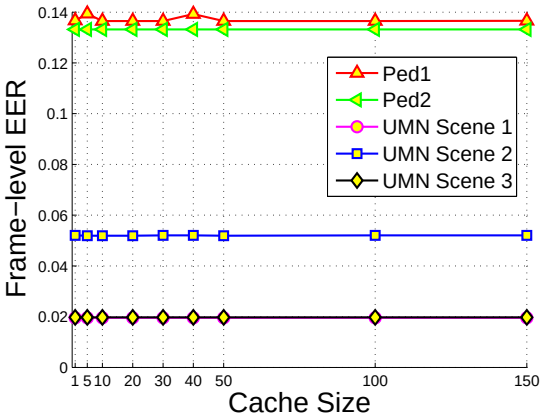
Figure 6.14: Detection results of the MIDL on the UMN dataset: (a) Scene 1, (b) Scene 2, and (c) Scene 3. Best viewed in colour.

6.4 Performance improvement by MTT

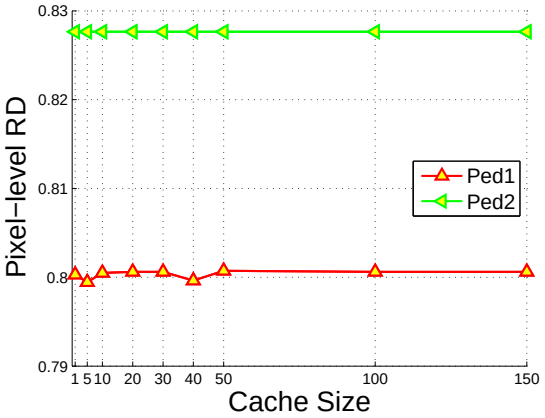
Although we have detected abnormal ROIs in each frame, there are still some false and missing detections due to the high occlusion of moving objects. Furthermore, to carry out further analysis, we would like to track an abnormal event from its appearance until end. With the hypothesis that an abnormal event is supposed to appear in a continuous spatial-temporal locations, we establish correspondence of abnormal ROIs between frames by using a simplified linear Multiple Target Tracking (MTT) algorithm

Table 6.4: Computational time on UCSD and UMN dataset (second/frame).

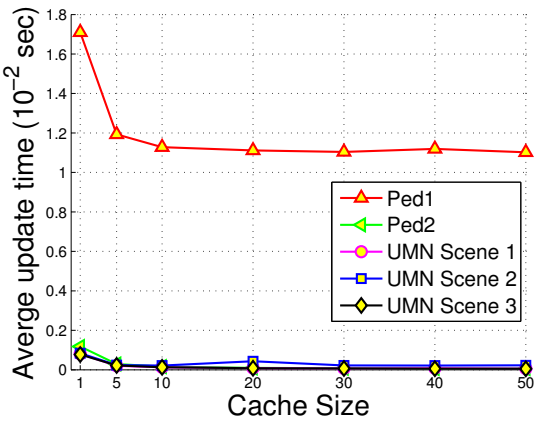
	MIDL			MIDL-KF		
Dataset	Training	Testing	Avg.	Training	Testing	Avg.
Ped1	0.06	0.05	0.06	0.25	0.06	0.15
Ped2	0.08	0.10	0.09	0.08	0.09	0.08
UMN	0.08	0.13	0.12	-	-	-



(a)



(b)



(c)

Figure 6.15: The effect of cache size on detection performance. (a) Frame-level EER comparison with various cache size. (b) Pixel-level RD comparison with various cache size. (c) The average update time with different cache size.

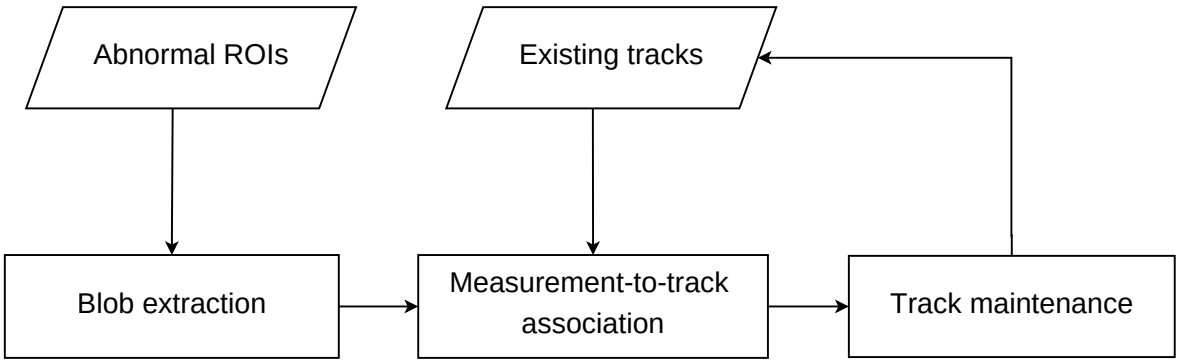


Figure 6.16: MTT flowchart.

(Blackman, 2004; Amditis *et al.*, 2012).

Figure 6.16 displays the flowchart of MTT. At each frame, we have an available set of existing tracks and a new set of blobs, which are extracted from the foreground pixels of the detected abnormal ROIs by using blob extraction. These blobs, also called measurements, can either be assigned to an existing track or initiates a new track by measurement-to-track association scheme. Track maintenance provides the capability of track initialization, continuation, and deletion, where the existing tracks are used for measurement-to-track association of next frame.

6.4.1 Blob extraction

The foreground pixels are labeled from all the detected abnormal ROIs, where a set of connected blobs are extracted by a connected-component labeling algorithm, e.g., Suzuki *et al.* (2003). As the background subtraction procedure has been employed in ROIs extraction (Section 6.2.1), there isn't any extra computation. However, the benefits of background subtraction are twofold. First, false detections due to small movement are removed. Second, it improves detection accuracy by filtering background pixels. Figure 6.17 illustrates the blob extraction process.

6.4.2 Measurement-to-track association

To clarify the procedure of MTT, at time t, let $\mathcal{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_m\}$ be the set of existing tracks, where the state of each track is modelled by a Kalman filter (Kalman, 1960; Welch and Bishop, 1995), denoted as $\mathbf{t}_i = (\mathbf{x}_{i,t-1}, P_{i,t-1}, A, H, Q, R)$, given the notation in Table 6.5. Meanwhile, let $\mathcal{Z} = \{\mathbf{z}_{1,t}, \ldots, \mathbf{z}_{n,t}\}$ be the centroids of the blobs.

For an existing track $\mathbf{t}_i$, we first predict its new state $\hat{\mathbf{x}}_{i,t}$ and state covariance $\hat{P}_{i,t}$

(a) (b) (c)

Figure 6.17: Blob extraction employed on the foreground pixels of the detected abnormal ROIs. (a) Abnormal ROIs. (b) Foreground. (c) Result after blob extraction. Best viewed in colour.

Table 6.5: Notation of the Kalman filter

Symbol	Description
$\mathbf{x}_{i,t}$	Estimated state of track i at time t
$\hat{\mathbf{x}}_{i,t}$	Predicted state of track i at time t
$\mathbf{z}_{j,t}$	The centroid of blob j at time t
$P_{i,t}$	Estimated state covariance of track i at time t
$\hat{P}_{i,t}$	Predicted state covariance of track i at time t
K	Kalman gain
A	State transition matrix, constant
H	Measurement matrix, constant
Q	Process noise covariance, constant
R	Measurement noise covariance, constant
I	Identity matrix

using the estimated state $\mathbf{x}_{i,t-1}$ and estimated state covariance $P_{i,t-1}$ at time $t-1$:

$$\begin{aligned} \hat{\mathbf{x}}_{i,t} &= A\mathbf{x}_{i,t-1}, \\ \hat{P}_{i,t} &= AP_{i,t-1}A^T + Q. \end{aligned} \tag{6.15}$$

Next, a distance matrix D is constructed, where D_{ij} measures the squared Mahalanobis distance (McLachlan, 1999) between $\hat{\mathbf{x}}_{i,t}$ and $\mathbf{z}_{j,t}$:

$$D_{ij} = (\mathbf{z}_{j,t} - H\hat{\mathbf{x}}_{i,t})^T (H\hat{P}_{i,t}H^T + R)^{-1}(\mathbf{z}_{j,t} - H\hat{\mathbf{x}}_{i,t}). \tag{6.16}$$

A track $\mathbf{t}_i$ is the candidate to be associated with a blob $\mathbf{z}_{j,t}$ only if $D_{ij} \leq G$, where G is the gate size. In other words, only the blobs within the gate are considered to be assigned to that track.

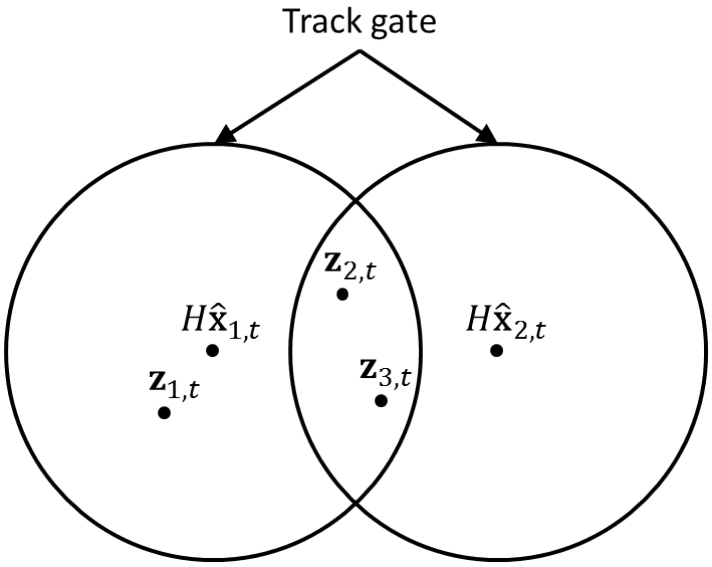


Figure 6.18: Illustration of 1-NN scheme. At time t, there are two tracks $\mathbf{t}_1$ and $\mathbf{t}_2$. The predicted position of track $\mathbf{t}_1$ is $H\hat{\mathbf{x}}_{1,t}$, while the predicted position of track $\mathbf{t}_2$ is $H\hat{\mathbf{x}}_{2,t}$. Based on the track gate, we associate $\mathbf{z}_{1,t}$, $\mathbf{z}_{2,t}$, and $\mathbf{z}_{3,t}$ to $\mathbf{t}_1$, while $\mathbf{z}_{2,t}$ and $\mathbf{z}_{3,t}$ to $\mathbf{t}_2$. Using the GNN scheme, $\mathbf{z}_{1,t}$ is assigned to $\mathbf{t}_1$, while $\mathbf{z}_{3,t}$ is assigned to $\mathbf{t}_2$.

However, there are multiple blobs within the gate of a track when some targets are close to each other. Under this circumstances, we simply adopt the 1-NN scheme to assign the nearest blob to the track. Figure 6.18 displays the 1-NN scheme. It works well in our approach due to the sparse distributed abnormal events and the proposed robust anomaly detector. However, Multiple Hypothesis Tracking (Reid, 1979; Cox and Hingorani, 1996; Kim *et al.*, 2015) may be considered under some more complicated environments.

6.4.3 Track maintenance

Tracks are maintained as follows:

1. Track initialization: a blob isn't assigned to any existing track, it initiates a new track.

2. Track continuation:

   (a) A blob $\mathbf{z}_{j,t}$ is assigned to an existing track $\mathbf{t}_i$ at time t, the track is updated

as:

$$
\begin{aligned}
K &= \hat{P}_{i,t} H^T (H \hat{P}_{i,t} H^T + R)^{-1}, \\
\mathbf{x}_{i,t} &= \hat{\mathbf{x}}_{i,t} + K(\mathbf{z}_{j,t} - H\hat{\mathbf{x}}_{i,t}), \\
P_{i,t} &= (I - KH)\hat{P}_{i,t},
\end{aligned}
\tag{6.17}
$$

(b) No blob is assigned to an existing track, the track is updated as:

$$
\begin{aligned}
\mathbf{x}_{i,t} &= \hat{\mathbf{x}}_{i,t}, \\
P_{i,t} &= \hat{P}_{i,t},
\end{aligned}
\tag{6.18}
$$

which provides a level of support for temporary occlusion.

3. Track deletion:

(a) It is invisible for a certain number of frames (e.g., 20 frames in our experiments), i.e., disappear.

(b) It appears in a very short time, i.e., noise.

6.4.4 Evaluation

Setup

We have conducted experimental results on UCSD dataset to evaluate the performance of MTT. The MTT algorithm is employed on the optimal result of MIDL-KF, which is referred as MIDL-KF-MTT. The parameters A, H, Q, and R for Kalman filter is predefined and given by:

$$
A = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}
$$

$$
Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 25 \end{bmatrix}
$$

$$R = \begin{bmatrix} 15 & 0 \\ 0 & 15 \end{bmatrix}$$

For a new blob $\mathbf{z}_{i,t}$, which is unable to assigned an existing track, we initiate a track as follows. Let $\mathbf{z}_{i,t} = \{x_{i,t}, y_{i,t}\}$ be the centroid of the blob, where $x_{i,t}$ and $y_{i,t}$ correspond to its spatial location in frame. Then, the estimated state $\mathbf{x}_{i,t}$ and estimated state covariance $\mathbf{P}_{i,t}$ are initialized as:

$$\mathbf{x}_{i,t} = \begin{bmatrix} x_{i,t} \\ 0 \\ y_{i,t} \\ 0 \end{bmatrix}$$

$$\mathbf{P}_{i,t} = \begin{bmatrix} 200 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix}$$

To filter out noise, we only keep the tracks whose length is greater than 20 and the proportion of visible frames in a track is greater than 0.6.

Further experiments

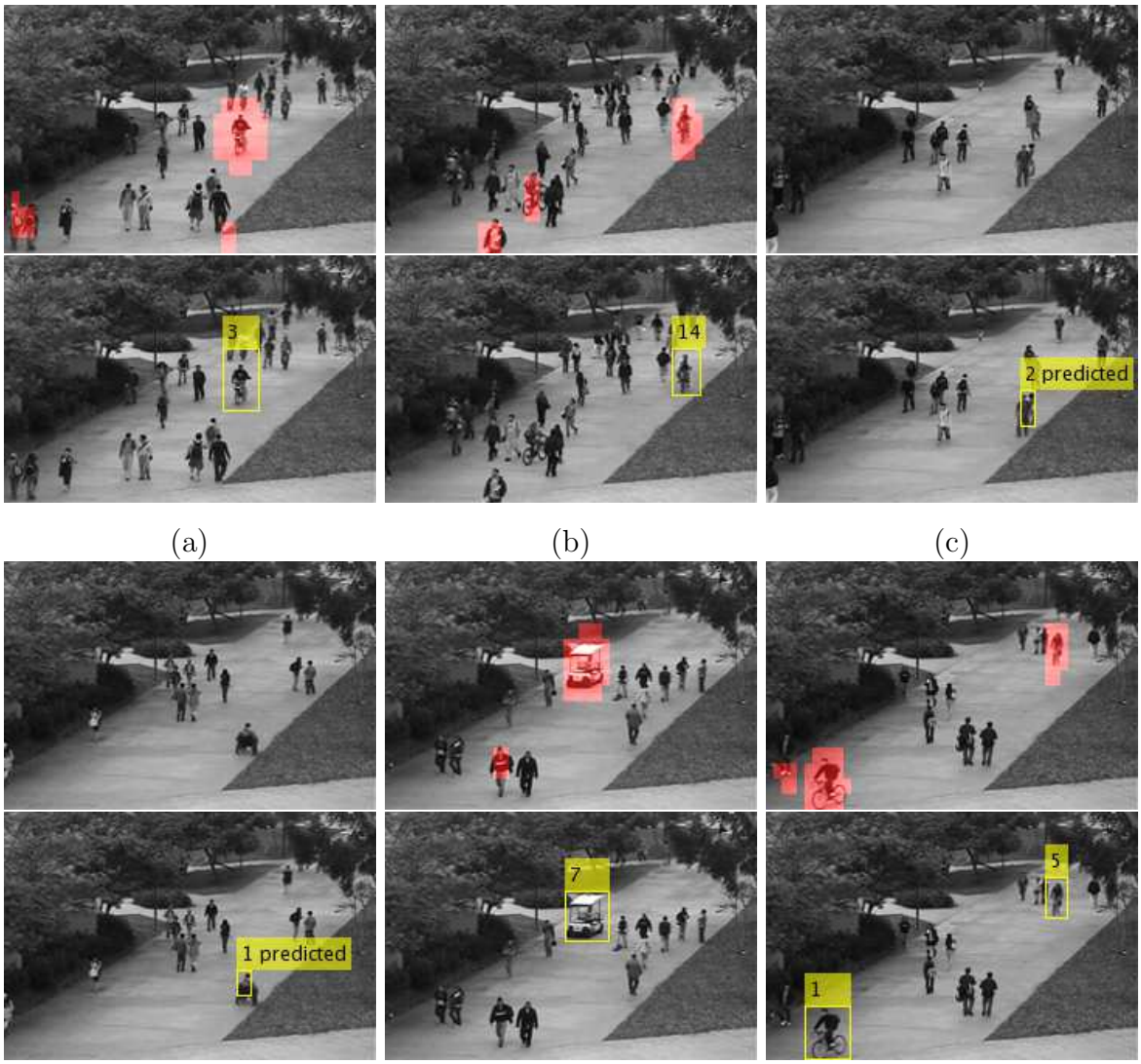
Figure 6.19 and Figure 6.20 illustrate the detection result comparison between MIDL-KF and MIDL-KF-MTT on UCSD data. As can be seen, MIDL-KF-MTT improves the detection results on several aspects. First, with the help of background subtraction and blob extraction, MIDL-KF-MTT displays more accurate anomaly localization than MIDL-KF. Besides, MTT algorithm not only filters out noise (e.g., Fig. 6.19(a)(b)) by removing short tracks, it also has the capability to support temporal occlusion or missing detection; for instance, in Fig. 6.19(c), where the abnormal target disappears in a few frames due to occlusion. This improvement, however, is unable to be measured based on the evaluation methodology proposed by Li *et al.* (2014).

6.5 Summary

In this chapter, we have proposed an improved framework to address the problem of AED in video surveillance. To handle the streaming video data we have employed a MIDL algorithm for OCSVM, which is reinforced by the adoption of an adaptive

feature descriptor, and a kernel fusion scheme that combines motion and spatial information together. The outcome is competitive when compared with the state-of-the-art approaches, and much improved computational efficiency compared with our previous online adaptive OCSVM algorithm. An additional benefit is that our method can handle both motion and contextual anomalies. Furthermore, the final detection results have been improved by MTT.

The MIDL provides an efficient approach to update OCSVM in an online fashion, in the next chapter, we will apply it to another video content analysis task - shot boundary detection.



(a) (b) (c)

(d) (e) (f)

Figure 6.19: Detection result comparison of MIDL-KF and MIDL-KF-MTT on Ped1 dataset. In each sub-figure, the top row is the result detected by MIDL-KF, while the bottom row is the result detected by MIDL-KF-MTT. Best viewed in colour.

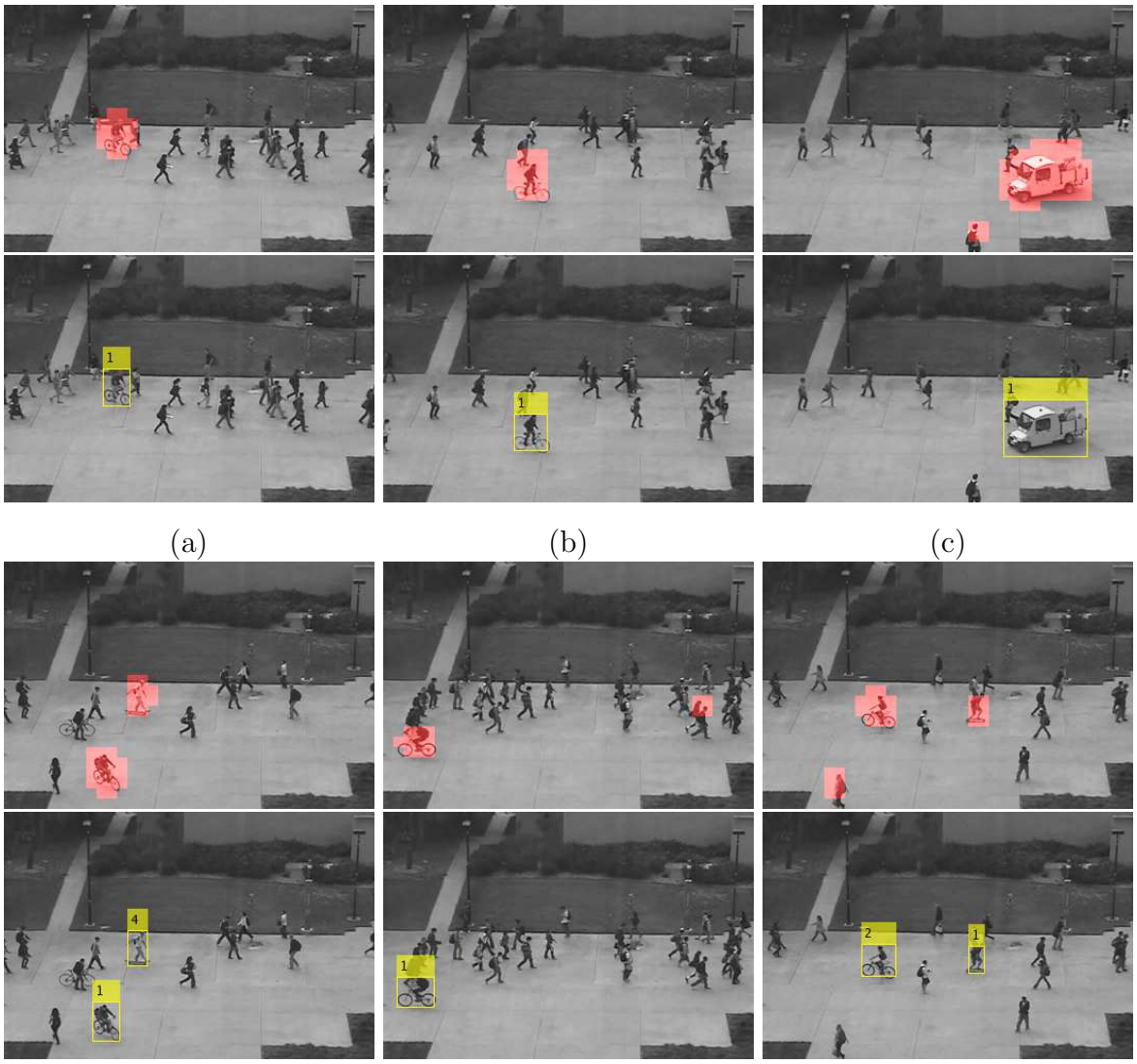


Figure 6.20: Detection result comparison of MIDL-KF and MIDL-KF-MTT on Ped2 dataset. In each sub-figure, the top row is the result detected by MIDL-KF, while the bottom row is the result detected by MIDL-KF-MTT. Best viewed in colour.

Chapter 7

MIDL OCSVM for Shot Boundary Detection

7.1 Overview

A video shot, which represents a continuous action in time and space, is composed of a series of related, consecutive frames taken contiguously by a single camera (Hanjalic, 2002). Both pre-edited and unedited video footages may contain shots, and partitioning a video into shots automatically is the fundamental prerequisite for further video content analysis, editing, browsing and retrieval applications, a term referred to as Shot Boundary Detection (SBD). The shot boundary is divided into two types: abrupt change (or hard cut) and gradual transition. While abrupt change is the concatenation of two shots directly, gradual transition includes an artificial shot transition effect between two shots, which could last for a few frames. The gradual transition consists of dissolve, fade in/out, wipe, etc.

During the last decade, a broad spectrum of approaches (Fang *et al.*, 2006; Chen *et al.*, 2011; Mohanta *et al.*, 2012; Kowdle and Chen, 2012; Lu and Shi, 2013; Jiang *et al.*, 2013; Lakshmi Priya and Domnic, 2014) for SBD have been proposed. The simplest approach to detect a shot boundary is to analyze the difference between two successive frames (Huang and Liao, 2001). This is straightforward to implement and is effective to detect abrupt changes. However, it is sensitive to noise such as flashlight frames, and it cannot detect gradual transitions because the corresponding pair-wise frame difference is rather small. Automatic thresholding is employed in Kowdle and Chen (2012), where changes on optical flows of frames within a sliding window are thresholded by a value that is equal to the change median plus two times of their standard deviation. In

Yuan *et al.* (2007), the strength of using graph partition for SBD is discussed. To deal with varying characteristics of videos that challenge the threshold setting, a SVM based approach is adopted to treat the SBD as a classification problem. This approach however has two deficiencies: firstly the requirement on frame-by-frame ground-truth data for training, and secondly, the costly training time because of the use of SVM. In Chen *et al.* (2011), decision trees and a finite state machine are adopted for abrupt cut detection and dissolving detection respectively for compressed MPEG videos. A novel feature extraction process using the Walsh-Hadamard Transform (WHT) kernel is proposed in Lakshmi Priya and Domnic (2014). A promising performance is achieved however at the high computing costs resulting from the sophisticated WHT kernel. Another recent work (Lu and Shi, 2013) adopts a candidate segment and singular value decomposition to cut down the processing time.

In this chapter, we propose a novel framework for SBD by means of MIDL OCSVM framework. We regard a shot boundary as an anomaly in temporal space in a video, thus our previous work can be extended to this scenario smoothly. However, due to the various shot boundary types, we cannot apply OCSVM to SBD directly as it is sensitive to object motion or flashlight frames. Fortunately, with the hypothesis that most of the frames in one shot is supposed to be different from those in another shot, we propose to measure the divergence between two OCSVM classifiers, which are learnt from two contextual sets, i.e., Immediate Past Set (IPS) and Immediate Future Set (IFS).

The proposed framework thus has the following advantages:

- We present a unified framework to detect different types of shot boundary rather than adopting a set of classifiers for each specific shot boundary type. This simplifies the computational complexity of the approach while maintaining a high accuracy.

- Our approach, which inherits the advantages of OCSVM, is robust to noise such as abrupt illumination changes and large object or camera movements.

- MIDL updates OCSVM classifier in an online fashion, requiring very low computational cost.

7.2 The computational framework

The flowchart is given in Figure 7.1. Given the IPS and IFS with fixed length M at time t, we first extract corresponding feature descriptor for each frame in each set, where an OCSVM classifier is trained respectively. We measure the divergence between the two sets based on the concept of OCSVM. From t to $t+1$, we add a new frame and remove the oldest one in each set while updating the classifier using the MIDL algorithm, with both incremental and decremental learning involved. Shot boundaries are detected based on divergence output.

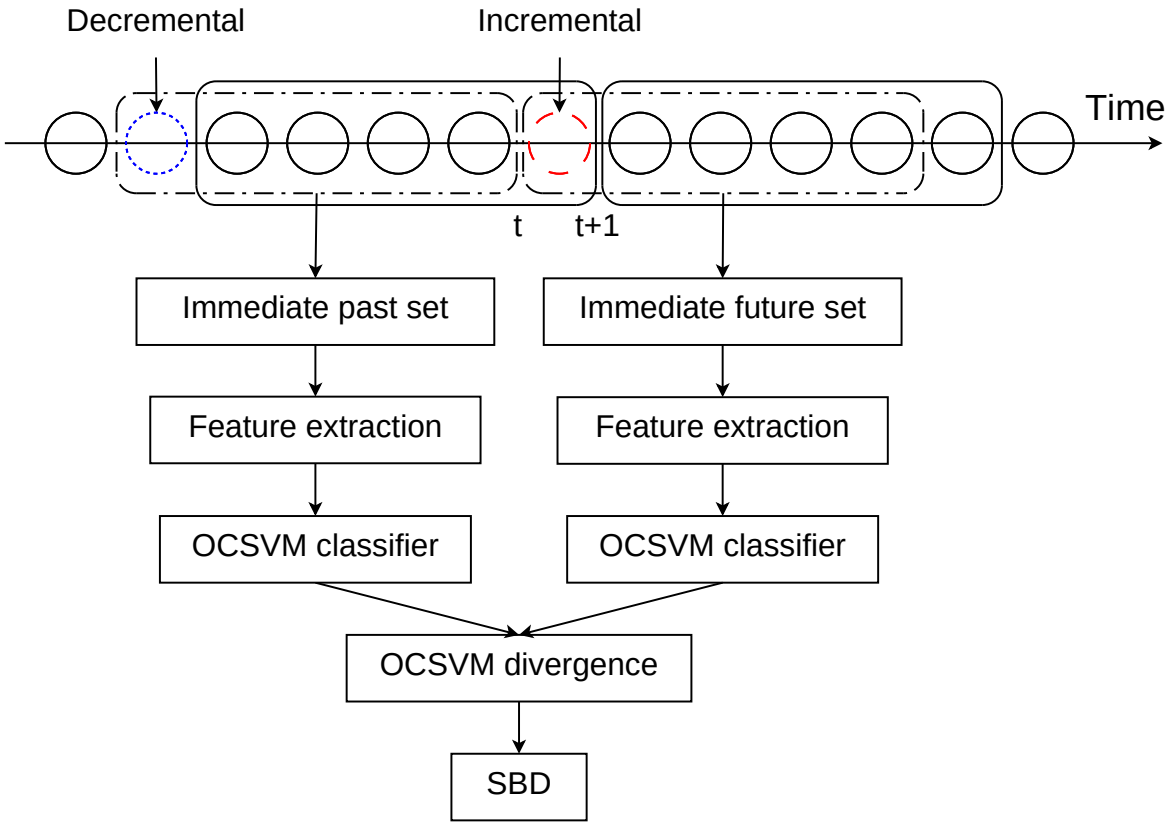


Figure 7.1: The flowchart of SBD using OCSVM divergence.

7.2.1 Feature extraction and kernel selection

Following Yuan *et al.* (2007), we adopt a block-based histogram as our feature representation. Specifically, a frame is divided into $2^L \times 2^L$ blocks, where L is the granularity level, $L = 0, 1, \ldots, l$. In each block a 48-bin (16 bins for each of RGB channels) colour histogram is calculated, and the frame is represented by concatenating colour histograms from all blocks. We chose this feature for several reasons. First, it is simple to

implement and has little computational cost. Secondly, it provides a trade-off between sensitivity and invariance. Note that we have also investigated other feature schemes, e.g., pixel-based (Huang and Liao, 2001), but the performance is worse than that of the block-based histogram, so we will not report the results.

Considering the histogram-based representation of our feature descriptor, we adopt the Histogram Intersection kernel (Swain and Ballard, 1991) as the similarity measure between two inputs:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_b \min(\mathbf{x}_i^b, \mathbf{x}_j^b), \tag{7.1}$$

where b indicates the corresponding bin in $\mathbf{x}_i$ and $\mathbf{x}_j$.

7.2.2 OCSVM Divergence

To measure the divergence between two OCSVM classifiers, we analyse the representation of OCSVM in the feature space $\mathcal{F}$. Using $\mathbf{v}_i$ to denote $\Phi(\mathbf{x}_i)$, for any $\mathbf{v}_i$ we have $\|\mathbf{v}_i\| = k(\mathbf{x}_i, \mathbf{x}_i) = 1$. In other words, the training set X are all mapped on a hypersphere S with origin of $\mathbf{o}$ and radius $r = 1$. According to the definition of OCSVM in Section 5.2, the OCSVM in $\mathcal{F}$ corresponds to finding the optimal hyperplane $\mathbf{w}$ such that most mapped training set $\mathbf{v}_i$ have $\mathbf{w} \cdot \mathbf{v}_i - \rho > 0$ - as shown in Figure 7.2. Here the optimal hyperplane corresponds to maximizing the margin $\rho/\|\mathbf{w}\|$, namely, the distance from $\mathbf{o}$ to $\mathbf{w}$. The same as in Section 5.3.2, there are three types of different vectors: margin support vectors, which are located on the edge of the segment; error support vectors, which are located outside of the segment; and other support vectors, which are located in the segment. On the basis of the definition, although the composition of three different vectors between two OCSVM classifiers is different from each other, they are in the same feature space and are comparable by computing the divergence of their formed segments.

Let $B_1^t = \{\mathbf{x}_{t-m+1}, \mathbf{x}_{t-m+2}, \ldots, \mathbf{x}_t\}$ and $B_2^t = \{\mathbf{x}_{t+1}, \mathbf{x}_{t+2}, \ldots, \mathbf{x}_{t+m}\}$ be the IPS and IFS at time t, where the length of both sets are M. If there is a shot boundary, their segments in $\mathcal{F}$ should be different from each other. In other words, the size as well as location of the two segments are different from each other. Figure 7.3 shows a simple example of our approach. To this end, we employ a divergence function based on the work of Desobry *et al.* (2005), which has been deployed to detect abrupt change of music signals.

Let $\mathbf{c}_1^t$ and $\mathbf{c}_2^t$ be the centre points of segments learnt from B_1^t and B_2^t respectively, and $\mathbf{p}_1^t$ and $\mathbf{p}_2^t$ be arbitrary points residing on the boundary of their corresponding

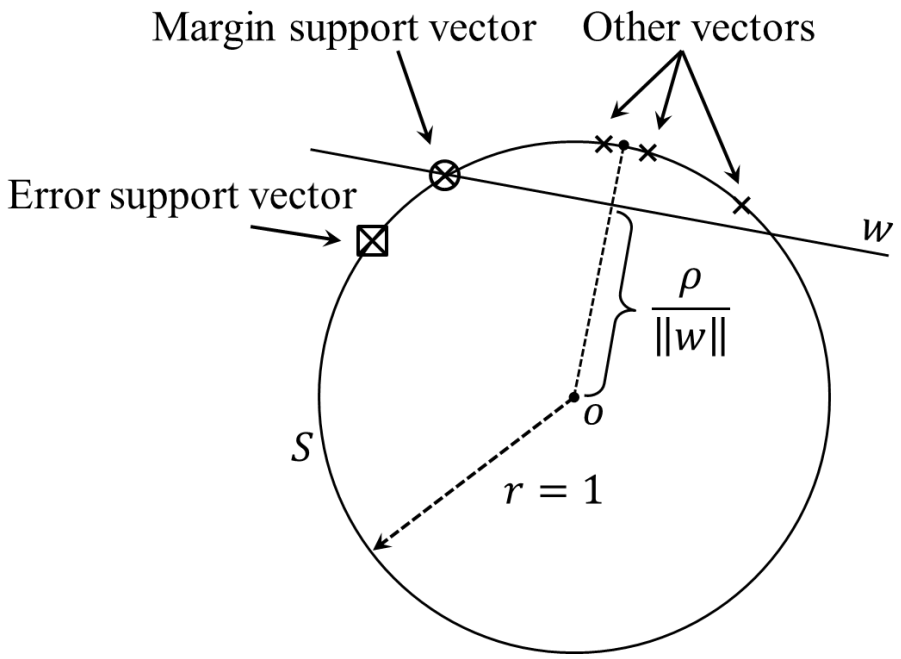


Figure 7.2: In the feature space $\mathcal{F}$, OCSVM aims to find the smallest segment while enclosing the mapped training set $\mathbf{v}_i$ as many as possible, i.e., maximizing the margin $\rho/\|\mathbf{w}\|$.

segments, as shown in Figure 7.4. The divergence function D between two OCSVMs, trained on B_1^t and B_2^t respectively, is given as:

$$D_{\text{OCSVM}}(B_1^t, B_2^t) = \frac{\overset{\frown}{\mathbf{c}_1^t\mathbf{c}_2^t}}{\overset{\frown}{\mathbf{c}_1^t\mathbf{p}_1^t} + \overset{\frown}{\mathbf{c}_2^t\mathbf{p}_2^t}}, \tag{7.2}$$

where $\overset{\frown}{\mathbf{c}_1^t\mathbf{c}_2^t}$ is the arc distance from $\mathbf{c}_1^t$ to $\mathbf{c}_2^t$, and $\overset{\frown}{\mathbf{c}_1^t\mathbf{p}_1^t}$ ($\overset{\frown}{\mathbf{c}_2^t\mathbf{p}_2^t}$) is the arc distance from $\mathbf{c}_1^t$ ($\mathbf{c}_2^t$) to $\mathbf{p}_1^t$ ($\mathbf{p}_2^t$). Eq. (7.2) indicates the divergence is large if two segments are well separated, whereas it is small for strongly overlapped segments.

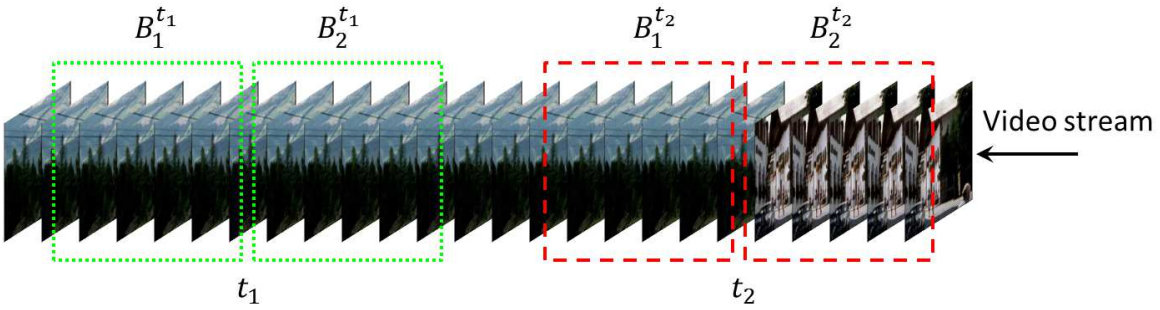
However, we cannot calculate Eq. (7.2) directly in the feature space because we do not know the explicit representation of the feature map $\Phi(\mathbf{x})$. The calculation of D however can be done in the kernel space. Specifically, for any two points $\mathbf{a}$ and $\mathbf{b}$ lying on an arbitrary sphere, the arc distance is given by:

$$\overset{\frown}{\mathbf{ab}} = r\theta, \tag{7.3}$$

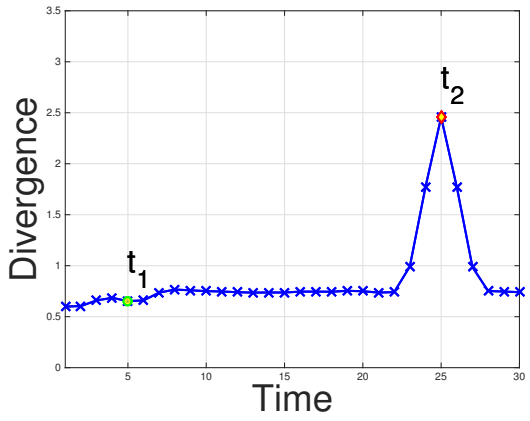
where r is the radius, and θ is the central angle between $\mathbf{a}$ and $\mathbf{b}$.

Meanwhile, the dot product between vector $\mathbf{a}$ and vector $\mathbf{b}$ is given by:

$$\mathbf{a} \cdot \mathbf{b} = \|\mathbf{a}\|\|\mathbf{b}\| \cos\theta, \tag{7.4}$$



(a)



(b)

Figure 7.3: A simple example of our proposed SBD algorithm. At time t_1, $B_1^{t_1}$ and $B_2^{t_1}$ (green dotted rectangles in (a)) are in the same shot, the divergence (green circle in (b)) between them is low. At time t_2, there is a shot boundary between $B_1^{t_2}$ and $B_2^{t_2}$ (red dashed rectangles in (a)), hence the divergence (red diamond in (b)) is high. Best viewed in colour.

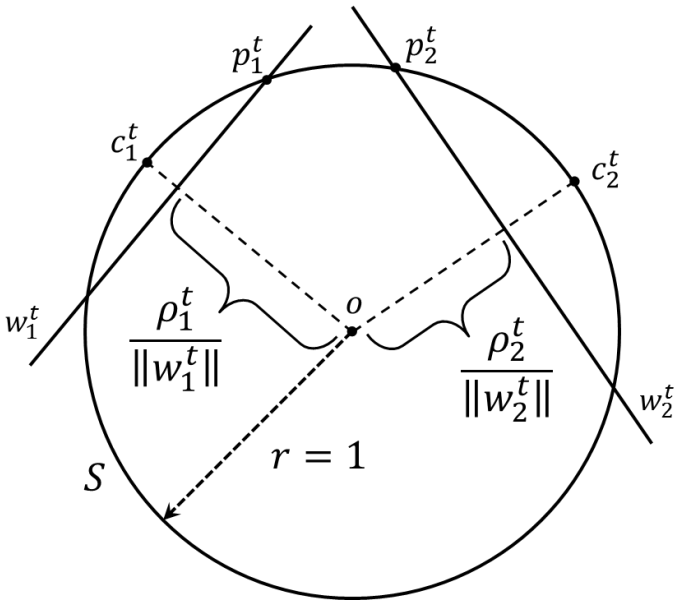


Figure 7.4: Two OCSVM classifiers in $\mathcal{F}$, where $\mathbf{w}_1^t$ and $\mathbf{w}_2^t$ are the optimal hyperplanes of the immediate past set B_1^t and the immediate future set B_2^t. This situation corresponds to a shot boundary as both segments get separated from each other.

where $\|\mathbf{a}\| = \|\mathbf{b}\| = r$.

Combining Eq. (7.3) and Eq. (7.4) together with r equal to 1, we have:

$$\widehat{\mathbf{ab}} = \arccos(\mathbf{a} \cdot \mathbf{b}). \tag{7.5}$$

Therefore, computing $\widehat{\mathbf{ab}}$ is simplified to find coordinates of $\mathbf{a}$ and $\mathbf{b}$ that reside on the sphere.

As seen in Figure 7.4, based on the concept of OCSVM, we know line $\mathbf{oc}_1^t$ is perpendicular to $\mathbf{w}_1^t$, and $\mathbf{oc}_2^t$ to $\mathbf{w}_2^t$, so we can replace $\mathbf{c}_1^t$ ($\mathbf{c}_2^t$) with $\mathbf{w}_1^t/\|\mathbf{w}_1^t\|$ ($\mathbf{w}_2^t/\|\mathbf{w}_2^t\|$) after some geometric computations. The arc distance $\widehat{\mathbf{c}_1^t\mathbf{c}_2^t}$ therefore is calculated as:

$$\widehat{\mathbf{c}_1^t\mathbf{c}_2^t} = \arccos\left(\frac{\mathbf{w}_1^t \cdot \mathbf{w}_2^t}{\|\mathbf{w}_1^t\|\|\mathbf{w}_2^t\|}\right). \tag{7.6}$$

By employing the kernel transform, we have:

$$\widehat{\mathbf{c}_1^t\mathbf{c}_2^t} = \arccos\left(\frac{(\boldsymbol{\alpha}_1^t)^T k_{12}^t \boldsymbol{\alpha}_2^t}{\sqrt{(\boldsymbol{\alpha}_1^t)^T k_{11}^t \boldsymbol{\alpha}_1^t}\sqrt{(\boldsymbol{\alpha}_2^t)^T k_{22}^t \boldsymbol{\alpha}_2^t}}\right), \tag{7.7}$$

where $\boldsymbol{\alpha}_1^t$ and $\boldsymbol{\alpha}_2^t$ are the coefficient sets of the OCSVM classifier learnt from B_1^t and B_2^t respectively. k_{12}^t is the kernel matrix, where its rows are indexed by B_1^t, and its columns are indexed by B_2^t.

Similarly, the arc distance $\mathbf{c}_1^t\widehat{\mathbf{p}_1^t}$ ($\mathbf{c}_2^t\widehat{\mathbf{p}_2^t}$) is given as:

$$\mathbf{c}_i^t\widehat{\mathbf{p}_i^t} = \arccos\left(\frac{\rho_i^t}{\sqrt{(\boldsymbol{\alpha}_i^t)^T k_{ii}^t \boldsymbol{\alpha}_i^t}}\right),\ i = 1, 2. \tag{7.8}$$

Compared with the traditional approach of comparing features of successive frames directly, assessing the divergence between two OCSVMs trained on frame sets gives two potential advantages. First, it is more robust to noises such as flashlight frames, as these will be regarded as error support vectors and will not affect the OCSVM classifiers. Second, it is effective to detect gradual transitions, e.g., dissolve, and wipe etc. Even though the difference between two consecutive frames is not significant in these gradual transitions, the divergence between two sets is supposed to be large as the overall distribution will be quite different.

7.3 Performance evaluation

7.3.1 Experimental results

We have carried out experiments on the TRECVID 2007 SBD dataset. The process of performance evaluation is as follows: we first compute the divergence output for each video sequence, then we tune the threshold on the divergence to identify shot boundaries.

To obtain the optimal performance, we first evaluate the impact of parameters on SBD performance. Three parameters have to be evaluated, namely, parameter C of OCSVM, set length M, and the granularity level L. The optimal settings are acquired by fine-tuning one parameter while fixing the rest of them. The impact of C on performance is displayed in Figure 7.5(a), where M and L are set as 20 and 2 randomly. It shows that the OCSVM classifier with C of 0.2 achieves the best performance. With C of 0.2 and L of 2, Fig 7.5(b) illustrates the impact of M, where 20 attains the best recall and precision. Figure 7.5(c) shows the influence of different L, where C and M are set as 0.2 and 20. It is improved with L increasing from 0 to 3, and saturates when L enlarges from 3 to 4. Therefore, we report our final results and compare with the start-of-the-art approaches using the aforementioned optimal settings, i.e., $C = 0.2$, $M = 20$, and $L = 3$.

Our final SBD results using OCSVM divergence are reported in Table 7.1, achieving **92.0**% on the F_1 metric. It is comparable to Kawai *et al.* (2007) (92.4%), Ren *et al.* (2007) (93.0%), better than Mühling *et al.* (2007) (91.9%), Zhao *et al.* (2007) (90.6%),

Table 7.1: Performance comparison on TRECVID 2007 SBD dataset.

Method	Recall (%)	Precision (%)	F_1 (%)	Running time (seconds)
Mühling *et al.* (2007)	93.1	90.7	91.9	7000
Zhao *et al.* (2007)	91.3	90.0	90.6	-
Kawai *et al.* (2007)	90.5	94.4	92.4	1697
Ren *et al.* (2007)	94.1	91.9	93.0	5185
Chen *et al.* (2011)	88.9	88.7	88.8	5358
Lakshmi Priya and Domnic (2014)	96.5	95.7	96.1	13800
OCSVM divergence	**90.7**	**93.4**	**92.0**	**4468**

and Chen *et al.* (2011) (88.8%), however, worse than Lakshmi Priya and Domnic (2014) (96.1%).

On the other hand, the overall running time our approach is 4468 seconds, faster than most of the state-of-the-art algorithms except Kawai *et al.* (2007). Specifically, using the same platform, i.e., MATLAB R2014a on a 2.7GHz Intel Core i5 and 8GB RAM, the speed has increased from 101 frames/second using batch mode OCSVM to 143 frames/second using MIDL, where the processing time for each OCSVM classifier training has decreased from 5.4×10^{-3} second to 2.5×10^{-3} second.

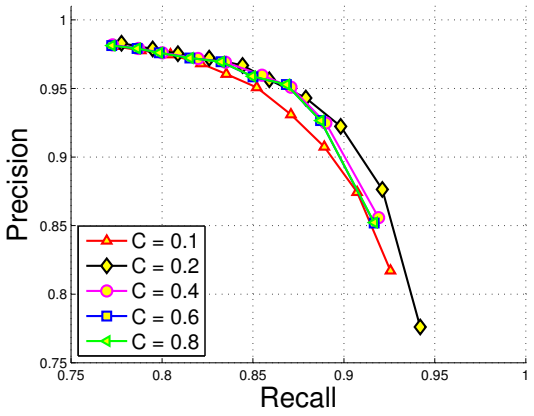
7.3.2 Robustness evaluation

Finally we examine the robustness the proposed OCSVM divergence in dealing with gradual transitions and noises. For this purpose two video clips each with a length of 100 frames are generated using the TRECVID data. The first testing clip has a hard-cut shot boundary at Frame 52, and three noise frames located at Frames 14, 34, and 78 respectively. The second clip includes two gradual transition shot boundaries: fade-out and fade-in. The fade-out spans from Frames 31 to 40, the fade-in from Frames 64 to 70. Three noise frames are also inserted, as Frames 12, 57, and 89 respectively. For demonstration purposes we only consider two approaches: the proposed OCSVM divergence using MIDL, and the pairwise consecutive frames dissimilarity using histogram intersection. Figure 7.6 and 7.7 illustrate the results on these two video clips. As can be seen from the results, the dissimilarity based on the successive frames is very sensitive to those noise frames, whereas our approach can effective detect both hard cuts and gradual transitions even though noise frames exist.

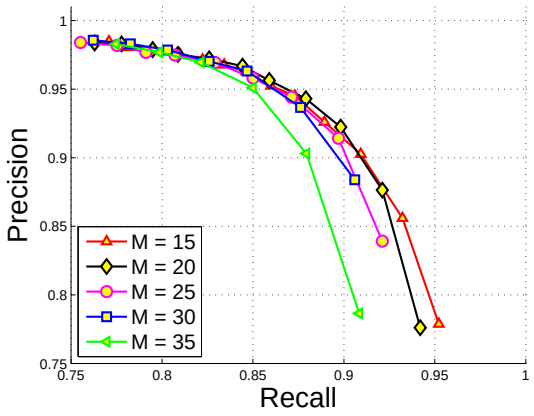
7.4 Summary

In the chapter we propose a novel approach to address the problem of SBD. Instead of comparing the difference between pair-wise consecutive frames at a specific time, we propose a divergence metric to measure the difference between two OCSVM classifiers, which are learnt from two contextual sets, i.e., immediate past set and immediate future set. One significant advantage is our approach is a unified framework to detect all types of shot boundaries. In addition, using the MIDL, we reduce the computational cost effectively. By inheriting the properties of OCSVM, our method is robust to noises while effective to gradual transitions. Experimental results on a challenging benchmark dataset exhibit the competitive performance of our approach compared with the state-of-the-art.

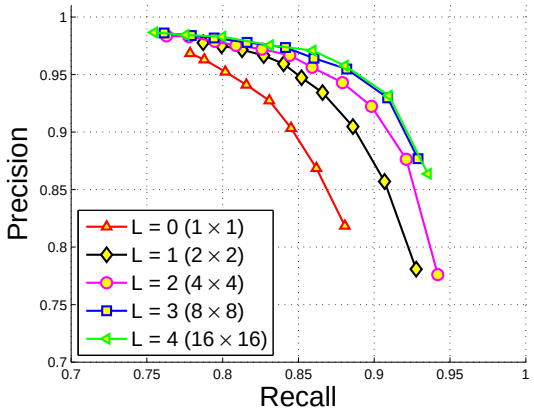
The future work of SBD is twofold. On the one hand, more datasets will be applied and more extensive experiments will be carried out. On the other hand, as the shot boundaries are identified exclusively by tuning the threshold on the divergence output for each video sequence, we will further analyse the divergence output to improve the SBD performance as well as classify the different types of boundaries.



(a)

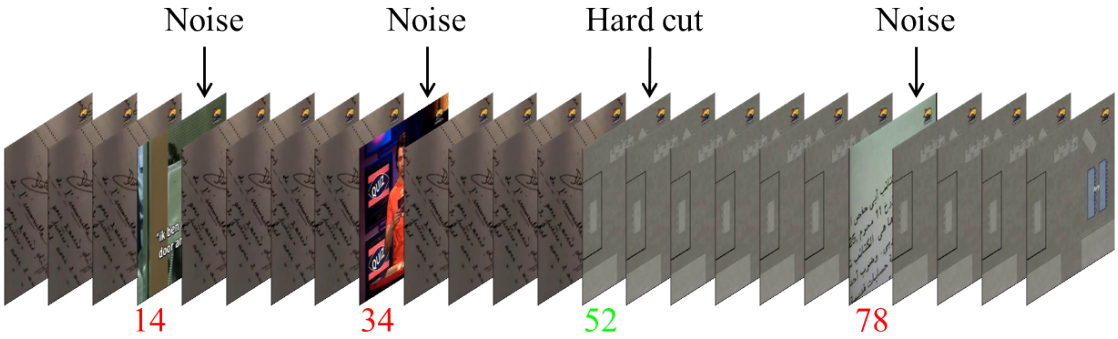


(b)

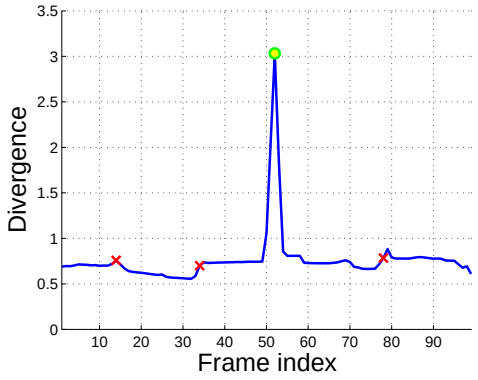


(c)

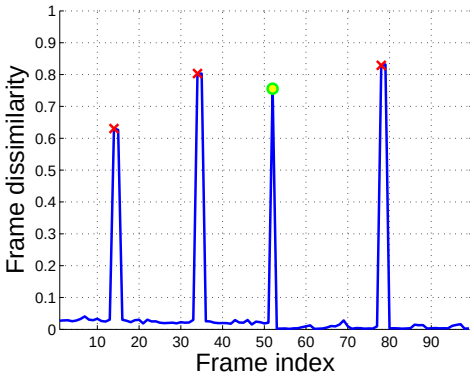
Figure 7.5: Parameter tuning on the TRECVID 2007 SBD dataset. (a) Different settings of C in OCSVM; (b) Different set length M; (c) Different granularity levels L.



(a)

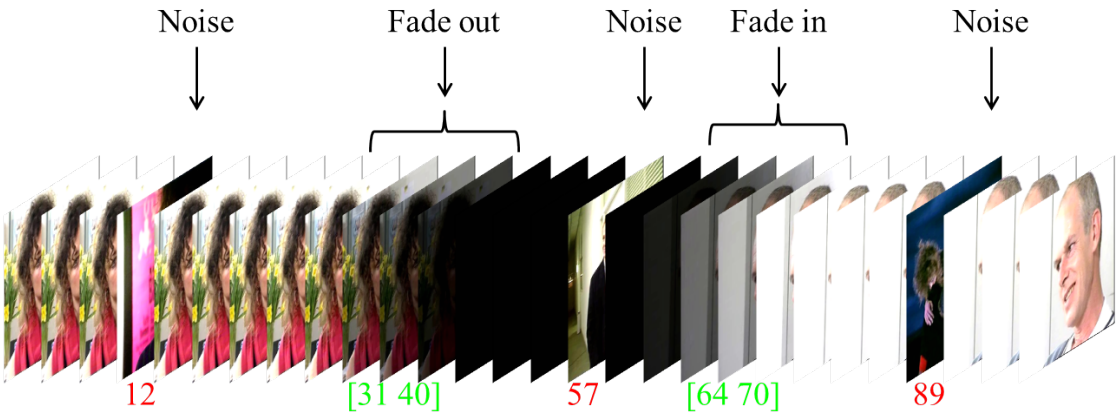


(b)

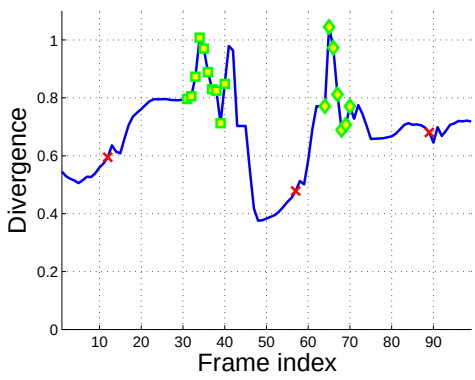


(c)

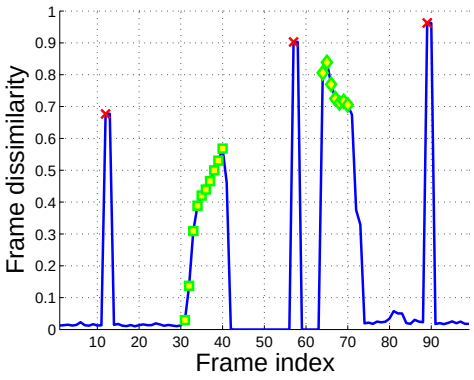
Figure 7.6: Testing scenario 1. (a) The testing video sequence with a hard-cut and three noise frames, locations marked; (b) Divergence output of MIDL, where the threshold for SBD is 0.905; (c) Pairwise consecutive frames dissimilarity output. The green circles and red crosses in (b)(c) indicate the hard-cut and noise frames respectively.



(a)



(b)



(c)

Figure 7.7: Testing scenario 2. (a) The testing video sequence with a fade-in, a fade-out and three noise frames, locations marked; (b) Divergence output of MIDL, where the threshold for SBD is 0.905; (c) Pairwise consecutive frames dissimilarity output. In (b)(c), the green squares and diamonds indicate fade-out and fade-in respectively, and red crosses indicate noise frames.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In recent years, the pervasive video surveillance systems deployed in crowded public places bring about increasing research interest in crowd scene analysis in video surveillance with the increasing importance of public security and safety. The general theoretical literature on this topic is inconclusive on several vital questions. This thesis aims to answer a few of these questions as follows:

1. Can we develop an efficient dimension reduction algorithm to consider both spatial and temporal similarities between frames when generating video manifolds?

2. Can we find a simple but efficient feature descriptor to capture the motion information in a crowded video surveillance stream?

3. Can an online learning algorithm be developed so that the learnt model can evolve on-the-fly from continuous data streams?

4. Combining the feature descriptor with the online learning algorithm, can we propose a unified framework for AED so that it meets the aforementioned challenges while providing real-time response?

We have answered these research questions either in a specific chapter, or in relevant chapters collectively, as follows:

- **Proposing a novel dimension reduction algorithm "STLPP" in Chapter 4.**
  To meet the challenge of the curse of dimensionality while considering the strong

correlation between frames in video surveillance, we propose the STLPP to generate video manifolds for crowd scene analysis. We introduce two metrics to measure the similarity between two frames when constructing the weight matrix. The former, called spatial similarity, measures the motion similarity using the optical-flow based descriptor. The latter, called temporal similarity, measures the temporal correlation between two frames. For two frames, the closer, the more similar. The overall similarity between two frames is the product of the spatial similarity and the temporal similarity. By incorporating both similarities, the generated video manifolds not only cluster similar crowd events together, but also explicitly display the temporal change of crowd events.

- **Proposing three different optical-flow based feature descriptors.**
  Three optical-flow based feature descriptors are proposed to model the motion information in crowd scenes. Firstly, in Chapter 4, a frame is divided into $m \times n$ regions, in each region a 8 bin HOF is computed. As a result, a frame is represented as a $8 \times m \times n$ dimensional histogram. Next, the QBC is proposed to increase the dissimilarity between motion patterns by applying adaptive quantization and binarization. Because the temporal information is incorporated in manifold learning algorithm, here the QBC represent exclusively the motion information in each frame. Secondly, in Chapter 5, we obtain a set of video segments using a sliding window, in which a set of video events are extracted. Non-overlapping spatial-temporal patches are densely sampled from each event and represented as HOG descriptors. A video event is represented as a histogram using the BoW model. Compared with the previous HOF descriptor, the HOF descriptor contains the crowd motion information in a number of continuous frames, thus more stable. Finally, to reduce computational cost, we optimize the feature extraction process in Chapter 6. Rather than extract a HOF descriptor from each spatial-temporal patch, the optical flow vectors in each video event are directly assigned to $M \times N$ bins, where M is number of directions, and N is the discretization level of motion intensities, determined by performing k-means clustering. Thus, a video event is specified by a simplified descriptor, called AMHOF and used for AED.

- **Developing an online adaptive learning of OCSVM for data stream learning in Chapter 5 and 6.**
  Because of the large size as well as the streaming nature of video data, we develop an online adaptive learning of OCSVM for data stream learning. The core part is

to deploy an incremental and decremental OCSVM within a sliding buffer. From time t to $t+1$, we add the new arriving data and remove the obsolete data to/from the buffer through the incremental and decremental procedures respectively. By solving a linear system and recursively constructing new exact solutions to keep the KKT satisfied for the updated data, this approach is able to process video stream in an online manner, thus increase computational efficiency significantly compared with the batch mode manner. Later on, we improve the framework by introducing MIDL. Instead of adding or removing each single data separately, MIDL is capable of adding and removing multiple data simultaneously, which requires less iterations to update the innate model and reduces computational cost therefore.

- **Proposing a comprehensive computational framework for AED in Chapter 6.**

  To detect abnormal events in video surveillance, we propose a computational framework by combining MIDL with KF. A background subtraction approach is deployed to extract ROIs, which enhances the robustness of the OCSVM classifier and reduces computational time at the same time. Two types of features, motion feature and spatial information, are extracted from these ROIs. The motion feature descriptor entitled AMHOF adaptively decides its scale binning using a cluster process. By incorporating motion information and spatial location information via KF, the computational framework is capable of detecting motion anomalies and contextual anomalies even under perspective distortion. The MIDL adopts a variable cache to store the new arriving data in each frame and updates them together, thus reduces update time significantly. In addition, we improve the detection result by a MTT algorithm. Experimental results demonstrate the improvements of the framework on both effectiveness and efficiency.

- **Proposing a comprehensive computational framework for SBD in Chapter 7.**

  Apart from the AED task, we extend the MIDL framework to another video content analysis task - Shot Boundary Detection (SBD). Two successive but non-overlapping sets, i.e., immediate past set and immediate future set, are defined, in which a block-based colour histogram is extracted from each frame. Two OCSVM classifiers are trained based on these histograms in each set. Rather

than measuring the difference between pair-wise consecutive frames at a specific time, we propose a divergence metric to measure the difference between the two OCSVM classifiers. Experimental results show that the advantages are twofold. On the one hand, the framework, which inherits the advantages of OCSVM, is robust to noise such as abrupt illumination changes and large object or camera movements. On the other hand, the MIDL, which updates the OCSVM classifier in an online fashion, reduces the computational cost effectively.

8.2 Limitations and future work

The scope of crowd scene analysis in video surveillance is extensive and multifaceted. In the following, we intend to highlight a few promising future directions that may extend from this thesis work, aiming to achieve a powerful solution for online, adaptive crowd scene analysis system.

The literature in crowd scene analysis mostly used one training set and one testing set for each experiment without any variances or any statistical significant information. However, because of the temporal property of video, it should be noted that cross-validation is not suitable here as it doesn't make much sense to use later segments for training, earlier segments for testing. Therefore, one future direction is to improve the validation model.

The STLPP operates in a batch mode, which results in two limitations. Firstly, it requires the the entire dataset to be made available before computing the low-dimensional data representations. Secondly, it is infeasible to solve a generalized eigenvalue problem when dealing with large scale video data. Therefore, it will be promising direction if an online, incremental, STLPP-like dimension reduction algorithm can be developed. On the other hand, since the two modifications in Figure 3.1 are separate steps, further exploration that combined the feature projection with the online learning model will be taken into consideration.

Following most of the state-of-the-art approaches, the abnormal event in this thesis is merely defined as an abnormal motion pattern with spatial context. In other applications, however, an abnormal event may cover a variety of suspicious objects or activity. For example, a pedestrian leaves a bag in train station and walks away. Under this scenario, the bag is defined as an anomaly but our approach cannot detect it as it is not an abnormal motion pattern. Therefore, another future work is to incorporate object detection approaches within the framework using PGM (Koller and Friedman,

2009).

Another limitation in AED is that most of the benchmark datasets are synthetic or subjective. Namely, these datasets are collected based on the opinion of the published authors or organisations, and the subjectivity of the ground truth can not be ruled out. Therefore, a method with good performance for those datasets does not mean it will work well in a real application. However, it should be noted that because of privacy, it is very difficult to acquire real data. As a result, we would like to apply our framework to some real environments, e.g., train station, airport, etc, to evaluate its performance.

The MTT with the 1-NN scheme works well in our approach because of the robust anomaly detector and the sparse distributed abnormal events in datasets. However, the performance of MTT will deteriorate quickly if there are multiple closely detected abnormal events and many missed true abnormal events. To enhance the robustness of our approach, Multiple Hypothesis Tracking (MHT) (Reid, 1979; Cox and Hingorani, 1996; Kim *et al.*, 2015) will be considered in our future work. Instead of finding the best assignment of detected observation to existing tracking, MHT propagates the hypotheses into the future for anticipating so that the uncertainty will be resolved by the subsequent observations. Furthermore, it will be possible to explore enhancing the tracking performance by replacing the linear Kalman filter with nonlinear filtering methods such as extended Kalman filter, or particle filters (Arulampalam *et al.*, 2002).

As we already have the tracks for all abnormal events, one future work is to recognize different abnormal events, e.g., pedestrian, bike, and car. To achieve this idea, we may extract spatial-temporal features from those tracks. A visual vocabulary is formed by performing k-means clustering of a random subset of those features. By assigning each feature to its closest vocabulary word, those tracks are represented as histograms, where "topic model" (Blei, 2012; Wang and Grimson, 2008) can be employed to group these abnormal events.

While by adopting an abrupt forgetting scheme especially in the multiple decremental learning part of the online OCSVM, this mechanism has the risk of capturing noises (Gama *et al.*, 2014). Therefore, a trade-off between quick adaptation to changes and improved robustness to noises can be further explored.

References

Adam, A., Rivlin, E., Shimshoni, I., and Reinitz, D. (2008). Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE TPAMI*, *30*(3), 555–560.

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 510–517. Ieee.

Ali, S. and Shah, M. (2007). A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 1–6. IEEE.

Amditis, A., Thomaidis, G., Karaseitanidis, G., Lytrivis, P., and Maroudis, P. (2012). *Multiple Hypothesis Tracking Implementation*. INTECH Open Access Publisher.

Andrade, E. L., Blunsden, S., and Fisher, R. B. (2006). Modelling crowd scenes for event detection. In *Proc. ICPR'06*, Volume 1, 175–178.

Antić, B. and Ommer, B. (2011). Video parsing for abnormality detection. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2415–2422. IEEE.

Antić, B. and Ommer, B. (2015). Spatio-temporal Video Parsing for Abnormality Detection. *arXiv preprint arXiv:1502.06235*.

Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *Signal Processing, IEEE Transactions on*, *50*(2), 174–188.

Baum, L. E. and Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, *37*(6), 1554–1563.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, 404–417. Springer.

Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *19*(7), 711–720.

Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, *15*(6), 1373–1396.

Bellman, R. and Corporation, R. (1957). *Dynamic Programming*. Rand Corporation research study. Princeton University Press.

Ben-Hur, A., Horn, D., Siegelmann, H. T., and Vapnik, V. (2002). Support vector clustering. *The Journal of Machine Learning Research*, *2*, 125–137.

Benezeth, Y., Jodoin, P.-M., Saligrama, V., and Rosenberger, C. (2009). Abnormal events detection based on spatio-temporal co-occurences. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2458–2465. IEEE.

Blackman, S. S. (2004). Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, *19*(1), 5–18.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Boiman, O. and Irani, M. (2007). Detecting irregularities in images and in video. *International Journal of Computer Vision*, *74*(1), 17–31.

Bosch, A., Zisserman, A., and Munoz, X. (2007). Image classification using random forests and ferns. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 1–8. IEEE.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bregonzio, M., Gong, S., and Xiang, T. (2009). Recognising action as clouds of space-time interest points. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 1948–1955. IEEE.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). LOF: identifying density-based local outliers. In *ACM sigmod record*, Volume 29, 93–104. ACM.

Camps-Valls, G., Gomez-Chova, L., Muñoz-Marí, J., Vila-Francés, J., and Calpe-Maravilla, J. (2006). Composite kernels for hyperspectral image classification. *Geoscience and Remote Sensing Letters, IEEE*, *3*(1), 93–97.

Cauwenberghs, G. and Poggio, T. (2001). Incremental and decremental support vector machine learning. *Advances in Neural Information Processing Systems*, 409–415.

Chan, A. B., Morrow, M., and Vasconcelos, N. (2009). Analysis of crowded scenes using holistic properties. In *Performance Evaluation of Tracking and Surveillance workshop at CVPR*, 101–108.

Chan, A. B. and Vasconcelos, N. (2005a). Mixtures of dynamic textures. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, Volume 1, 641–647. IEEE.

Chan, A. B. and Vasconcelos, N. (2005b). Probabilistic kernels for the classification of auto-regressive visual processes. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Volume 1, 846–851. IEEE.

Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 15.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27.

Chaudhry, R., Ravichandran, A., Hager, G., and Vidal, R. (2009). Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 1932–1939. IEEE.

Chen, J., Ren, J., and Jiang, J. (2011). Modelling of content-aware indicators for effective determination of shot boundaries in compressed MPEG videos. *Multimedia Tools and Applications*, *54*(2), 219–239.

Chen, J. and Tang, C. K. (2007). Spatio-temporal markov random field for video denoising. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 1–8. IEEE.

Chen, M., Yang, Q., Li, Q., Wang, G., and Yang, M.-H. (2014). Spatiotemporal background subtraction using minimum spanning tree and optical flow. In *Computer Vision–ECCV 2014*, 521–534. Springer.

Ching, J. Y., Wong, A. K., and Chan, K. C. (1995). Class-dependent discretization for inductive learning from continuous and mixed-mode data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *17*(7), 641–651.

Choi, J., Wang, Z., Lee, S.-C., and Jeon, W. J. (2013). A spatio-temporal pyramid matching for video retrieval. *Computer Vision and Image Understanding*, *117*(6), 660–669.

Cong, Y., Yuan, J., and Liu, J. (2011). Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 3449–3456. IEEE.

Cong, Y., Yuan, J., and Liu, J. (2013). Abnormal event detection in crowded scenes using sparse representation. *Pattern Recognition*, *46*(7), 1851–1864.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cox, L. J. and Hingorani, S. L. (1996). An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *18*(2), 138–150.

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, Volume 1, 1–2.

Cui, X., Liu, Q., Gao, M., and Metaxas, D. N. (2011). Abnormal detection using interaction energy potentials. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 3161–3167. IEEE.

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

Dalal, N., Triggs, B., and Schmid, C. (2006). Human detection using oriented histograms of flow and appearance. In *Computer Vision–ECCV 2006*, 428–441. Springer.

Das, S., Matthews, B. L., Srivastava, A. N., and Oza, N. C. (2010). Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 47–56. ACM.

Desobry, F., Davy, M., and Doncarli, C. (2005). An online kernel change detection algorithm. *Signal Processing, IEEE Transactions on*, *53*(8), 2961–2974.

Diethe, T. and Girolami, M. (2013). Online learning with (multiple) kernels: a review. *Neural computation*, *25*(3), 567–625.

Dollár, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, 65–72. IEEE.

Doretto, G., Chiuso, A., Wu, Y. N., and Soatto, S. (2003). Dynamic textures. *International Journal of Computer Vision*, *51*(2), 91–109.

Dutta, J. K. and Banerjee, B. (2015). Online detection of abnormal events using incremental coding length. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Elgammal, A., Harwood, D., and Davis, L. (2000). Non-parametric model for background subtraction. In *Computer Vision—ECCV 2000*, 751–767. Springer.

Elgammal, A. and Lee, C.-S. (2009). Tracking people on a torus. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *31*(3), 520–538.

Fang, H., Jiang, J., and Feng, Y. (2006). A fuzzy logic approach for detection of video shot boundaries. *Pattern Recognition*, *39*(11), 2092–2100.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861–874.

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, Volume 2, 524–531. IEEE.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *32*(9), 1627–1645.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, *7*(2), 179–188.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, *46*(4), 44.

Garate, C., Bilinsky, P., and Bremond, F. (2009). Crowd event recognition using hog tracker. In *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*, 1–6. IEEE.

Geman, S. and Graffigne, C. (1986). Markov random field image models and their applications to computer vision. In *Proceedings of the International Congress of Mathematicians*, Volume 1, 2.

Golub, G. H. and van Loan, C. F. (1996). *Matrix computations*. Johns Hopkins University Press.

Gómez-Verdejo, V., Arenas-García, J., Lázaro-Gredilla, M., and Navia-Vazquez, A. (2011). Adaptive one-class support vector machine. *Signal Processing, IEEE Transactions on*, *59*(6), 2975–2981.

Gönen, M. and Alpaydın, E. (2011). Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, *12*, 2211–2268.

Grundmann, M., Meier, F., and Essa, I. (2008). 3D shape context and distance transform for action recognition. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 1–4. IEEE.

Guerbai, Y., Chibani, Y., and Hadjadji, B. (2014). Writer-independent Handwritten Signature Verification based on One-Class SVM classifier. In *Neural Networks (IJCNN), 2014 International Joint Conference on*, 327–331. IEEE.

Hanjalic, A. (2002). Shot-boundary detection: unraveled and resolved? *Circuits and Systems for Video Technology, IEEE Transactions on*, *12*(2), 90–105.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, Volume 15, 50. Citeseer.

He, X., Yan, S., Hu, Y., Niyogi, P., and Zhang, H.-J. (2005). Face recognition using Laplacianfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *27*(3), 328–340.

He, X., Zemel, R. S., and Carreira-Perpiñán, M. Á. (2004). Multiscale conditional random fields for image labeling. In *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on*, Volume 2, II–695. IEEE.

Helbing, D. and Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical review E*, *51*(5), 4282.

Hoffman, M., Bach, F. R., and Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, 856–864.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, *42*(1-2), 177–196.

Horn, B. (1986). *Robot vision*. MIT press.

Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. In *1981 Technical Symposium East*, 319–331. International Society for Optics and Photonics.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, *24*(6), 417.

Hou, X. and Zhang, L. (2007). Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 1–8. IEEE.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, *13*(2), 415–425.

Huang, C.-L. and Liao, B.-Y. (2001). A robust scene-change detection method for video segmentation. *Circuits and Systems for Video Technology, IEEE Transactions on*, *11*(12), 1281–1288.

Huttenlocher, D. P., Klanderman, G. A., and Rucklidge, W. J. (1993). Comparing images using the Hausdorff distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *15*(9), 850–863.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11), 1254–1259.

Jiang, X., Sun, T., Liu, J., Chao, J., and Zhang, W. (2013). An adaptive video shot segmentation scheme based on dual-detection model. *Neurocomputing*, *116*, 102–111.

Joachims, T. (1998). *Text categorization with support vector machines: Learning with many relevant features*. Springer.

Joachims, T., Cristianini, N., and Shawe-Taylor, J. (2001). Composite kernels for hypertext categorisation. In *ICML*, Volume 1, 250–257.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, *82*(1), 35–45.

Karasuyama, M. and Takeuchi, I. (2009). Multiple incremental decremental learning of support vector machines. In *Advances in Neural Information Processing Systems*, 907–915.

Kawai, Y., Sumiyoshi, H., and Yagi, N. (2007). Shot Boundary Detection at TRECVID 2007. In *TRECVID*.

Ke, Y., Sukthankar, R., and Hebert, M. (2005). Efficient visual event detection using volumetric features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, Volume 1, 166–173. IEEE.

Ke, Y., Sukthankar, R., and Hebert, M. (2007). Event detection in crowded videos. In *Proc. ICCV'07*, 1–8.

Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., and Murthy, K. R. K. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, *13*(3), 637–649.

Kembhavi, A., Siddiquie, B., Miezianko, R., McCloskey, S., and Davis, L. S. (2009). Incremental multiple kernel learning for object recognition. In *Computer Vision, 2009 IEEE 12th International Conference on*, 638–645. IEEE.

Khan, S. S. and Madden, M. G. (2009). A survey of recent trends in one class classification. In *Artificial Intelligence and Cognitive Science*, 188–197. Springer.

Kim, C., Li, F., Ciptadi, A., and Rehg, J. M. (2015). Multiple Hypothesis Tracking Revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, 4696–4704.

Kim, I. S., Choi, H. S., Yi, K. M., Choi, J. Y., and Kong, S. G. (2010). Intelligent visual surveillance—a survey. *International Journal of Control, Automation and Systems*, *8*(5), 926–939.

Kim, J. and Grauman, K. (2009). Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2921–2928. IEEE.

Kim, K. I., Jung, K., and Kim, H. J. (2002). Face recognition using kernel principal component analysis. *Signal Processing Letters, IEEE*, *9*(2), 40–42.

Kivinen, J., Smola, A. J., and Williamson, R. C. (2004). Online learning with kernels. *Signal Processing, IEEE Transactions on*, *52*(8), 2165–2176.

Klaser, A., Marszałek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3d-gradients. In *BMVC 2008-19th British Machine Vision Conference*, 275–1. British Machine Vision Association.

Kolev, D., Suvorov, M., Morozov, E., Markarian, G., and Angelov, P. (2015). Incremental Anomaly Identification in Flight Data Analysis by Adapted One-Class SVM Method. In *Artificial Neural Networks*, 373–391. Springer.

Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Kowdle, A. and Chen, T. (2012). Learning to Segment a Video to Clips Based on Scene and Camera Motion. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, Berlin, Heidelberg, 272–286. Springer-Verlag.

Kratz, L. and Nishino, K. (2009). Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 1446–1453. IEEE.

Kullback, S. (1968). *Information theory and statistics*. Courier Corporation.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, San Francisco, CA, USA, 282–289. Morgan Kaufmann Publishers Inc.

Lakshmi Priya, G. and Domnic, S. (2014). Walsh–Hadamard Transform Kernel-Based Feature Vector for Shot Boundary Detection. *Image Processing, IEEE Transactions on*, *23*(12), 5187–5197.

Lanckriet, G. R., De Bie, T., Cristianini, N., Jordan, M. I., and Noble, W. S. (2004). A statistical framework for genomic data fusion. *Bioinformatics*, *20*(16), 2626–2635.

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*(2-3), 107–123.

Laptev, I., Marszałek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1–8. IEEE.

Laptev, I. and Pérez, P. (2007). Retrieving actions in movies. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 1–8. IEEE.

Larose, D. T. (2005). k-Nearest Neighbor Algorithm. *Discovering Knowledge in Data: An Introduction to Data Mining*, 90–106.

Laskov, P., Gehl, C., Krüger, S., and Müller, K.-R. (2006). Incremental support vector learning: Analysis, implementation and applications. *The Journal of Machine Learning Research*, *7*, 1909–1936.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, Volume 2, 2169–2178. IEEE.

Lee, H., Battle, A., Raina, R., and Ng, A. Y. (2006). Efficient sparse coding algorithms. In *Advances in neural information processing systems*, 801–808.

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2548–2555. IEEE.

Li, T., Chang, H., Wang, M., Ni, B., Hong, R., and Yan, S. (2015). Crowded scene analysis: A survey. *Circuits and Systems for Video Technology, IEEE Transactions on*, *25*(3), 367–386.

Li, W., Mahadevan, V., and Vasconcelos, N. (2014). Anomaly Detection and Localization in Crowded Scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *36*(1), 18–32.

Lin, H., Deng, J. D., and Woodford, B. J. (2013). Event detection using quantized binary code and spatial-temporal locality preserving projections. In *AI 2013: Advances in Artificial Intelligence*, 123–134. Springer.

Lin, H., Deng, J. D., and Woodford, B. J. (2014). Spatial-Temporal Pyramid Matching for Crowd Scene Analysis. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, 12. ACM.

Lin, H., Deng, J. D., and Woodford, B. J. (2015). Anomaly Detection in Crowd Scenes via Online One-Class Support Vector Machines. In *IEEE Conference on Image Processing (ICIP'2015)*, 2434–2438. IEEE Press.

Lin, H., Deng, J. D., and Woodford, B. J. (2016). Shot Boundary Detection Using Multi-instance Incremental and Decremental One-Class Support Vector Machine. In *Advances in Knowledge Discovery and Data Mining*, 165–176. Springer.

Liu, C., Freeman, W. T., Adelson, E. H., and Weiss, Y. (2008). Human-assisted motion annotation. In *Proc. CVPR'08*, 1–8.

Liu, H., Chen, S., and Kubota, N. (2013). Intelligent video systems and analytics: a survey. *Industrial Informatics, IEEE Transactions on*, *9*(3), 1222–1233.

Liu, W., Hua, G., and Smith, J. (2014). Unsupervised one-class learning for automatic outlier removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3826–3833.

Ljung, L. (1998). *System identification*. Springer.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91–110.

Lu, C., Shi, J., and Jia, J. (2013). Abnormal event detection at 150 fps in matlab. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 2720–2727. IEEE.

Lu, K. and He, X. (2005). Image retrieval based on incremental subspace learning. *Pattern Recognition*, *38*(11), 2047–2054.

Lu, Z.-M. and Shi, Y. (2013). Fast video shot boundary detection based on SVD and pattern matching. *Image Processing, IEEE Transactions on*, *22*(12), 5136–5145.

Lucas, B. D., Kanade, T., *et al.* (1981). An iterative image registration technique with an application to stereo vision. In *IJCAI*, Volume 81, 674–679.

Ma, J. and Perkins, S. (2003). Time-series novelty detection using one-class support vector machines. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, Volume 3, 1741–1745. IEEE.

Mahadevan, V., Li, W., Bhalodia, V., and Vasconcelos, N. (2010). Anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 1975–1981. IEEE.

Mahadevan, V. and Vasconcelos, N. (2010). Spatiotemporal saliency in dynamic scenes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *32*(1), 171–177.

Mairal, J., Bach, F., and Ponce, J. (2014). Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*.

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, *11*, 19–60.

Manevitz, L. M. and Yousef, M. (2002). One-class SVMs for document classification. *the Journal of Machine Learning Research*, *2*, 139–154.

Manjunath, B. S., Ohm, J.-R., Vasudevan, V. V., and Yamada, A. (2001). Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, *11*(6), 703–715.

McLachlan, G. J. (1999). Mahalanobis distance. *Resonance*, *4*(6), 20–26.

Mehran, R., Oyama, A., and Shah, M. (2009). Abnormal crowd behavior detection using social force model. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 935–942. IEEE.

Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International journal of computer vision*, *60*(1), 63–86.

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *27*(10), 1615–1630.

Mohanta, P. P., Saha, S. K., and Chanda, B. (2012). A model-based shot boundary detection technique using frame transition parameters. *Multimedia, IEEE Transactions on*, *14*(1), 223–233.

Moosmann, F., Nowak, E., and Jurie, F. (2008). Randomized clustering forests for image classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *30*(9), 1632–1646.

Moreno, P. J., Ho, P. P., and Vasconcelos, N. (2003). A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications. In *Advances in neural information processing systems*, None.

Mühling, M., Ewerth, R., Stadelmann, T., Zöfel, C., Shi, B., and Freisleben, B. (2007). University of Marburg at TRECVID 2007: Shot Boundary Detection and High Level Feature Extraction. In *TRECVID*.

Nguyen, H. T., Ji, Q., and Smeulders, A. W. (2007). Spatio-temporal context for robust multitarget tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *29*(1), 52–64.

Niebles, J. C., Wang, H., and Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, *79*(3), 299–318.

Nistér, D. and Stewénius, H. (2008). Linear time maximally stable extremal regions. In *Computer Vision–ECCV 2008*, 183–196. Springer.

Niyogi, X. (2004). Locality preserving projections. In *Neural information processing systems*, Volume 16, 153. MIT.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods*, 185–208. MIT press.

Pless, R. (2003). Image Spaces and Video Trajectories: Using Isomap to Explore Video Sequences. In *ICCV*, Volume 3, 1433–1440.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. (2008). SimpleMKL. *Journal of Machine Learning Research*, *9*, 2491–2521.

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

Ren, J., Jiang, J., and Chen, J. (2007). Determination of Shot Boundary in MPEG Videos for TRECVID 2007. In *TRECVID*.

Roshtkhari, M. and Levine, M. (2013a). Online dominant and anomalous behavior detection in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2611–2618.

Roshtkhari, M. J. and Levine, M. D. (2013b). An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. *Computer vision and image understanding*, *117*(10), 1436–1452.

Roth, V. (2006). Kernel fisher discriminants for outlier detection. *Neural computation*, *18*(4), 942–960.

Rougier, C., Meunier, J., St-Arnaud, A., and Rousseau, J. (2011). Robust video surveillance for fall detection based on human shape deformation. *Circuits and Systems for Video Technology, IEEE Transactions on*, *21*(5), 611–622.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, *290*(5500), 2323–2326.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: an efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2564–2571. IEEE.

Saghafi, B. and Rajan, D. (2012). Human action recognition using pose-based discriminant embedding. *Signal Processing: Image Communication*, *27*(1), 96–111.

Saligrama, V. and Chen, Z. (2012). Video anomaly detection based on local statistical aggregates. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2112–2119. IEEE.

Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of computer vision*, *37*(2), 151–172.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, *13*(7), 1443–1471.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, *10*(5), 1299–1319.

Scholkopf, B. and Smola, A. J. (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Schölkopf, B., Smola, A. J., Williamson, R. C., and Bartlett, P. L. (2000). New support vector algorithms. *Neural computation*, *12*(5), 1207–1245.

Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th international conference on Multimedia*, 357–360. ACM.

Shechtman, E. and Irani, M. (2007). Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, 1–8. IEEE.

Shi, Y., Gao, Y., and Wang, R. (2010). Real-time abnormal event detection in complicated scenes. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, 3653–3656. IEEE.

Sivic, J., Russell, B. C., Efros, A. A., Zisserman, A., and Freeman, W. T. (2005). Discovering objects and their location in images. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, Volume 1, 370–377. IEEE.

Sivic, J. and Zisserman, A. (2009). Efficient visual search of videos cast as text retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *31*(4), 591–606.

Smeaton, A. F., Over, P., and Doherty, A. R. (2010). Video shot boundary detection: Seven years of TRECVid activity. *Computer Vision and Image Understanding*, *114*(4), 411–418.

Sodemann, A. A., Ross, M. P., and Borghetti, B. J. (2012). A review of anomaly detection in automated surveillance. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *42*(6), 1257–1272.

Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large scale multiple kernel learning. *The Journal of Machine Learning Research*, *7*, 1531–1565.

Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, Volume 2, 246–252. IEEE.

Su, H., Yang, H., Zheng, S., Fan, Y., and Wei, S. (2012). Crowd event perception based on spatio-temporal viscous fluid field. In *Advanced video and signal-based surveillance (AVSS), 2012 IEEE Ninth International Conference on*, 458–463. IEEE.

Su, H., Yang, H., Zheng, S., Fan, Y., and Wei, S. (2013). The large-scale crowd behavior perception based on spatio-temporal viscous fluid field. *Information Forensics and Security, IEEE Transactions on*, *8*(10), 1575–1589.

Suzuki, K., Horiba, I., and Sugie, N. (2003). Linear-time connected-component labeling based on sequential local operations. *Computer Vision and Image Understanding*, *89*(1), 1–23.

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, *7*(1), 11–32.

Tax, D. M. and Duin, R. P. (2004). Support vector data description. *Machine learning*, *54*(1), 45–66.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *science*, *290*(5500), 2319–2323.

Thida, M., Eng, H.-L., Dorothy, M., and Remagnino, P. (2010). Learning video manifold for segmenting crowd events and abnormality detection. In *Computer Vision–ACCV 2010*, 439–449. Springer.

Thida, M., Eng, H.-L., Monekosso, D. N., and Remagnino, P. (2012). Learning video manifolds for content analysis of crowded scenes. *IPSJ Transactions on Computer Vision and Applications*, *4*, 71–77.

Thompson, D. R., Majid, W. A., Reed, C., and Wagstaff, K. L. (2013). Semi-supervised Eigenbasis novelty detection. *Statistical Analysis and Data Mining*, *6*(3), 195–204.

Tian, Y., Sukthankar, R., and Shah, M. (2013). Spatiotemporal deformable part models for action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2642–2649.

Tipping, M. E. and Bishop, C. M. (1999). Mixtures of probabilistic principal component analyzers. *Neural computation*, *11*(2), 443–482.

Torki, M. and Elgammal, A. (2010a). One-shot multi-set non-rigid feature-spatial matching. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3058–3065. IEEE.

Torki, M. and Elgammal, A. (2010b). Putting local features on a manifold. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 1743–1750. IEEE.

Torrence, C. and Compo, G. P. (1998). A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, *79*(1), 61–78.

Tsai, C.-J., Lee, C.-I., and Yang, W.-P. (2008). A discretization algorithm based on class-attribute contingency coefficient. *Information Sciences*, *178*(3), 714–731.

Tseng, B. L., Lin, C.-Y., and Smith, J. R. (2002). Real-time video surveillance for traffic monitoring using virtual line analysis. In *Multimedia and Expo, 2002. ICME'02. Proceedings. 2002 IEEE International Conference on*, Volume 2, 541–544. IEEE.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, *3*(1), 71–86.

Tziakos, I., Cavallaro, A., and Xu, L.-Q. (2009). Video event segmentation and visualisation in non-linear subspace. *Pattern Recognition Letters*, *30*(2), 123–131.

Tziakos, I., Cavallaro, A., and Xu, L.-Q. (2010). Event monitoring via local motion abnormality detection in non-linear subspace. *Neurocomputing*, *73*(10), 1881–1891.

Valera, M. and Velastin, S. A. (2005). Intelligent distributed surveillance systems: a review. In *Vision, Image and Signal Processing, IEE Proceedings-*, Volume 152, 192–204. IET.

Van De Sande, K. E., Gevers, T., and Snoek, C. G. (2010). Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *32*(9), 1582–1596.

van Gemert, J., Veenman, C., Smeulders, A., and Geusebroek, J.-M. (2010). Visual Word Ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(7), 1271–1283.

Vedaldi, A. and Zisserman, A. (2012). Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *34*(3), 480–492.

Viola, P. and Jones, M. (2001). Robust real-time object detection. *International Journal of Computer Vision*, *4*.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, *57*(2), 137–154.

Viola, P., Jones, M. J., and Snow, D. (2005). Detecting pedestrians using patterns of motion and appearance. *International Journal of Computer Vision*, *63*(2), 153–161.

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *BMVC 2009-British Machine Vision Conference*, 124–1. BMVA Press.

Wang, L. and Suter, D. (2007). Learning and matching of dynamic shape manifolds for human action recognition. *Image Processing, IEEE Transactions on*, *16*(6), 1646–1661.

Wang, L. and Suter, D. (2008). Visual learning and recognition of sequential data manifolds with applications to human movement analysis. *Computer Vision and Image Understanding*, *110*(2), 153–172.

Wang, T., Chen, J., Zhou, Y., and Snoussi, H. (2013). Online Least Squares One-Class Support Vector Machines-Based Abnormal Visual Event Detection. *Sensors*, *13*(12), 17130–17155.

Wang, X. (2013). Intelligent multi-camera video surveillance: A review. *Pattern recognition letters*, *34*(1), 3–19.

Wang, X. and Grimson, E. (2008). Spatial latent dirichlet allocation. In *Advances in neural information processing systems*, 1577–1584.

Wedel, A., Brox, T., Vaudrey, T., Rabe, C., Franke, U., and Cremers, D. (2011). Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision*, *95*(1), 29–51.

Weinshall, D., Levi, G., and Hanukaev, D. (2013). LDA topic model with soft assignment of descriptors to words. In *Proceedings of the 30th International Conference on Machine Learning*, 711–719.

Welch, G. and Bishop, G. (1995). An Introduction to the Kalman Filter. Technical report, Chapel Hill, NC, USA.

Willems, G., Tuytelaars, T., and Van Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *Computer Vision–ECCV 2008*, 650–663. Springer.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., and Ma, Y. (2009). Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *31*(2), 210–227.

Wu, S., Moore, B. E., and Shah, M. (2010). Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 2054–2060. IEEE.

Yang, J., Yu, K., Gong, Y., and Huang, T. (2009). Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 1794–1801. IEEE.

Yu, T.-H., Woodford, O. J., and Cipolla, R. (2013). A performance evaluation of volumetric 3D interest point detectors. *International journal of computer vision*, *102*(1-3), 180–197.

Yuan, J., Wang, H., Xiao, L., Zheng, W., Li, J., Lin, F., and Zhang, B. (2007). A formal study of shot boundary detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, *17*(2), 168–186.

Yuan, Y., Fang, J., and Wang, Q. (2015). Online anomaly detection in crowd scenes via structure analysis. *Cybernetics, IEEE Transactions on*, *45*(3), 548–561.

Zhao, B., Fei-Fei, L., and Xing, E. P. (2011). Online detection of unusual events in videos via dynamic sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 3313–3320. IEEE.

Zhao, Z.-C., Zeng, X., Liu, T., and Cai, A.-N. (2007). BUPT at TRECVID 2007: Shot Boundary Detection. In *TRECVID*.

Zhu, Q., Yeh, M.-C., Cheng, K.-T., and Avidan, S. (2006). Fast human detection using a cascade of histograms of oriented gradients. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, Volume 2, 1491–1498. IEEE.

Zhu, X., Liu, J., Wang, J., Li, C., and Lu, H. (2014). Sparse representation for robust abnormality detection in crowded scenes. *Pattern Recognition*, *47*(5), 1791–1799.

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