Abnormal detection in video streams via one-class learning methods

Tian Wang

Thèse
de doctorat
de l'UTT

Tian WANG

Abnormal Detection in Video Streams via One-class Learning Methods

Spécialité :
Optimisation et Sûreté des Systèmes

2014TROY0018

Année 2014

utt
université de technologie
Troyes

THESE

pour l'obtention du grade de

DOCTEUR de l'UNIVERSITE DE TECHNOLOGIE DE TROYES

Spécialité : OPTIMISATION ET SURETE DES SYSTEMES

présentée et soutenue par

Tian WANG

le 6 mai 2014

Abnormal Detection in Video Streams via One-class Learning Methods

JURY

M. F. DORNAIKA	PROFESSOR	Président
M. F. ABDALLAH	MAITRE DE CONFERENCES - HDR	Rapporteur
M. P. HONEINE	MAITRE DE CONFERENCES - HDR	Examinateur
M. A. RAKOTOMAMONJY	PROFESSEUR DES UNIVERSITES	Rapporteur
M. H. SNOUSSI	PROFESSEUR DES UNIVERSITES	Directeur de thèse

Acknowledgments

I would like to express my gratitude to all those who helped me during my doctoral studies and the writing of this thesis.

My deepest gratitude goes first to my supervisor Professor Hichem Snoussi, for his constantly constant encouragement and guidance of my research. He provides me with an excellent atmosphere for doing research through the three and a half years. I wish to express my gratitude to China Scholarship Council (CSC) and University of Technology of Troyes (UTT) for the financial support during these three and a half years on France.

I would like to express my sincere gratitude to Mr. Paul Honeine, Mr. Xiaolu Gong, Ms. Ling Gong and Ms. Muriel Whitchurch in University of Technology of Troyes, Mr. Jie Chen in University of Nice Sophia Antopolis, and Yi Zhou in Dalian Maritime University, for their valuable comments on my research. Thanks the secretaries of the pôle ROSAS Ms. Marie-José Rousselet, Ms. Veronique Banse and Ms. Bernadette Andre, and the secretaries of the doctoral school: Ms. Isabelle Leclercq, Ms. Pascale Denis and Ms. Therese Kazarian, for their help throughout my PhD study.

I want to thank my friends in UTT for their valuable supports and aids, and all my other friends in France or in China. Special thanks to Aichun Zhu, Syrine Roufaida Ait Haddanene, Lei Qin, Xiaowei Lv, Yuan Dong, Guoliang Zhu, Jian Zhang, Wenjin Zhu, Kun jia, Zhenming Yue and Huan Wang, they always help me and give me their best suggestions.

Lastly, I offer sincere thanks to my parents, my brother and all my family members, for their loving considerations and great confidence in me all through these years. My father is the greatest person in my heart, he always encourage me, and help me to analysis the problems. My mother raises me up with her excellent caring, and trusts me in all the condition.

Abnormal Detection in Video Streams via One-class Learning Methods

Abstract: One of the major research areas in computer vision is visual surveillance. The scientific challenge in this area includes the implementation of automatic systems for obtaining detailed information about the behavior of individuals and groups. Particularly, detection of abnormal individual movements requires sophisticated image analysis. This thesis focuses on the problem of the abnormal event detection, including feature descriptor design characterizing the movement information and one-class kernel-based classification methods. In this thesis, three different image features have been proposed: (i) global optical flow features, (ii) histograms of optical flow orientations (HOFO) descriptor and (iii) covariance matrix (COV) descriptor. Based on these proposed descriptors, one-class support vector machines (SVM) are proposed in order to detect abnormal events. Two online strategies of one-class SVM are proposed: the first strategy is based on support vector description (online SVDD) and the second strategy is based on online least squares one-class support vector machines (online LS-OC-SVM).

Keywords: Signal detection; Multivariate analysis; Support vector machines; Analysis of covariance.

Algorithmes d'apprentissage mono-classe pour la détection d'anomalies dans les flux vidéo

Résume: La vidéo surveillance représente l'un des domaines de recherche privilégiés en vision par ordinateur. Le défi scientifique dans ce domaine comprend la mise en œuvre de systèmes automatiques pour obtenir des informations détaillées sur le comportement des individus et des groupes. En particulier, la détection de mouvements anormaux de groupes d'individus nécessite une analyse fine des frames du flux vidéo. Dans le cadre de cette thèse, la détection de mouvements anormaux est basée sur la conception d'un descripteur d'image efficace ainsi que des méthodes de classification non linéaires. Nous proposons trois caractéristiques pour construire le descripteur de mouvement : (i) le flux optique global, (ii) les histogrammes de l'orientation du flux optique (HOFO) et (iii) le descripteur de covariance (COV) fusionnant le flux optique et d'autres caractéristiques spatiales de l'image. Sur la base de ces descripteurs, des algorithmes de machine learning (machines à vecteurs de support (SVM)) mono-classe sont utilisés pour détecter des événements anormaux. Deux stratégies en ligne de SVM mono-classe sont proposées : la première est basée sur le SVDD (online SVDD) et la deuxième est basée sur une version "moindres carrés" des algorithmes SVM (online LS-OC-SVM).

Les mots clés: Détection du signal; Analyse multivariée; Machines à vecteurs support; Analyse de covariance.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Contents

One of the major research areas in computer vision is visual surveillance. The scientific challenge in this area includes the implementation of automatic systems for obtaining detailed information about the behavior of individuals and groups. Obtaining detailed information about the behavior of individuals from video frames obtained by a visual sensor, is a challenging task. Particularly, detection of abnormal individual movements requires sophisticated image analysis.

1.1 Overview of video abnormal detection

The abnormal detection problems have other names in the literature, such as suspicious event, irregular behavior, uncommon behavior, unusual activity/event/behavior, abnormal behavior, anomaly, etc. [Popoola 2012]. The research focused on news broadcast video; conference video; unmanned aerial vehicle (UAV) motion imagery and ground recognition video; surveillance video of the areas including market, museum, warehouse, room of old people, plaza, airport terminal, parking lot, traffic, subway stations, aerial surveillance, and sign language data. In this section, firstly, several video abnormal detection systems are introduced. And then, the abnormal event detection handled in this thesis is generally described.

1.1.1 Video abnormal detection systems

Video analytics gained significant research interest in the 90s of the last century, when the defense advanced research projection agency (DARPA) started sponsoring detection, recognition, and understanding of moving object events [Candamo 2010]. Digital image processing, advanced video codec techniques and pattern recognition algorithms have been applied to the visual surveillance field.

The video analysis and content extraction (VACE) project focused on automatic video content extraction, multi-model fusion, event recognition and understanding. DARPA has supported several research projects, which include visual surveillance and monitoring (VSAM, 1997) [Collins 2000], human identification at a distance (HID,2000), video and image retrieval analysis tool (VIRAT, 2008) [Candamo 2010].

The public transportation system is also a domain related to computer vision problems. The New York city transit system is the busiest metro system in the U.S.A. (based on 2006 statistics) [Metro b, Candamo 2010], Moscow metro is the busiest metro in Europe (based on 2007 statistics) [Metro a], Paris public transportation network (RATP) is the second busiest metro system in Europe [Metro c]. The challenge for real-time events detection solutions (CREDS) [Ziliani 2005] defined by the needs of RATP focused on proximity warning dropping objects on tracks, launching objects across platforms, persons trapped by the door of a moving train, walking on rails, failing on the track and crossing the rails. The French project SAMSIT (Système d'Analyse de Médias pour une Sécurité Intelligente dans les Transports publics) aims at designing solutions for the automatic surveillance in public transport vehicles, such as trains and metros, by analyzing human behaviors based on audio-video stream interpretation [Vu 2006].

1.1.2 Definition of abnormal detection

Several normal and abnormal scenes are shown in Fig.1.1. In Fig.1.1(a)(b), all the people are walking, these scenes are considered as normal. In Fig.1.1(d), an unusual group movement is detected, the people are suddenly running in different directions. Another abnormal example is shown in Fig.1.1(d), the major people in the frame are walking, while one person is running. In abnormal detection problems, it is supposed that the samples from a positive class are available. Thus, the one-class classification method is used in this thesis.

1.2 Summary of the thesis

The main contributions in this thesis and the layout are briefly summarized below.

1.2.1 Main contributions

This thesis focuses on the abnormal detection problem via one-class classification methods. The main thesis contributions are as follows:

Firstly, the algorithm is based on features of optical flow and one-class support vector machine (OC-SVM). The optical flow is computed at each pixel of the video frame, and the nonlinear one-class SVM, after a learning period characterizing normal behavior, detects the abnormal pixels or blobs in the current frame. The blob extraction method in the crowded video scenes is proposed to detect abnormal blob events. A structural high dimensional descriptor, histograms of optical flow orientation (HOFO) is proposed as a descriptor encoding the moving information of each video frame.

(a) Normal plaza scene (b) Normal indoor scene (c) Mall scene

(d) Abnormal plaza scene (e) PETS (f) 2 people scene

Figure 1.1: Examples of the normal and abnormal scenes. (a) All the people are walking, the normal plaza scene in UMN datasets [UMN 2006]. (b) All the people are walking, the normal indoor scene in UMN datasets. (c) One person is running and the others are walking, the normal and abnormal blobs. (d) All the people are running, the abnormal plaza scene. (e) All the people are walking, the normal scene in PETS dataset [PETS 2009]. (f) Two people are walking, a normal scene of UTT dataset.

Secondly, the covariance matrix descriptor (COV) is proposed to fuse the image intensity and the optical flow. A multi-kernel learning strategy improving the detection performance is proposed as well.

Thirdly, as the abnormal detection problem usually concerns a long video sequence, we propose two online detection algorithms, online support vector data description (online SVDD) and online least squares one-class support vector machine (online LS-OC-SVM).

The proposed feature descriptor, online one-class classification methods, and the datasets on which the proposed methods are tested on are abstracted in the TABLE 1.2.1.

1.2.2 Layout of the thesis

The thesis is organized as follows.

In Chapter 2, the state of the art of the abnormal detection and event recognition methods is introduced. Two main components, abstraction and event modeling, are identified.

In Chapter 3, the basic structure of our work, which is based on event represention descriptor and pattern classification method is introduced. The algorithm is based on optical flow descriptor and one-class SVM classifier. Three feature extraction strategies, pixel-by-pixel, block-by-block, and block$^{\text{all}}$-by-block are proposed. A blob extraction method is presented to extract blobs from crowded scenes. We propose histogram of optical flow orientation (HOFO) as a descriptor encoding the moving information of each video frame.

Chapter	Method		Dataset			
Chapter 3	Optical flow	Pixel-by-Pixel	UTT	UMN		
		Block-by-Block	UTT	UMN		
		$\text{Block}^{\text{all}}$-by-Block	UTT	UMN		
	HOFO Blob		UTT	UMN	Mall	
	HOFO Frame			UMN		PETS
Chapter 5	COV Blob		UTT	UMN	Mall	
	COV Frame			UMN		PETS
Chapter 6	online SVDD	dictionary fixed through train		UMN		
		dictionary fixed through test		UMN		
	online LS-SVM	no dictionary through train		UMN		
		dictionary through train		UMN		

Table 1.1: The proposed feature descriptors and online one-class classification methods. The proposed feature descriptors include block optical flow feature descriptor, histograms of optical flow orientations (HOFO), and covariance matrix descriptor (COV). The proposed online one-class classification methods include: online support vector data description (online SVDD), online least squares one-class support vector machine (online LS-OC-SVM), sparse online least squares one-class support vector machine (sparse online LS-OC-SVM). The datasets used for presenting the method performance are labeled.

In Chapter 4, we propose the covariance matrix descriptor fusing the image intensity and the optical flow to encode moving information and image characteristics of a blob or a frame. A multi-kernel strategy which consists of several parts tuning the importance of each sub-image is proposed to improve the detection accuracy.

In Chapter 5, abnormal detection via online support vector data description (online SVDD) and via online least squares one-class support vector machine (online LS-OC-SVM) are proposed. Covariance matrix descriptor is used for these online implementations.

Chapter 6 concludes this thesis and discusses the future work.

CHAPTER 2

State of the art of abnormal detection

Contents

The abnormal events detection is the focus of this thesis, it includes feature descriptor characterizing the movement information and one-class classification methods. In this chapter, the state of the art related to abnormal event detection problems and event recognition problems [Lavee 2009a, Lavee 2009b], are introduced. Two main components of the abnormal detection and event recognition, namely abstraction and event modeling, are identified. Abstraction is the process of modeling the data into informative units to be used as input to the event model. Event modeling is devoted to formally describe events of interest, enabling recognition of these events as they occur in the video sequence.

2.1 Abstraction

Abstraction is the organization of low-level inputs into various constructs (or "primitives") representing the properties of the video data. There are three main categories of abstraction approaches: pixel-based, object-based, and logic-based abstractions. Pixel-based abstraction describes the properties of pixel features in the low-level input. Object-based abstraction describes the low-level input in terms of semantic objects. Logic-based abstraction organizes the low-level input into statement of semantic knowledge [Lavee 2009b].

2.1.1 Pixel-based abstraction

Pixel-based abstraction does not attempt to group pixel regions into blobs or objects, but simply computes features based on the salient pixel regions of an input video sequence. It relies on pixel or pixel group features such as color, texture and gradient. This method is the organization of low-level input into vectors in an N-dimensional metric space [Ribeiro 2005, Zhong 2004, Shechtman 2005]. Additional information related to trajectory could be also included in this category, such as in [Ribeiro 2005] where the speed of the object is used as an additional feature.

Pixel-based abstraction methods include histograms of spatio-temporal gradients [Zelnik-Manor 2006]; spatio-temporal patches [Dollár 2005, Laptev 2007, Niebles 2008, Haines 2011, Kim 2009, Benezeth 2011, Benezeth 2009, Bregler 1997, Wang 2006]; self-similarity surfaces [Shechtman 2005]; motion history images (MHI) motion energy images (MEI) and pixel change history (PCH) [Bobick 2001, Zhong 2004, Ng 2001, Ng 2003, Gong 2003, Kosmopoulos 2010, Jiménez-Hernández 2010, Bradski 2002, Davis 2001]; optical flow [Utasi 2010, Utasi 2008a, Utasi 2008b, Kwak 2011, Adam 2008, Varadarajan 2009]; middle-level feature consisting of serval patches [Boiman 2007] (please refer to details of middle-level feature in [Singh 2012, Doersch 2012]).

2.1.2 Object-based abstraction

Object-based abstraction is an approach based on the intuition that a description of the objects participating in the video sequence is a good intermediate representation for event reasoning. Thus the low-level input is abstracted into a set of objects within their associated properties such as speed, position and trajectory. The objects of the interest are labeled by bounding boxes[Hongeng 2001, Xiang 2008a, Xiang 2005, Xiang 2008b, Xiang 2002, Starner 1995, Medioni 2001, Varadarajan 2009, Yao 2010], silhouettes[Blank 2005, Schuldt 2004, Wang 2007, Singh 2008, Chen 2007, Sminchisescu 2006], trajectories [Piciarelli 2008b, Piciarelli 2006, Piciarelli 2005, Piciarelli 2007, Piciarelli 2008a, Calavia 2012, Jiang 2011, Jiang 2012] and 3D trajectories [Lee 2012].

2.1.3 Logic-based abstraction

Logic-based abstraction aims at abstracting low-level inputs into statements of semantic knowledge which can be reasoned on by a rule-based event model. This abstraction is motivated by the observation that the world is not described by multi-dimensional parameterizations of pixel distributions, or even a set of semantic objects and their properties, but rather by a set of semantic rules and concepts, which act upon units of knowledge [Lavee 2009b]. The representation space after the abstraction is smaller than the original space, the influence of uncertainty errors is reduced.

In [Siskind 2000], low-level input is abstracted into line segments associated by kinematic stability concepts such as grounding and support. In [Cohn 2003], the chosen abstraction scheme focuses mainly on the spatial aspects of the event, where a set of qualitative spatial relations is applied to the video sequence.

2.2 Event modeling

Event modeling is the subsequent problem to abstraction. Given the choice of an abstraction scheme, event modeling seeks formal ways to describe and recognize events in a particular domain. There are roughly three categories: pattern-recognition methods, state event models and semantic event models.

2.2.1 Pattern-recognition methods

The classifiers in this category do not consider the problem of event representation, they focus on the event recognition problem formulated as a traditional pattern recognition problem. This class consists of nearest neighbor, support vector machines, neural networks [Lavee 2009a]. The main advantage of these techniques is that they can be fully specified from a set of training data. As these methods exclude semantics, i.e. high-level knowledge about the event domain, from the specification of the classifier, they are usually simple and straightforward to be implemented. The representational considerations are usually left to the abstraction scheme associated with the event recognition method.

2.2.1.1 Nearest neighbors

Nearest neighbors is widely used for classification. An unlabeled sample is labeled using its "nearest" labeled neighbor in the database. K-nearest neighbor is a variation of nearest neighbors methods where the K nearest neighbors vote the label of the test example. The notion of closeness is defined by a distance measure decided upon during the model specification [Bishop 2006]. The distance measure can be Euclidean [Blank 2005, Gorelick 2007, Masoud 2003], Chi-squared [Zelnik-Manor 2006] and Linear programming based distance [Jiang 2006]. Event-domain dependent metrics such as spatio-temporal region intersection [Ke 2007] and gradient matrix of motion field [Shechtman 2005] are also used as distance measures. Template matching methods [Bobick 2001, Ng 2001, Ng 2003] also use nearest neighbors models.

In [Bobick 2001], motion-energy images (MEI) and motion-history image (MHI) are used to represent the movement. There were two component version of the templates. The first value was a binary value indicating the presence of motion, and the second value was a function of the recency of motion in a sequence. The Mahalanobis distance was used in the nearest neighbor event model.

One can note that, the abstraction of video events is often high-dimensional, a sufficiently dense nearest neighbor event model is intractable for recognition (complexity grows with the dataset size).

2.2.1.2 Support vector machines

Support vector machines (SVM) [Cristianini 2000, Burges 1998] is a group of models designed to find the optimal hyperplane separating two classes, or clustering one-class, in a multi-dimensional space. Support vector machines (SVM) was initially proposed by Vapnik and Lerner [Vapnik 1963], it attempts to find a compromise between the minimization

of empirical risk and the prevention of the overfitting. By applying a kernel trick, SVM can handle nonlinear classification problems [Boser 1992, Piciarelli 2008b, Cristianini 2000, Canu 2005].

The basic two class SVM can be generalized to multi-class decision problem (see for example [Pittore 1999] for an application of a multi-class SVM in office surveillance).

Based on the theory of SVM and the soft-margin trick [Schölkopf 2000, Ben-Hur 2002], one-class SVM is proposed to address the problem where only one category of samples (the positive samples) with a few outliers are available. In [Piciarelli 2008b, Piciarelli 2006, Piciarelli 2005, Piciarelli 2007, Piciarelli 2008a], the authors presented a method for anomalous event detection by means of trajectory analysis. The trajectories were subsampled to a fixed-dimension vector representation and clustered with an one-class support vector machines (SVM) algorithm. In these works, SVM classifiers are coupled with various feature representation methods including pixel-based [Pittore 1999], object based [Piciarelli 2008b, Piciarelli 2006, Piciarelli 2005, Piciarelli 2007, Piciarelli 2008a, Chen 2007]. In [Schuldt 2004], an algorithm constructed video representations in terms of local space-time features based on the silhouettes, integrated such representations with SVM classification schemes for recognition, the gestures of one person, such as walking, jogging, running, hand-waving, boxing and hand clapping were detected.

2.2.1.3 Neural networks

Neural networks is an another well know pattern recognition technique. It simulates the biological system by linking several decision nodes in layers. In [Vassilakis 2002], gesture recognition problems such as recognizing head movements were addressed by applying temporal data to both feedforward and generative feedback naturally static network models. In [Casey 2011], a neural network was used to model the superior colliculus (SC) to detect abnormalities in a panoramic image.

2.2.2 State event models

State event models are a class of techniques which are designed using semantic knowledge of the state of the video event in space and time. Reasonable assumptions about the nature of video events have been included in these technologies. State event models capture both the hierarchical nature and the temporal evolution of the state.

2.2.2.1 Finite-state machines

Finite state machine (FSM) is a deterministic formalism useful for modeling the temporal aspects of video events, it extends a state transition diagram with start and accept states to allow recognition of processes. The hidden Markov model (HMM) could be considered as a “probabilistic FSM”

In [Hongeng 2001], multi-agent event recognition was proposed, a single thread of action was recognized from the characteristics of the trajectory and moving blob of the actor by using finite state machine (FSM), a multi-agent event was represented by a number

of action threads related by temporal constraints, multi-agent events were recognized by propagating the constraints and likelihoods of event threads in a temporal logic network.

In [Medioni 2001], the moving regions in the sequence were detected and tracked, the trajectories together with additional information in the form of geo-spatial context and goal context were used to instantiate likely scenarios, in order to recognize aerial events.

2.2.2.2 Bayesian Networks

In order to deal with the uncertainty of observations existing in video events, Bayesian Networks are used. Bayesian Networks (BN) is a class of directed acyclic graphical models. Nodes in the BN represent random variables which may be discrete (finite set of states) or continuous (described by a parametric distribution). Conditional independence between these variables are represented by the structure of the graph [Jensen 2007, Pearl 1988]. BN achieves a probability score indicating how likely the event could occur given the input. A typical approach to anomaly detection is the basic latent Dirichlet allocation (LDA) model [Blei 2003]. LDA is a typical standard topic model which has been used to model video clips as being derived from a bag of topics drawn from a fixed (usually uniform) set of proportions [Popoola 2012]. Other Bayesian modeling approaches are probabilistic latent semantic analysis (pLSA) and hierarchical Dirichlet processes (HDP).

BN models do not have an inherent capacity of modeling temporal composition. Solutions to this problem include single-frame classification [Buxton 1995] and choosing abstraction schemes which encapsulate temporal properties [Lv 2006, Intille 1999].

Dynamic Bayesian Networks (DBN) benefits from a factorization of the state and the observation space, and the temporal evolution of state. DBN generalizes BN to a temporal context. It can be described formally by intra-temporal dependencies and inter-temporal dependencies.

2.2.2.3 Hidden Markov models

HMM is a class of directed graphical models extended to model the temporal evolution of the state. The HMM structure describes a model where the observations are dependent only on the current state. The state is only dependent upon the state at the previous "time slice" [Rabiner 1989, Ghahramani 1997].

In [Kosmopoulos 2010], multistream-fused HMM model (MFHMM) was used to recognize the real-life visual behavior understanding scenarios in a warehouse monitored by camera networks. In [Utasi 2010, Utasi 2008a, Utasi 2008b], Gaussian mixture model (GMM) and hidden Markov model (HMM) were used to detect the abnormal events of outdoor traffic areas based on the optical flow features. In [Jiménez-Hernández 2010], HMM model was used to identify uncommon motion events based on motion coding. Motion coding was similar to motion history image (MHI), it encoded the information and discovered the intrinsic dynamics using only the visual information. In [Kim 2009], a space-time Markov random field (MRF) model was proposed to detect abnormal activities in video. Optical flow features were extracted at each frame, and then a mixture of probabilistic principal component analyzers (MPPCA) was utilized to identify the typical patterns.

In [Benezeth 2011, Benezeth 2009], an approach using spatio-temporal models of scenes was presented. A Markov random field model parameterized by a co-occurrence matrix was built. Abnormal activities in the direction, speed and size of objects were detected. The work is similar to the change detection method when the background is not stable. In [Bregler 1997], low level primitive were areas of coherent motion found by expectation maximization (EM) maximum likelihood clustering, mid-level categories were simple movements represented by dynamical systems, and high-level complex gestures were represented by hidden Markov models (HMM) as successive phases of simple movements. Human gait was recognized. In [Jiang 2011, Jiang 2012], a context-aware method was proposed to detect anomalies, all moving objects in the video were tracked, a hierarchical data mining approach, the co-occurrence anomaly detection, considered as an observation sequence generated from hidden Markov model (HMM), was used to detect abnormal trajectories in the traffic scenes. In [Zhu 2011b, Zhu 2011a], the people in the parking lot were labeled by blobs, a clustering algorithm using hidden Markov models and latent Dirichlet allocation based (HMM-LDA based) on action words. A runtime accumulative anomaly was measured, an online likelihood ration test based (LRT-based) normal activity recognition method was proposed for online anomaly detection.

2.2.2.4 Conditional Random Fields

Conditional random field (CRF) is based on the idea that in a discriminative statistical framework only the conditional distribution is modeled. CRF is introduced in [Lafferty 2001], it is an undirected graphical model generalizing the hidden Markov model by putting feature functions conditioned on the global observation instead of the transition probabilities. Learning of CRF parameters can be achieved by using convex optimization methods such as conjugate gradient decent [Sutton 2007]. CRF based event detection offers several particular advantages including the abilities to relax strong independence assumptions in the state transition [Wang 2006]. In [Yao 2010], the authors developed a random field model using a structure learning method to learn important connectivity patterns between objects and human body parts. In [Wang 2006], the event was presented by semantic, conditional random field (CRF) was used to fuse temporal multi-modality cues for event detection in the football match scene.

2.2.3 Semantic event models

The semantic event models are usually applied when the events of interest are relatively complex with large variations in their appearance. These events can be described as a sequence of a number of states, they can be defined by semantic relationships between their composing sub-events. This type of approach allows the event model to capture high-level semantics such as long-term temporal dependence, hierarchy, partial ordering, concurrency and complex relations among sub-events and abstraction primitives.

2.2.3.1 Grammars

Grammar models [Aho 1972] specify the structure of video events as sentences composed of words corresponding to abstraction primitives, it has been used in computer vision [Chanda 2004]. The grammar formalism allows for mid-level semantic concepts (parts of speech in language processing). In the event model context, these mid-level concepts are used to model composing sub-events. This formalism naturally captures sequence and hierarchical composition as well as long-term temporal dependencies. A grammar model consists of three components: a set of terminals, a set of non-terminals and a set of production rules. Terminals correspond to abstraction primitives. Non-terminals correspond to semantic concepts. Production rules correspond to the semantic structure of the event. The recognition of an event is reduced to determining whether a particular video sequence abstraction (sequence of terminals) constitutes an instance of an event. This process is called parsing. The particular set of production rules used in recognizing the event is called the parse.

There are two extension models, stochastic grammars and attribute grammars. The stochastic grammars allow probabilities to be associated with each production rule, it can give a probability score to a number of legal parses [Stolcke 1995]. Attribute grammars associate conditions with each production rule, each terminal has certain attributes associated with it [Knuth 1968]. Stochastic grammars allow reasoning with uncertainty, attribute grammars allow further semantic knowledge to be introduced into the parsing process, it can describe constraints on features in addition to the syntactic structure of the input.

In [Calavia 2012], alarm detection in traffic was performed on the basis of the parameters of the moving objects and their trajectories by using semantic reasoning and ontologies. In [Antic 2011], the author parsed video frames by establishing a set of hypotheses that jointly explain all the foreground, and by trying to find normal training samples that explain the hypotheses. Abnormalities in the traffic scene were discovered indirectly as those hypotheses which were needed for covering the foreground without finding an explanation by normal samples for themselves. In [Ryoo 2006], a context-free grammar (CFG) based representation scheme was used to recognize two-people activities, which were represented as a composition of simpler actions and interactions. Eight types of interactions: approach, depart, point, shake-hands, hug, punch, kick and push were recognized. In [Joo 2006], the anomalies in a parking lot were detected by using attribute grammars, abnormal events were detected when the input did not follow syntax of the grammar or the attributes did not satisfy the constrains in the attribute grammar to some degree.

2.2.3.2 Petri Net

Petri Net (PN) formalism is a bipartite graph, which allows a graphical representation of the event model and can be used to naturally model non-sequential temporal relations as well as other semantic relations that often occur in video events. Place nodes are represented as circles and transition nodes are represented as rectangles. Place nodes hold tokens and transition nodes specify the movement of tokens between places when a state change occurs. Transition nodes are enabled if all input place nodes connected to that transition

node have tokens. In [Ghanem 2004, Ghanem 2007], events were composed by combining primitive events and previously defined events by spatial, temporal, and logical relations, these primitive events are then filtered by Petri Nets filters to recognize composite events of interest to recognize airports and traffic intersection events. In [Albanese 2008], a probabilistic Petri Net was proposed to recognize human activities in restricted settings such as airports, parking lots and banks, the minimal sub-videos in a given activity was identified with a probability above a certain threshold, and the activity from a given set with the highest probability was detected .

2.2.3.3 Constraint satisfaction

Constraint satisfaction is used to recognize the event as a set of semantic constraints on the abstraction. The event recognition task in this method is reduced to mapping the set of constraint to a temporal constraint network and determining whether the abstracted video sequence satisfies these constraints. Constraint satisfaction event models represent video events as a set of semantic constraints which include spatial, temporal and logical relationships. An event is then recognized by determining wether a particular video sequence abstraction is consistent with these constraints. In [Vu 2003, Vu 2004], the authors represented a scenario model by specifying the characters involved in the scenario, the sub-scenario composing the scenario and the constraints combining the sub-scenarios. Stores totally recognized scenarios (STRS) algorithm recognized usually a scenario by performing an exponential combination search. Stores partially recognized scenarios (SPRS) algorithms tried all combinations of actors to recognize “multi-actors” scenarios. In [Fusier 2007], a video understanding system based on scene tracking, coherence maintenance and scene understanding was proposed, the events in airport surveillance have been recognized.

2.2.3.4 Logic Approaches

In logic approaches, an event domain is specified as a set of logic predicates. A particular event is recognized using logical inference techniques such as resolution. These techniques are useful as long as the number of predicates, inference rules and groundings are kept low. In [Shet 2005, Shet 2006], the architecture of a visual surveillance system that combined real time computer vision algorithms with logic programming to represent and recognize activities involving interactions amongst people, pages and the environment through which they moved was described.

2.3 One-class classification

This section presents the theoretical framework of statistical learning theory. The early work comes back to 1960s, and becomes popular at 1990s since the support vector machines (SVM) have been proposed by Vapnik [Vapnik 2000, Vapnik 1998]. Brief introductions to this theory can be found in [Gunn 1998, Burges 1998, Bousquet 2004, Cristianini 2000].

In classification problems, the objective is to find the relation between each sample (input) and the tag (output). The linear models are firstly reascended, then, the kernel

trick extends the framework into a nonlinear setting, via reproducing kernel Hilbert spaces [Aronszajn 1950, Shawe-Taylor 2004].

2.3.1 Support vector machines for binary classification

Support vector machines (SVM) are initially proposed by Vapnik and Lerner [Vapnik 1963]. SVM for classification and regression provides a powerful tool for learning models that generalize well even in sparse, high dimension settings [Diehl 2003]. Traditional techniques for pattern recognition are based on the minimization of the empirical risk, which attempt to optimize the performance on the training set. SVM minimizes the structural risk, the probability of misclassifying patterns for a fixed but unknown probability distribution of the data [Pontil 1998]. It attempts to find a compromise between the minimization of empirical rick and the prevention of overfitting. By applying kernel trick, SVM can handle nonlinear classification problems [Boser 1992, Piciarelli 2008b, Cristianini 2000]. Consider the problem of separating the set of training data $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, $\boldsymbol{x} \in \mathbb{R}^d$ belong to two separate classes $y_i = \pm 1$, the constraint is that $y_i \varphi^*(\boldsymbol{x}_i) = 1$. In linear classification, the data are separated by a hyperplane,

$$\boldsymbol{w}^\top \boldsymbol{x}_i + \rho = 0, \tag{2.1}$$

where $\boldsymbol{w}$ is a vector, ρ is a constant.

The decision function for each datum $\boldsymbol{x}$ is:

$$\varphi(\boldsymbol{x}) = \mathbf{sgn}(\boldsymbol{w}^\top \boldsymbol{x}_i + \rho). \tag{2.2}$$

Assuming the minimization distance of the data to the separation plane is 1, one has:

$$\begin{cases} \boldsymbol{w}^\top \boldsymbol{x}_i + \rho \geq +1, y_i = +1, \\ \boldsymbol{w}^\top \boldsymbol{x}_i + \rho \leq -1, y_i = -1. \end{cases} \tag{2.3}$$

The two equations above can be rewritten as:

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + \rho) \geq 1. \tag{2.4}$$

The distance of each datum to the decision plane is:

$$d(\boldsymbol{x}) = \frac{y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + \rho)}{\|\boldsymbol{w}\|} \geq \frac{1}{\|\boldsymbol{w}\|}, \tag{2.5}$$

The problem maximizing the margin becomes minimizing $\|\boldsymbol{w}\|$ under constraints.

By introducing Lagrange multipliers α_i composing the vector $\boldsymbol{\alpha}$, the corresponding Lagrangian is,

$$L(\boldsymbol{w}, \rho, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + \rho) - 1). \tag{2.6}$$

Taking the derivatives of function (2.6) with respect to $\boldsymbol{w}$ and ρ, we have:

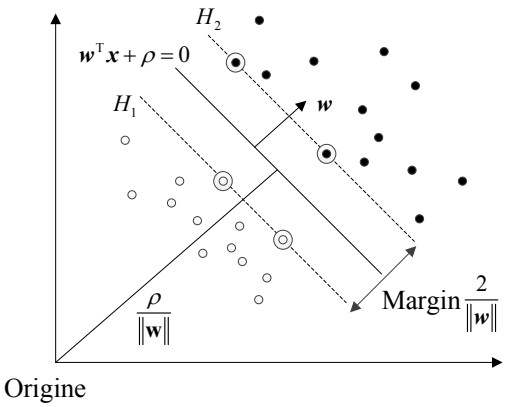


Figure 2.1: Principle of support vector machines for two classes classification. The support vectors are labeled by circle.

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \qquad \Rightarrow \boldsymbol{w}^\top = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i^\top, \tag{2.7}$$

$$\frac{\partial L}{\rho} = 0 \qquad \Rightarrow \sum_{i=1}^{n} y_i \alpha_i = 0. \tag{2.8}$$

Replace (2.7) (2.8) into (2.6), the optimization problem becomes:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j, \tag{2.9}$$

$$\text{subject to: } \sum_{i=1}^{n} \alpha_i y_i = 0, \alpha_i \geq 0. \tag{2.10}$$

This problem can be addressed by standard quadratical program method. Only few entries of $\boldsymbol{\alpha}$ are not 0, these correspond training samples are called support vector (SV). Once the $\boldsymbol{\alpha}$ are calculated, the optimal hyperplane is:

$$\boldsymbol{w} = \sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i, \tag{2.11}$$

$$\rho = -\frac{1}{2} \boldsymbol{w}^\top (\boldsymbol{x}_r + \boldsymbol{x}_s), \tag{2.12}$$

where $\boldsymbol{x}_r$ and $\boldsymbol{x}_s$ are any support vectors from each class satisfying:

$$\alpha_r, \alpha_s > 0, y_r = -1, y_s = +1. \tag{2.13}$$

As shown in Fig.2.1, the samples marked by circle in supplementary hyperplane H_1 and H_2 are support vectors. The hard margin classifier is then,

$$\varphi(\boldsymbol{x}) = \mathbf{sgn}\,(\boldsymbol{w}^\top \boldsymbol{x} + \rho) = \mathbf{sgn}\,(\sum_{i=1}^{n} \alpha_i y_i \boldsymbol{x}_i^\top \boldsymbol{x} + \rho). \tag{2.14}$$

Usually, the data cannot be separated linearly, it is needed to tolerate the errors of classification results of some samples. The error of classification of sample $\boldsymbol{x}_i$ is quantified by relaxation variable ξ_i, $\xi_i \geq 0$. The optimization problem becomes:

$$\min\ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n}\xi_i, \tag{2.15}$$

$$\text{subject to: } y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + \rho) \geq 1 - \xi_i,\ \ \xi_i \geq 0,\ \ i = 1, \ldots, n. \tag{2.16}$$

Address this optimization problem as the the method in hard margin classifier, we have:

$$\max\ \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^\top \boldsymbol{x}_j, \tag{2.17}$$

$$\text{subject to: } \sum_{i=1}^{n}\alpha_i y_i = 0,\ \ 0 < \alpha_i \leq C,\ \ i = 1, 2, \ldots, n. \tag{2.18}$$

The standard quadratical program is used to address this soft margin problem. In nonlinear situation, the scale product is replaced by a define positive kernel which implicitly transformers each sample by a nonlinear function. If an kernel κ is given, the decision function becomes:

$$\varphi(\boldsymbol{x}) = \mathbf{sgn}(\sum_{i=1}^{n}\alpha_i y_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + \rho). \tag{2.19}$$

2.3.2 Hyperplane one-class support vector machines

Based on the theory of SVM, one-class SVM is proposed to deal with problems that only one category of (the positive) samples are available. One-class SVM aims to determine a suitable region in the input data space $\mathcal{X}$ which includes mostly the samples drawn from an unknown probability distribution P. It detects objects which resemble training samples. Hyperplane based one-class SVM is the extended version of the original SVM to one-class problems [Schölkopf 2001], it is also be called as $\nu-$SVM. It identifies outliers by fitting a hyperplane from the origin, Fig.2.2 illustrates the hyperplane. Hyperplane one-class SVM is used as the one-class classification method in Chapter 3 and Chapter 4. The hyperplane one-class SVM is formulated as a constrained minimization optimization problem:

$$\begin{aligned} &\min_{\omega,\xi,\rho}\ \frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + C\sum_{i=1}^{n}\xi_i, \\ &\text{subject to: } \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle \geq \rho - \xi_i,\ \ \xi_i \geq 0, \end{aligned} \tag{2.20}$$

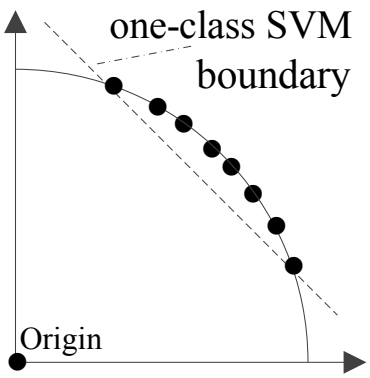


Figure 2.2: The decision hyperplane of one-class SVM divides the data in the feature space.

where $\boldsymbol{x}_i \in \mathcal{X}$, $i \in \{1, \ldots, n\}$ are n training samples in the input data space $\mathcal{X}$, ξ_i is the slack variable for penalizing the outliers. The hyperparameter C is the weight for restraining slack variable, it tunes the number of acceptable outliers. $\|\cdot\|$ denotes Euclidean norm of a vector. $\langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i) \rangle - \rho = 0$ is the decision hyperplane. $\boldsymbol{w}$ defines a hyperplane in feature space separating the coordinate origin from the projections of training data. The nonlinear function $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ maps datum $\boldsymbol{x}_i$ from the input space $\mathcal{X}$ into the feature space $\mathcal{H}$, which allows to solve a nonlinear classification problem by designing a linear classifier in the feature space $\mathcal{H}$. For computing dot products in $\mathcal{H}$, the positive definite kernel function κ is defined as $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle$ to implicitly map the training or testing data $\boldsymbol{x}$ into a higher (possibly infinite) dimensional feature space and compute the dot product. Introducing the Lagrangian multipliers α_i, the decision function in the input data space $\mathcal{X}$ is defined as:

$$f(\boldsymbol{x}) = \mathbf{sgn}(\sum_{i=1}^{n} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) - \rho). \tag{2.21}$$

When $f(\boldsymbol{x}) = -1$, the datum $\boldsymbol{x}$ is classified as anomaly, otherwise $\boldsymbol{x}$ is considered as normal.

If proper parameters are given, classical kernels, such as Gaussian, polynomial, and sigmoidal kernel, have similar performances [Schölkopf 2002]. Gaussian kernel is chosen for handling spatial features in our work. It is a semi-positive definite kernel that contents Mercer condition [Vapnik 2000, Vapnik 1998]. Gaussian kernel is defined as the following expression:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}), \qquad (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{X} \times \mathcal{X}, \tag{2.22}$$

where $\boldsymbol{x}_i, \boldsymbol{x}_j$ are the data in the original data space $\mathcal{X}$, the variance σ indicates the scale factor at which the data should be clustered.

2.3.3 Hypersphere one-class support vector machines

Hypersphere one-class SVM was proposed in [Tax 2001, Tax 2004], it identifies outliers by fitting a hypersphere with a minimal radius, it is also be called support vector data description (SVDD). The problem can be written as the following objective function to be minimized:

$$\min_{R,c,\xi} \quad R^2 + C\sum_{i=1}^{n}\xi_i, \tag{2.23}$$

$$\text{subject to: } \|\Phi(\boldsymbol{x})_i - \boldsymbol{c}\| \leq R^2 + \xi_i, \quad i = 1, 2, \ldots, n. \tag{2.24}$$

By introducing the Lagrange multipliers $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$, the Lagrangian becomes:

$$L(\boldsymbol{c}, R, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\gamma}) = R^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(R^2 + \xi_i - \|\Phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2) - \sum_{i=1}^{n}\gamma_i\xi_i. \tag{2.25}$$

By KKT conditions, we have:

$$\sum_{i=1}^{n}\alpha_i = 1, \tag{2.26}$$

$$\boldsymbol{c} = \sum_{i=1}^{n}\alpha_i\boldsymbol{x}_i, \tag{2.27}$$

$$C - \alpha_i - \gamma_i = 0, \quad i = 1, 2, \ldots, n. \tag{2.28}$$

α_i is obtained by:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{n}\alpha_i\,\kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{n}\alpha_i\alpha_j\,\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{2.29}$$

$$\text{subject to: } \sum_{i=1}^{n}\alpha_i = 1, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \ldots, n. \tag{2.30}$$

Each sample $\boldsymbol{x}_i$ is classified into 3 categories: the samples with $\alpha_i = C$ are outside the sphere, with $0 < \alpha_i < C$ are on the sphere, with $\alpha_i = 0$ are inside the sphere. The samples with $\alpha_i \neq 0$ are called support vector (SV), they can be expressed as $i \in \mathcal{I}_{sv}$. The radius is computed as:

$$R = \min_{i \in \mathcal{I}_{sv}} \|\Phi(\boldsymbol{x}_i) - \boldsymbol{c}\|. \tag{2.31}$$

For classifying each sample, the distance is $dis = \|\Phi(\boldsymbol{x}) - \boldsymbol{c}\|$. If $dis < R$, the sample is normal. The distance is computed as:

$$\|\Phi(\boldsymbol{x}) - \boldsymbol{c}\|^2 = \sum_{i,j \in \mathcal{I}_{SV}} \alpha_i \alpha_j \, \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) - 2 \sum_{i \in \mathcal{I}_{SV}} \alpha_i \, \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + \kappa(\boldsymbol{x}, \boldsymbol{x}). \tag{2.32}$$

Fig.2.3 illustrates the hyperplane one-class SVM (ν-SVC) and hypershpere one-class SVM (or support vector data description, SVDD).

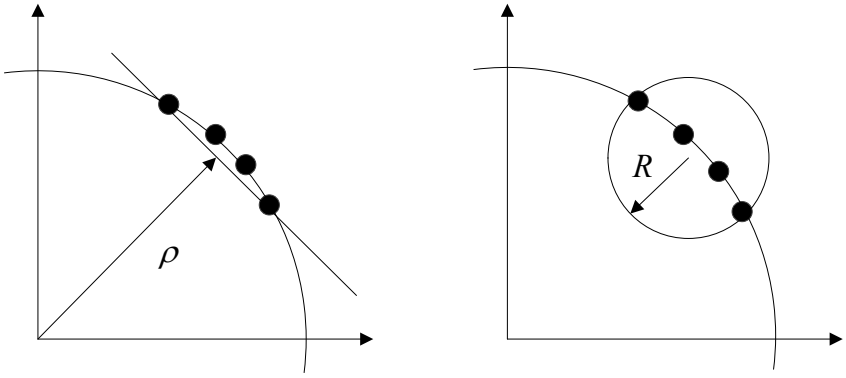


Figure 2.3: Data descriptions by the ν-SVC and the SVDD where the data is normalized to unit norm [Tax 2001]. ν-SVC is for hyperplane one-class SVM or one-class SVM [Schölkopf 2001]. SVDD is for hypershpere one-class or support vector data description.

2.3.4 Kernel PCA for abnormal detection

Kernel PCA extends standard principal component analysis (PCA) to a nonlinear setting. Before performing a PCA, one can map the n data points $\mathbf{x}_i \in \mathbb{R}^d$ to a higher-dimensional feature space $\Phi(\boldsymbol{x}_i) \in \mathcal{H}$ where standard PCA is performed [Hoffmann 2007]:

$$\boldsymbol{x}_i \rightarrow \Phi(\boldsymbol{x}_i). \tag{2.33}$$

This mapping can be omitted by adopting a kernel function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$, which replaces the scalar product $\langle(\Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x}')\rangle$. In kernel PCA, an eigenvector $\mathbf{V}$ of the covariance matrix in $\mathcal{H}$ is a linear combination of points $\Phi(\mathbf{x}_i)$:

$$\mathbf{V} = \sum_{r=1}^{n} \alpha_i \widetilde{\Phi}(\boldsymbol{x}_i), \tag{2.34}$$

$$\begin{aligned} \widetilde{\Phi}(\boldsymbol{x}_i) &= \Phi(\boldsymbol{x}_i) - \Phi_0 \\ &= \Phi(\boldsymbol{x}_i) - \frac{1}{n} \sum_{r=1}^{n} \Phi(\boldsymbol{x}_r), \end{aligned} \tag{2.35}$$

where the α_i are the components of a vector $\boldsymbol{\alpha}$, which is an eigenvector of the Gram matrix $\widetilde{K}_{ij} = \langle \widetilde{\Phi}(\boldsymbol{x}_i) \cdot \widetilde{\Phi}(\boldsymbol{x}_j)\rangle$. Φ_0 is the center of the data.

For abnormal (novelty) detection, the reconstruction error in the feature space $p(\widetilde{\boldsymbol{\Phi}}(\mathbf{x}))$ is defined as:

$$p(\widetilde{\Phi}(\boldsymbol{x})) = \langle\widetilde{\Phi}(\boldsymbol{x})\cdot\widetilde{\Phi}(\boldsymbol{x})\rangle - \langle(\widetilde{\Phi}(\boldsymbol{x})\cdot\mathbf{V}^l)\cdot(\widetilde{\Phi}(\boldsymbol{x})\cdot\mathbf{V}^l)\rangle \tag{2.36}$$

with

$$\begin{aligned}
&\langle\widetilde{\Phi}(\boldsymbol{x})\cdot\widetilde{\Phi}(\boldsymbol{x})\rangle \\
&= \langle(\Phi(\boldsymbol{x})-\Phi_0)\cdot(\Phi(\boldsymbol{x})-\Phi_0)\rangle \qquad (2.37)\\
&= \kappa(\boldsymbol{x},\boldsymbol{x}) - \frac{2}{n}\sum_{i=1}^{n}\kappa(\boldsymbol{x},\boldsymbol{x}_i) + \frac{1}{n^2}\sum_{i,j=1}^{n}k(\boldsymbol{x}_i,\boldsymbol{x}_j),\\
&\langle\widetilde{\Phi}(\boldsymbol{x})\cdot\mathbf{V}^l\rangle \\
&= \sum_{i=1}^{n}\alpha_i^l[\kappa(\boldsymbol{x},\boldsymbol{x}_i) - \frac{1}{n}\sum_{r=1}^{n}k(\boldsymbol{x}_i,\boldsymbol{x}_r) - \qquad (2.38)\\
&\frac{1}{n}\sum_{r=1}^{n}\kappa(\boldsymbol{x},\boldsymbol{x}_r) + \frac{1}{n^2}\sum_{r,s=1}^{n}\kappa(\boldsymbol{x}_r,\boldsymbol{x}_s)],
\end{aligned}$$

where $\langle\widetilde{\Phi}(\boldsymbol{x})\cdot\widetilde{\Phi}(\boldsymbol{x})\rangle$ is the potential of a point $\mathbf{x}$ in the original space, which is computed by the squared distance from the mapping $\widetilde{\Phi}(\boldsymbol{x})$ to the center Φ_0. The index l denotes the l-th eigenvector, with $l = 1$ for the eigenvector with the largest eigenvalue. $\langle\widetilde{\Phi}(\boldsymbol{x})\cdot\mathbf{V}^l\rangle$ is the projection of $\widetilde{\Phi}(\boldsymbol{x})$ onto the eigenvector $\mathbf{V}^l$.

If only first q rows of vector $\mathbf{V}^l$ are chosen, the reconstruction error of the original space data $\boldsymbol{x}$ can be expressed as:

$$p(\boldsymbol{x}) = \widetilde{\Phi}(\boldsymbol{x})^2 - \sum_{l=1}^{q}(\widetilde{\Phi}(\boldsymbol{x})\cdot\mathbf{V}^l)^2. \tag{2.39}$$

All the components in the eq. (2.39) can be computed by the kernel function while the data are in the original space.

2.4 Conclusion

The event understanding process can be generally decomposed into two parts, abstraction and event modeling, respectively. Abstraction is the organization of low-level video sequence data into intermediate units that capture salient and discriminative abstract properties of the video data. Event modeling is defined as the representation of occurrences of interest, using those units ("primitives") generated by the abstraction of the video sequence, in such a way that allows recognition of these events as they occur in unlabeled video sequences [Lavee 2009a]. Hyperplane one-class support vector machines (one-class SVM, or OC-SVM, or ν-SVC) method is used in Chapter 3 and Chapter 4. Chapter 5 has two parts. Hypersphere one-class support vector machines (support vector data description, or SVDD) based online algorithm is used in the first part of Chapter 5. Least squares one-class support vector machines (LS-OC-SVM) based online algorithm is used in the second part of Chapter 5.

Chapter 3

Abnormal detection based on optical flow and HOFO

Contents

Because abnormal visual events are mainly characterized by objects movements and interactions in the scene, the optical flow is chosen as the low-level features based on which various descriptors and classifiers could be designed to efficiently detect abnormal events. Also, because only normal-event video sequences are available, variants of nonlinear one-class support vector machines (OC-SVM) are used as classification algorithms. It is worth noting that the proposed detection methods do not require a prior step of object tracking in the scene, which makes it very efficient in practical situations.

The rest of the chapter is organized as follows. In Section 3.1, the abnormal events detection method based on optical flow is introduced. In Section 3.2, after presenting an efficient technique to extract the foreground, abnormal detection is locally applied to detect abnormal blobs (abnormal moving objects). In Section 3.3, the proposed histograms of optical flow orientation (HOFO) descriptor is described. Further, the fast version of the detection algorithm is designed by fusing the optical flow computation with a background subtraction step. Finally, Section 3.4 concludes this chapter.

3.1 Abnormal detection based on optical flow

3.1.1 Feature selection

The optical flow can provide important information about the spatial arrangement of the objects and the change rate of this arrangement [Horn 1981]. It is the apparent velocity distribution of brightness patterns movement in an image. B.Horn and B. Schunck [Horn 1981] proposed an algorithm computing the optical flow by introducing a global constraint of smoothness. We adopt the Horn-Schunck (HS) optical flow method combining a data term with a spatial term. The data term assumes constancy of the same image property, and the expected flow variation is modeled by the spatial term. The optical flow is formulated as the minimization of the following global energy functional:

$$E = \int\int [(I_x u + I_y v + I_t)^2 + \alpha^2(\|\nabla u\|^2 + \|\nabla v\|^2)]dxdy, \tag{3.1}$$

where I_x,I_y and I_t are the derivatives of the image intensity along the x, y and time t dimension, u and v are the horizontal and vertical components of the optical flow, α is the parameter representing the weight of the regularization term. Lagrange equations are utilized to minimize the functional E, yielding:

$$\begin{cases} I_x(I_x u + I_y v + I_t) - \alpha^2 \triangle u = 0 \\ \\ I_y(I_x u + I_y v + I_t) - \alpha^2 \triangle v = 0, \end{cases} \tag{3.2}$$

subject to

$$\begin{cases} \triangle u(x,y) = \overline{u}(x,y) - u(x,y) \\ \\ \triangle v(x,y) = \overline{v}(x,y) - v(x,y), \end{cases} \tag{3.3}$$

where $\overline{u}$ and $\overline{v}$ are weighted averages of u and v calculated in a neighborhood around the pixel location. The optical flow is computed in an iterative scheme as shown below:

$$\begin{cases} u^{k+1} = \overline{u}^k - \frac{I_x(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \\ \\ v^{k+1} = \overline{v}^k - \frac{I_y(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2}, \end{cases} \tag{3.4}$$

where k denotes the algorithm iteration. A single time step was taken for normal scene and abnormal scene, so that the computations are based on just two adjacent images.

3.1.2 Abnormal detection method

In this subsection, we describe a method of detecting abnormal events based on optical flow in video streams. Assume that a set of frames $\{I_1, I_2, \ldots, I_n\}$ in which the person is walking or loitering, are considered as normal events. The frames in which the person is running

or walking with a sudden split are regarded as abnormal events. In abnormal detection problem, it is assumed that the data from only one class, the positive class (or the normal scene), are available. The one-class SVM frameworks is then suitable to the specificity of the abnormal event detection problem where only normal scene samples are available. The general architecture of the abnormal detection method is presented in Fig.3.1, and outlined in the following.

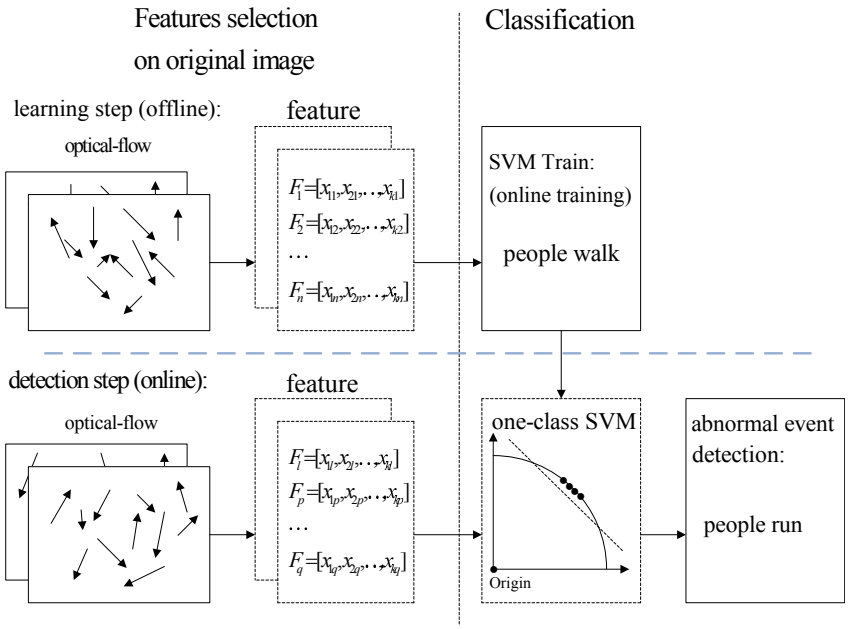


Figure 3.1: Major processing states of the proposed one-class SVM abnormal frame events detection method. The optical flow features is constructed.

Step 1: The first step consists of computing the optical flow features at gray scale image. Each training frame is processed via Horn-Schunck (HS) optical flow algorithm to get the moving features at every pixel. This step can be presented as the following:

$$\{I_1, I_2, \ldots, I_n\} \xrightarrow{HS} \{OP_1, OP_2, \ldots, OP_n\}, \tag{3.5}$$

where $\{I_1, I_2, \ldots, I_n\}$ are the training original images, $\{OP_1, OP_2, \ldots, OP_n\}$ are the corresponding optical flow.

Step 2: One-class SVM is used to classify feature samples of incoming video frames. Three strategies are proposed for obtaining the features of the image. The sketch image for choosing the features is shown in Fig.3.2.

Method 1: Take the optical flow at each pixel of the image as feature samples, as shown in Fig.3.2(a). In the dataset UMN [UMN 2006], define the movement of walking as the normal event, running as abnormal event. The video sequence in our work is labeled as normal and abnormal for performance evaluation. Training data for one-class SVM are extracted from the normal images. Take the optical flow $OP_{i,j,k}$ as feature $F_{i,j,k}$ for (i, j)-th pixel on the k-th frame. For each point at Cartesian coordinate (i, j) of n training frames, we

(a) Pixel by pixel (b) Block by block (c) Block^{all} by block

Figure 3.2: Three strategies for choosing the optical flow features. (a) Choose the features pixel-by-pixel. (b) Choose the features block-by-block. (c) Choose all the blocks in the frame as the training sample, and test by block.

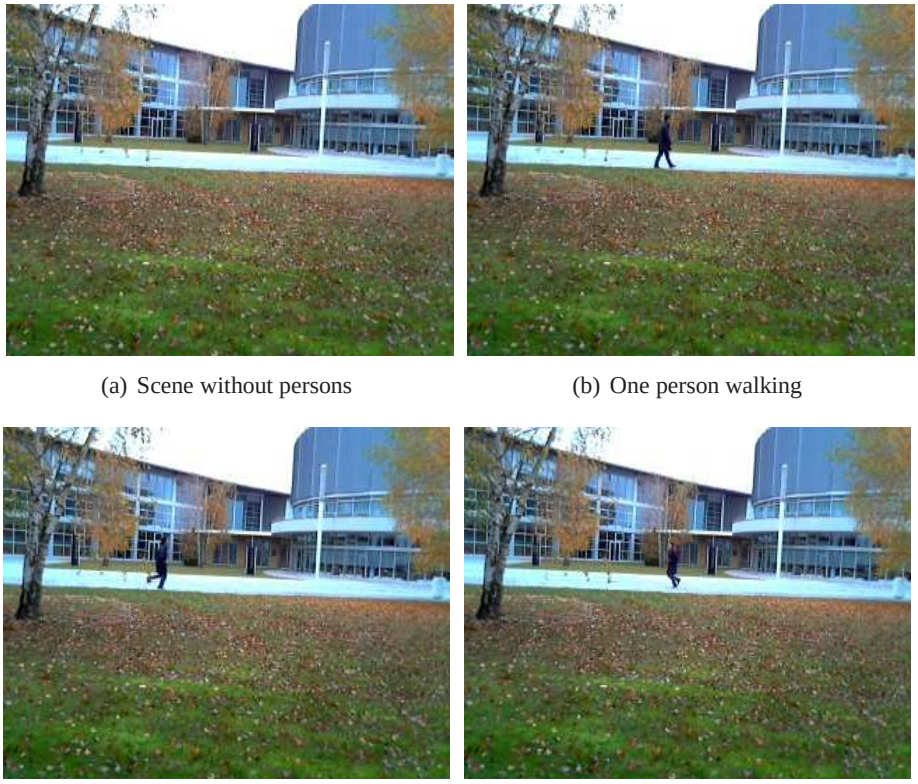
can get the training samples $F_{i,j,1\ldots n}$, $n \geq 1$, and then compute the support vectors. Based on the support vectors, the incoming samples $F_{i,j,n+1\ldots m}$ at coordinate (i, j) are classified. For the whole image, the abnormal events are detected pixel-by-pixel.

Method 2: Take the optical flow of all points in the block as feature samples. In this strategy, the image is segmented into several blocks, as shown in Fig.3.2(b), the image is separated into $p \times q$ blocks, p is the number of blocks at the vertical (height) dimension of the image, q is the number of blocks at the horizontal (length) dimension of the image. The height of the block is h pixels, the length of the block is w pixels, there are $h \times w$ points in the *block*. The feature of *block* at i-th row and j-th column in the k-th frame is noted as $F^{\text{block}}_{i,j,k}$. For each block, the feature F^{block} is arranged by the optical flow of all the points in the form $\{OP_1, OP_2, OP_3, \cdots, OP_{h\times w}\}$. For the video streams, take the features of block in the normal images as the training samples for one-class SVM, and then abnormal events are detected block-by-block.

Method 3: The image is also split into blocks, but the training samples are all the blocks at one frame, as shown in Fig.3.2(c). Similar to *Method 2*, we separate one frame into $p \times q$ blocks, the size of each block is $h \times w$. At k-th frame , the feature sample of all the blocks on this frame is $\{F^{\text{block}}_{1,1,k}, F^{\text{block}}_{1,2,k}, \ldots, F^{\text{block}}_{p,q,k}\}$, a vector of dimension $(p \times q) \times (h \times w)$. To get the training data in the normal frame from 1-st to n-th, the data are arranged as $\{F^{\text{block}}_{1,1,1}, F^{\text{block}}_{1,2,1}, \ldots, F^{\text{block}}_{p,q,1}, \ldots, F^{\text{block}}_{1,1,k}, \ldots, F^{\text{block}}_{p,q,k}, \ldots, F^{\text{block}}_{1,1,n}, \ldots, F^{\text{block}}_{p,q,n}\}$, a vector of dimension $(p \times q \times k) \times (h \times w)$. For abnormal detection, the test sample is the feature of one block.

The sequence which just has one person is taken as an example for detailing the algorithm performance. The scene is presented in Fig.3.3. Four pictures in Fig.3.3 show the scene without people, the person walking and the person running at different directions. The training sequence, where the person is walking, learnt by SVM is shown in Fig.3.3(b). The detected sequence, where the person is running, is shown in Fig.3.3(c)(d). The results of these three strategies are shown in Fig.3.4. In Fig.3.4(b)(c), the abnormal detections on the background are marked by white circles, they are taken as false alarms. The detection result via pixel-by-pixel feature selection strategy has more false alarms than others. Because pixel-by-pixel strategy takes the feature at one pixel, it is more susceptive to the optical flow changing. The feature chosen by block can get better detection result than

pixel-by-pixel result. The block-by-block strategy which is shown in Fig.3.4(c) take each block as the local monitor, it considers the situation of several pixels. The block-by-block strategy is more robust than pixel-by-pixel strategy. Taking all the blocks on the image as the training samples has no false alarms and has similar detected results on the person.



(a) Scene without persons (b) One person walking

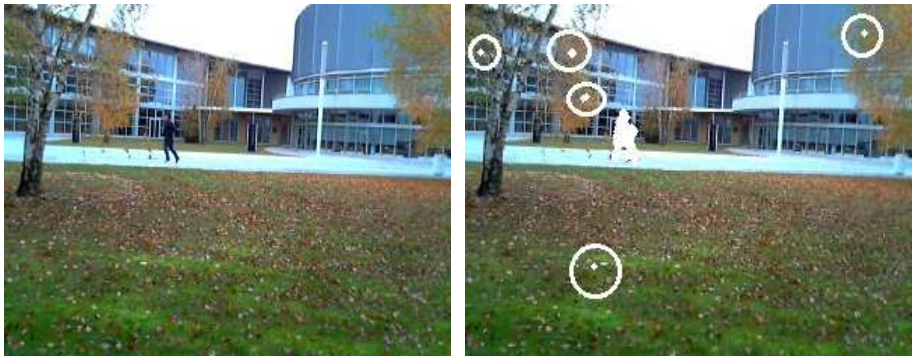
(c) One person running (d) One person running

Figure 3.3: Video stream of one person walking and running. (a) The scene without persons. (b) One person is walking. (c) One person is running. (d) The person is running toward another direction.

Step 3: As the objective of abnormal event detection problem is to analyze human action, the SVM detection result can be combined with foreground detection which extracts moving objects. The abnormal detections on the background can be deleted, they are considered as noise of the detection results. The background subtraction method presented by O. Tuzel et al.[Tuzel 2005, Porikli 2005] is adopted. Then, optical flow one-class SVM classification results and the foreground information are fused. When the points or blocks are detected as anomaly and also from the foreground, they are detected as abnormal finally.

Step 4: After acquiring detection results of each point or each block, then the decision of global frame anomaly is detected by presetting a number as threshold. If the number of abnormal points or blocks is larger than the threshold, the frame is considered as an abnormal one.

Case 1: If there are no abnormal detected points or blocks in the frame, this frame is



(a) One persons running (b) Result pixel by pixel

(c) Result block by block (d) Result block^{all} by block

Figure 3.4: Abnormal detection results of one person walking and running scene based on three optical flow feature selection strategies via one-class SVM. (a)One person is running. (b)Detection result via *Method 1*, pixel-by-pixel. (c)Detection result via *Method 2*, block-by-block. (d)Detection result via *Method 3*, training sample is all blocks on whole image.

considered as a normal one.

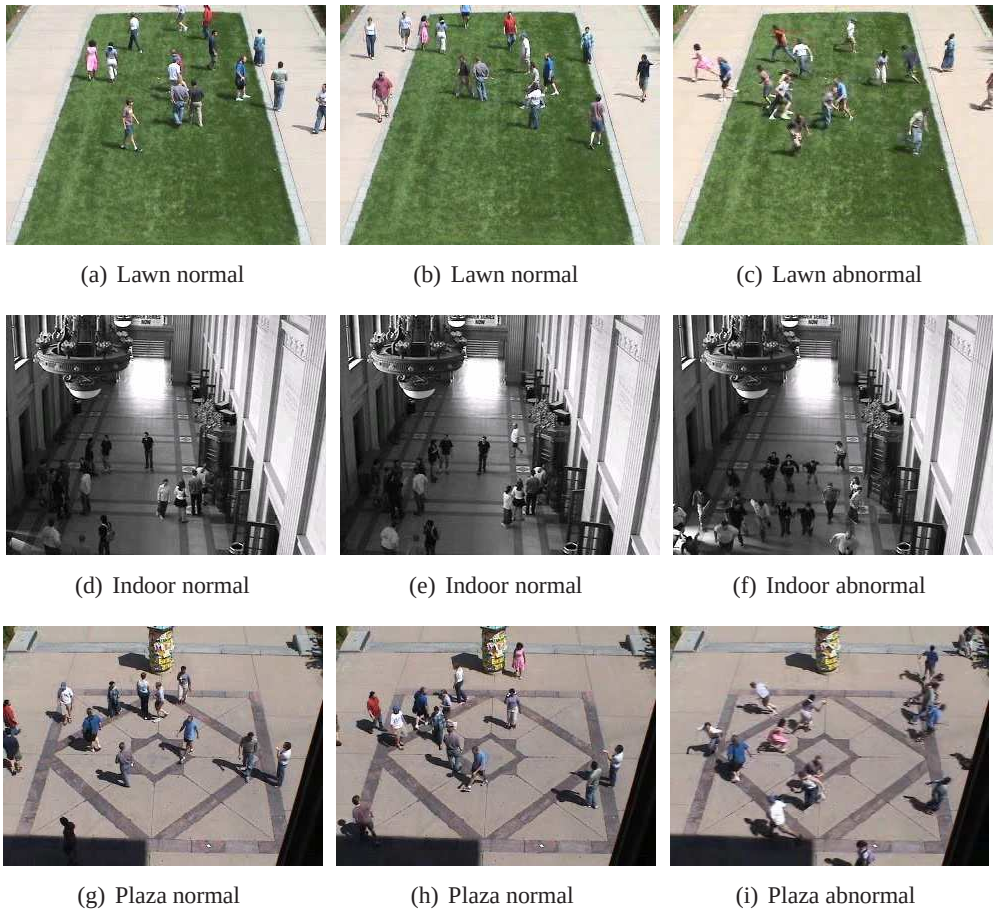
Case 2: If the number of abnormal points or blocks in the frame exceeds the threshold but this frame is labeled as a normal one, the detection result of the whole image via one-class SVM is considered as a false alarm.

Case 3: If the number of abnormal points or blocks on the frame exceeds the threshold and this frame is labeled as an abnormal one, then the detected result via one-class SVM is considered as a true positive.

3.1.3 Experimental Results

This section presents the results of experiments conducted to analyze the performance of the proposed method of detecting abnormal events based on optical flow features. The normal and abnormal scenes are shown in Fig.3.5.

The detection results of the lawn scene are shown in Fig.3.6. The points marked with white color are the abnormal detections via OC-SVM, the points marked with cyan color are the abnormal detections and also on the foreground. In Fig.3.6(b)(c), the abnormal



(a) Lawn normal (b) Lawn normal (c) Lawn abnormal

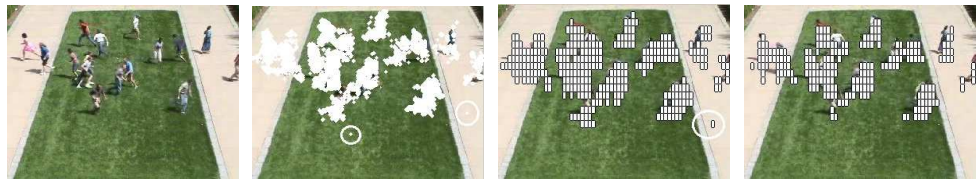
(d) Indoor normal (e) Indoor normal (f) Indoor abnormal

(g) Plaza normal (h) Plaza normal (i) Plaza abnormal

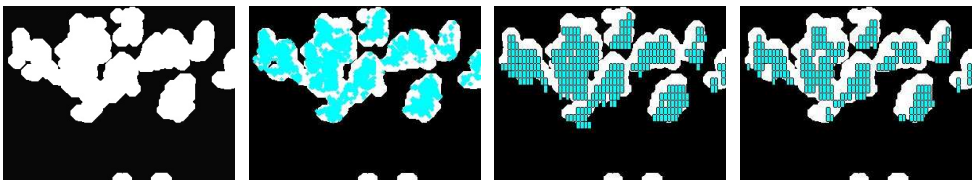
Figure 3.5: The lawn, indoor and plaza scenes of UMN dataset. (a)(b)(c) The first row is lawn scene. (d)(e)(f) The second row is scene indoor. (g)(h)(i) The third row is plaza scene. (a)(b)(d)(e)(g)(h) Normal events, all the persons are walking. (c)(f)(i) Abnormal events, all the persons are running.

detection results on the background are marked by white circles. Fig.3.6(d) is the result taking all blocks at the whole image as the training samples, it has the best performance.

We present one special situation of the abnormal events in the lawn scene. As presented in Fig.3.7, when most people are running, in the lower half part of the image, one person is walking. The walking person is cut out from the walking sequence at UMN dataset. The detected results of this special situation are shown in Fig.3.7. The pixel-by-pixel and block-by-block feature selection strategies detect the walking person as abnormal. These two strategies model the movement of pixel or block at the fixed positions in the frame. At the lower half part of image, there are no people on the training sequence, so the walking person is regarded as an abnormal event. The appropriate strategy should be chosen by depending on the application. If the region is “no admittance”, the walking person in this region is abnormal. The feature selected strategy can be pixel-by-pixel or block-by-block. If only the running movement is abnormal, the strategy for feature selection should take



(a) Original image (b) Pixel by pixel (c) Block by block (d) Block^{all} by block



(e) Foreground (f) P-by-p on fg (g) B-by-b on fg (h) B^{all}-by-b on fg

Figure 3.6: Abnormal *frame* detection results of the lawn scene based on three optical flow feature selection strategies via one-class SVM. (a)The original image. (b)The abnormal detection via pixel-by-pixel. (c)The abnormal detection via block-by-block. (d)The abnormal detection via taking all the blocks on the whole image as training samples. (e)The dilative foreground of the image. (f)The abnormal detection via pixel-by-pixel and also on the foreground. (g)The abnormal detection via block-by-block and also on the foreground. (h)The abnormal detection via taking all blocks on the whole image as training samples and also on the foreground.

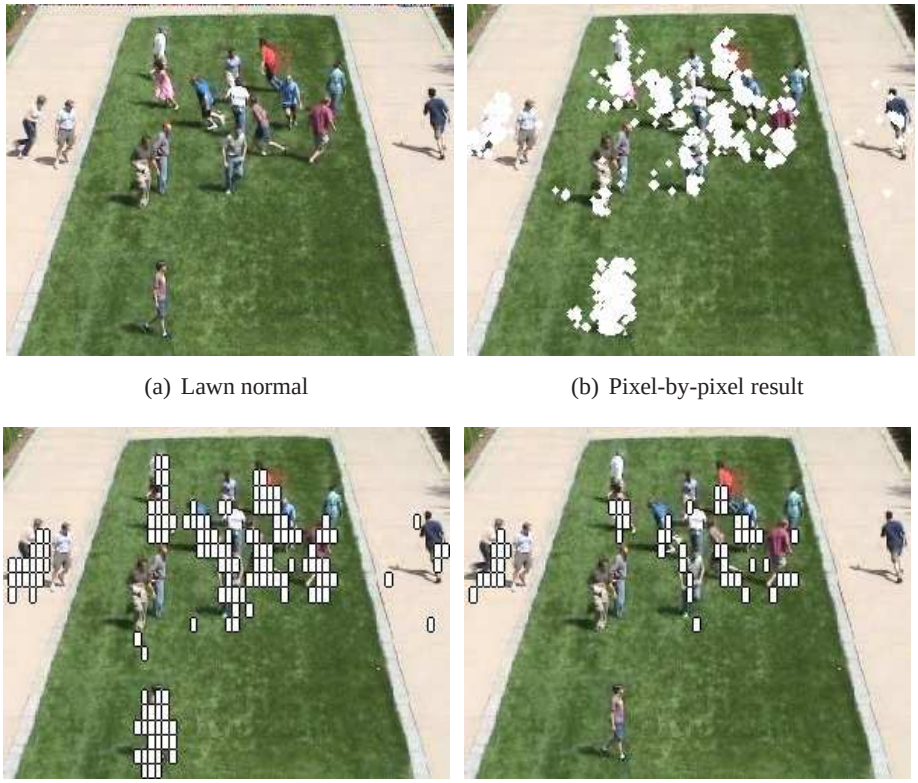
all the blocks on the whole image as training samples. Fig.3.7(d) has the less abnormal detections. Because the feature selection strategy taking all the blocks in the image as training samples considers an overall situation, it is the most robust and least sensitive. In Fig.3.7(b)(c)(d), the abnormal detection results are not on all the persons. Because the frame is the beginning of the running sequence, the optical flow is not much different from walking. Some parts of these persons are detected as normal.

The abnormal detected results of indoor scene and plaza scene are shown in Fig.3.8. The detection results show that the pixel-by-pixel feature selection strategy is the most sensitive method for abnormal events detection. While taking the blocks at the whole image as the training samples is the most robust method.

Performance summary on the UMN dataset compares with paper [Haque 2010] is in TABLE 3.1. For these three scenes, we get approximative detection rate with paper [Haque 2010], and the false alarms are reduced.

3.2 Blob extraction

In case of a stationary camera, the moving object segmentation becomes feasible due to a background subtraction algorithm. The foreground of each frame is obtained by the background subtraction method presented by O. Tuzel et al. [Tuzel 2005]. The moving objects are usually conflicted with others. As shown in Fig.3.9(a), the running person on



(a) Lawn normal (b) Pixel-by-pixel result

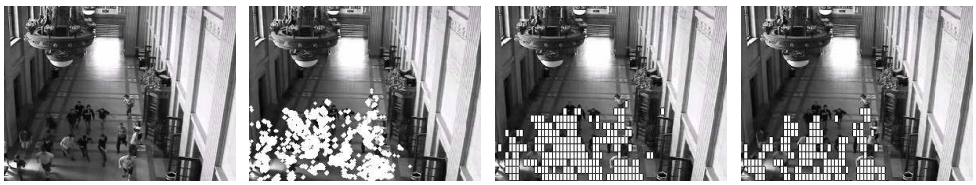
(c) Block-by-block result (d) Block$^{\text{all}}$-by-block result

Figure 3.7: Abnormal *frame* detection results of a special situation of the lawn scene based on three optical flow feature selection strategies via one-class SVM. (a)The original image of one person walks on the lower part of the image. (b)The abnormal detection by pixel-by-pixel strategy. (c)The abnormal detection by block-by-block strategy. (d)The abnormal detection by taken all the blocks on the whole image as training sample.

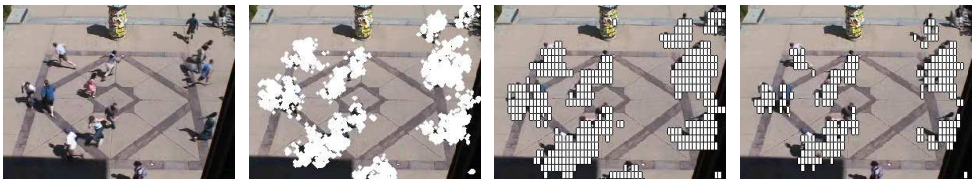
the upper half in the 1-st rectangle is overlapped with another walking person. The running person is moving from the right to left; the walking person is moving from the left to right. We present a method to improve the blob extraction performance by adopting optical flow, which presents the moving information. The method is summarized in Algorithm 3.9, and explained below in detail.

Step1: The first step consists of labeling connected components from a binary foreground image. Denote B_{FG}^{k} for the k-th blob in the foreground image. Because there are usually occlusions of the people, some rectangles contain several objects. As shown in Fig.1(a), the 1-st rectangle includes two people.

Step2: The second step is labeling the blobs based on the optical flow. If the size of the foreground blob is bigger than a presetting threshold T_{blb}, the optical flow in this area is taken into account to refine the blob extraction. T_{blb} is set with respect to the scene to represent the size of one person. In the mall scene, the size of the image is 240×320, T_{blb} is set as 50×100. As the action of the people can be exhibited by the direction and the



(a) Indoor (b) Point by point (c) Block by block (d) Block^{all} by block



(e) Plaza (f) Point by point (g) Block by block (h) Block^{all} by block

Figure 3.8: Abnormal *frame* detection results in the indoor and plaza scenes based on three optical flow feature selection strategies via one-class SVM. (a) The original image of indoor scene. (e) The original image of scene on the plaza. (b)(f) The abnormal detections by pixel-by-pixel strategy. (c)(g) The abnormal detections by block-by-block strategy. (d)(h) The abnormal detections taken all the blocks in the image as training samples.

	DR[6]	FPR[6]	DR	FPR
lawn	100%	0%	100%	0%
indoor	80%	12%	99.4%	1%
plaza	100%	4%	100%	2%

Table 3.1: The comparison of our proposed optical flow features and one-class SVM based method with the state-of-the-art methods for abnormal *frame* events detection of UMN dataset. DR="detection rate", FPR="false positive rate". The last two columns are the statistic results of the proposed method.

amplitude of the movement, the optical flow is chosen as the scene description. The optical flow algorithm introduced by Sun *et al.* [Sun 2010] is used in our work. It is a modified method of the formulation of Horn and Schunck [Horn 1981] allowing higher accuracy by using weights according to the spatial distance, brightness, occlusion state, and median filtering.

In the proposed method, we generate a color image I_{OP} from the optical flow, as shown in Fig.3.9(c), the mean-shift algorithm [Comaniciu 2002, Cheng 1995] is used to cluster each channel of the optical flow image into different patches. If the difference of the speed is larger than the bandwidth parameter, which is set as 0.2, in the mean-shift algorithm, these two objects can be distinguished. This blob labeling method can not only be used to distinguish different directions, but also be used suitably to distinguish two conflicted objects moving in the same direction with different speeds.

Step3: The third step consists of applying non-maximum suppression (NMS) algorith-

(a) (b) (c)

Figure 3.9: The blobs of the objects before and after our proposed blob extraction method. (a) 2 extracted blobs based on the foreground template. (b) 3 extracted blobs via the proposed blob extraction method, which is based on the foreground template and the optical flow. (c) The optical flow image of Fig.(a)(b). A black border is added to illustrate the image clearly.

Algorithm 1 Blob extraction.

Require:

Foreground image FG, optical flow OP

1: Label the separate blobs in FG, the blob of foreground image B^k_{FG} is obtained.
2: **if** Blob size in $FG \geq$ presetting size T_{blb} **then**
3: Draw the optical flow image I_{OP} in this blob.
4: The optical flows with similar magnitudes and directions are clustered by mean-shift algorithm.
5: Delete redundance cluster by NMS algorithm, blob of optical flow image B^i_{OP} is obtained. The remaining part of the blob $B_{\mathrm{RM}} = B_{\mathrm{FG}} - B_{\mathrm{OP}}$.
6: Traverse B_{RM} by a rectangle template to find the blobs overlapped by the foreground. NMS algorithm is used to delete the redundance templates. Blob B^j_{RM} of B_{RM} is obtained.
7: Replace foreground blob B^k_{FG} by $B^i_{\mathrm{OP}} + B^j_{\mathrm{RM}}$.
8: The blobs of the image are extracted.

m [Neubeck 2006] to select largest weight value blob B^i_{OP}. Take Fig.3.9 as an example, denote the moving direction from the left to the right by the value "1", and denote the moving direction from the right to the left by the value "-1". The summation of the directions of all the pixels in the blob is used as the weight of the NMS.

Step4: The fourth step is labeling the remaining region B_{RM}, which is in the blob B_{FG} except the B^i_{OP}. Traverse the remaining region by a preset size rectangle template, with the same size in **Step2**. The blob $B^{j'}_{\mathrm{RM}}$ overlapped by the foreground image is recorded. The non-maximum suppression (NMS) algorithm is used to choose the blob B^j_{RM} from the recorded blob set $\{B^{j'}_{\mathrm{RM}}\}$.

The foreground blob B^k_{FG} is replaced by the optical flow blob B^i_{OP} and the remaining part blob B^j_{RM}. As shown in Fig.3.9, the 1-st rectangle in Fig.3.9(a) is split into 3-rd and 4-th rectangle in Fig.3.9(b).

3.3 Abnormal detection based on histograms of optical flow orientations

In Section 3.1, optical flow has been used to characterize movement information in abnormal detection problems. The optical flow field was arranged in a vector form as an input to the classification algorithm. Although this technique showed good results for some visual scenes, using directly the optical flow does not ensure enough robustness for challenging situations. In this section, we propose histograms of optical flow orientations (HOFO) as a descriptor encoding moving information of each blob and also information about interacting parts in the whole video frame. Furthermore, a fast version of the detection algorithm is designed by fusing the optical flow computation with a background subtraction step.

3.3.1 Related work

Quantized optical flow directions have been used in several works. In [Dalal 2006b, Dalal 2006a], a histogram of optical flow method was used to identify human beings, the derivatives of optical flow, *du* and *dv*, were considered. In [Utasi 2010], a histogram of optical flow directions in region of interest (ROI) was applied to build the model, the magnitude of the optical flow vectors was neglected. While in our work, the two components, *u* and *v* of optical flow, are used to compute the orientation feature of each pixel at a fixed resolution, and then the magnitude of optical flow is considered as the weight to calculate the histogram. In [Adam 2008, Kwak 2011], optical flow was used as the basic feature to characterize behavior. The frame was split into small patches, and a bag-of-words feature was computed to represent the patch. In our work, the histograms of optical flow orientation (HOFO) descriptor is computed over dense grids of overlapping blocks. Further, each block is split into small cells, for example one block is split into 4 cells, and then the histograms of the cells are gathered into a high dimensional vector to represent the whole block. In [Laptev 2008], a histogram of optical flow was computed in the neighborhood of detected points to build a spatio-temporal descriptor. In our work, no feature points are pre-detected.

3.3.2 Histograms of optical flow orientations (HOFO) descriptor

In this subsection, we propose a novel scene descriptor computing the histogram of optical flow orientation (HOFO) of the *original* image, or the *foreground* image obtained after applying background subtraction. The HOFO descriptor is computed over dense and overlapping grids of spatial blocks, with optical flow orientation features extracted at fixed resolution and gathered into a high dimensional feature vector to represent the movement information of the frame. Fig.3.10 illustrates the HOFO feature descriptor of the *original* image and *foreground* image. Each block is divided into cells where HOFO is computed. A weighted vote of each pixel is calculated for the edge orientation histogram channel based on the optical flow element orientation centered on it, then the votes are gathered into orientation bins over local spatial regions. The optical flow magnitude of a pixel is considered as a weight in the voting process.

The calculation procedures of HOFO in *original* frame and *foreground* frame are similar. The HOFO descriptor is calculated at each block, and then accumulated into one global vector denoted as feature F_k for the k-th frame. Fig.3.11 shows the computation of HOFO, it is a feature vector in $n^{\text{blocks}} \times n^{\text{bins}}$ dimension. Horizontal and vertical optical flow (u and v fields) are distributed into 9 orientation bins, over a horizon 0°-360°. The HOFO is computed with an overlapping proportion set as 50% of two contiguous blocks. A block contains $b^h \times b^w$ cells of $c^h \times c^w$ pixels, where b^h and b^w are the number of cells in y and x direction in cartesian coordinates respectively, c^h is the height of the cell, and c^w is the width of the cell. Analyzing jointly local HOFO blocks permits us to consider the behavior in the global frame. Put another way, concatenation of HOFO cells allows us to model the interaction between the motions of the local blocks.

Fig.3.12 illustrates HOFO descriptor of the blobs. Each blob is taken as one frame, and the HOFO computation processes are the same as the ones in Fig.3.11. In SVM abnormal detection algorithm, all the blobs in normal frame are taken as training samples or normal testing samples, while the blobs in the abnormal frame are considered as abnormal samples.

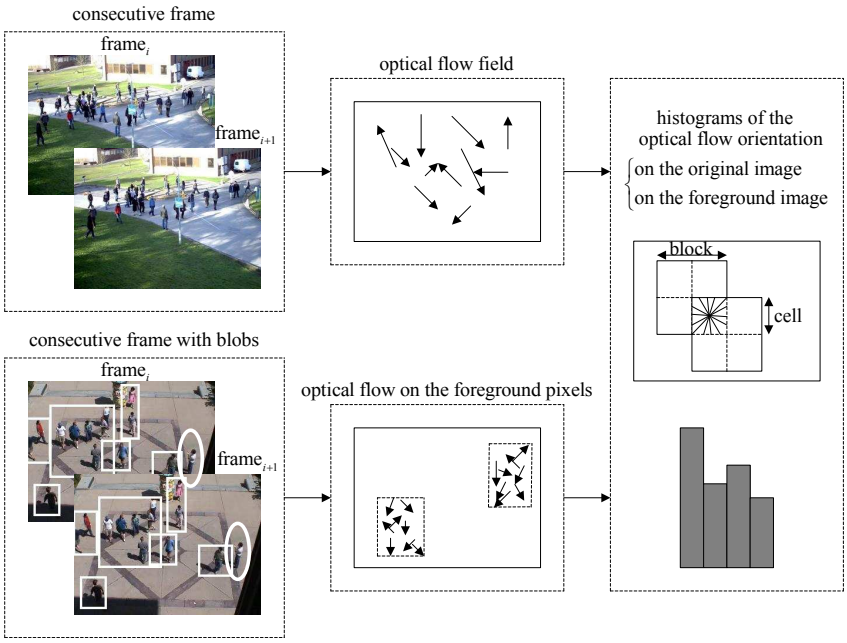


Figure 3.10: Histograms of optical flow orientations (HOFO) of the *original frame*, and of the *foreground frame* obtained after applying background subtraction.

3.3.3 Abnormal detection method

For a given scene in video streams, suppose that a set of training blobs or training frames describing the normal behavior is available. The abnormal behavior is defined as the event deviating from the training set behavior. In this subsection, the abnormal events detection consist of three parts. Firstly, the abnormal blob event detection based on HOFO is proposed. Secondly, the abnormal global frame event detection is introduced. Thirdly, a

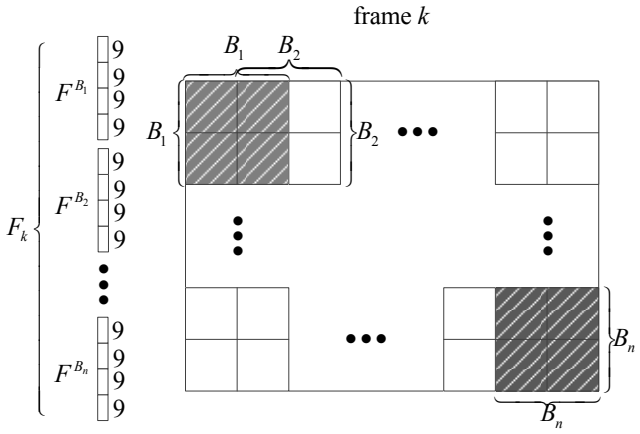


Figure 3.11: Histograms of optical flow orientation (HOFO) computation of the *k*-th *frame*.

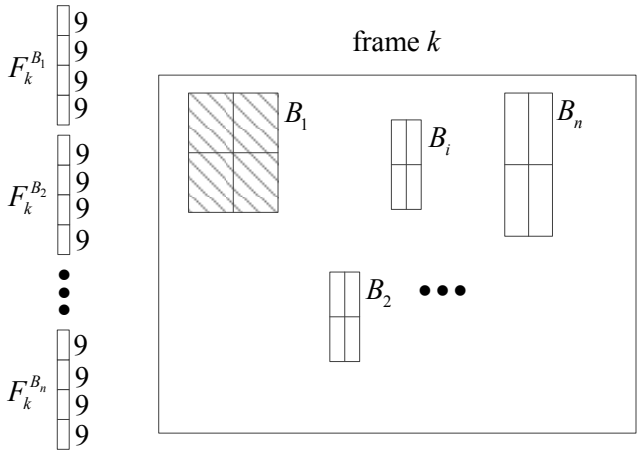


Figure 3.12: Histograms of optical flow orientations (HOFO) computation of the *blob* in the k^{th} *frame*.

fast implementation of the HOFO descriptor will be given later.

3.3.3.1 Abnormal blob events detection method

Assume that a set of blobs $\{B_i^{m_i'}\}$ of the image set $\{I_1^{n^{\text{trn}}+n^{\text{tst}}}\}$, $1 \leq i \leq (n^{\text{trn}}+n^{\text{tst}})$, $1 \leq m_i' \leq m_i$ describing the training (normal) and testing (normal and abnormal) blob behavior of the given scene is available, n^{trn} is the number of the training frames, n^{tst} is the number of the testing frames, m_i is the number of the blobs in the *i*-th frame, m_i' is the index of a blob, $B_i^{m_i'}$ is the m'-th blob in the *i*-th frame. The abnormal blob behavior is defined as an event which deviates from the training set of the blob events. The general architecture of the abnormal *blob* event detection via one-class SVM is explained below.

Step 1: The first step consists of computing the optical flow features at a gray scale image. The blobs are extracted via the method introduced in Section 3.2.

$$\{I_1, I_2, \ldots, I_{n^{\text{trn}}+n^{\text{tst}}}\} \tag{3.6}$$
$$\longrightarrow \{(FG_1, OP_1), \ldots, (FG_{n^{\text{trn}}+n^{\text{tst}}}, OP_{n^{\text{trn}}+n^{\text{tst}}})\} \tag{3.7}$$
$$\longrightarrow \{(B_1^1, \ldots, B_1^{m_1}), \ldots, (B_{n^{\text{trn}}+n^{\text{tst}}}^1, \ldots, B_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}})\} \tag{3.8}$$
$$\longrightarrow \{(OP_1^1, \ldots, OP_1^{m_1}), (OP_2^1, \ldots, OP_2^{m_2}), \ldots, (OP_{n^{\text{trn}}+n^{\text{tst}}}^1, \ldots, OP_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}})\}, \tag{3.9}$$

where I_i is the i-th frame, (FG_i, OP_i) are the foreground image and optical flow of the i-th frame, $\{B_i^1, B_i^2, \ldots, B_i^{m_i}\}$ are the 1-st to m-th blobs in the i-th frame, m_i is the number of the blobs in the i-th frame, $\{OP_i^1, \ldots, OP_i^{m_i}\}$ are the corresponding optical flow of the blobs.

Step 2: The second step is calculating the covariance matrix feature of the blobs. Fig.3.12 illustrates the details of this step.

$$\begin{aligned}&\{(OP_1^1, B_1^1, \ldots, OP_1^{m_1}, B_1^{m_1}), \ldots, (OP_{n^{\text{trn}}+n^{\text{tst}}}^1, B_{n^{\text{trn}}+n^{\text{tst}}}^1, \ldots, OP_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}}, B_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}})\}\\ &\longrightarrow \{(HOFO_1^1, \ldots, HOFO_1^{m_1}), \ldots, (HOFO_{n^{\text{trn}}+n^{\text{tst}}}^1, \ldots, HOFO_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}})\},\end{aligned} \tag{3.10}$$

where $\{HOFO_i^1, \ldots, HOFO_i^{m_i}\}$ are the corresponding HOFO descriptor of the blobs in the i-th frame.

Step 3: The third step is applying one-class SVM on the extracted descriptors of the training normal blobs to obtain the support vectors.

$$\begin{aligned}&\{(HOFO_1^1 \ldots HOFO_1^{m_1}), \ldots, (HOFO_{n^{\text{trn}}}^1 \ldots HOFO_{n^{\text{trn}}}^{m_{n^{\text{trn}}}})\}\\ &\xrightarrow{SVM} \text{support vector } \{Sp_1, Sp_2, \ldots, Sp_o\},\end{aligned} \tag{3.11}$$

where $\{(HOFO_1^1 \ldots HOFO_1^{m_1}), \ldots, (HOFO_{n^{\text{trn}}}^1 \ldots HOFO_{n^{\text{trn}}}^{m_{n^{\text{trn}}}})\}$ are the covariance matrix descriptors of the training blobs. The number of blobs in the i^{th} frame is m_i, the total number of training samples is $m_{N^{\text{trn}}} = m_1+m_2+...+m_{n^{\text{trn}}}$. A subset $[Sp_1, Sp_2, \ldots, Sp_o], o \ll m_N$ are the support vectors contributing to the decision function.

Step 4: Based on the support vectors obtained from the training blobs, an incoming blob sample $HOFO_l^{m_l'}$ is classified. The flowchart of the abnormal detection method is shown in Fig.3.13, and described as the following equation:

$$f(HOFO_l^{m_l'}) = \mathbf{sgn}(\sum_{i=1}^{o} \alpha_i \kappa(Sp_i, HOFO_l^{m_l'}) - \rho) \tag{3.12}$$

$$= \begin{cases} 1 & \text{if } f(HOFO_l^{m_l'}) \geq 0 \\ -1 & \text{if } f(HOFO_l^{m_l'}) < 0, \end{cases} \tag{3.13}$$

where $HOFO_l^{m_l'}$ is the HOFO descriptor of the m_l'-th blob in the l-th frame needed to be classified. Sp_i is the support vector. "1" corresponds to the normal blob, "-1" corresponds to the abnormal blob.

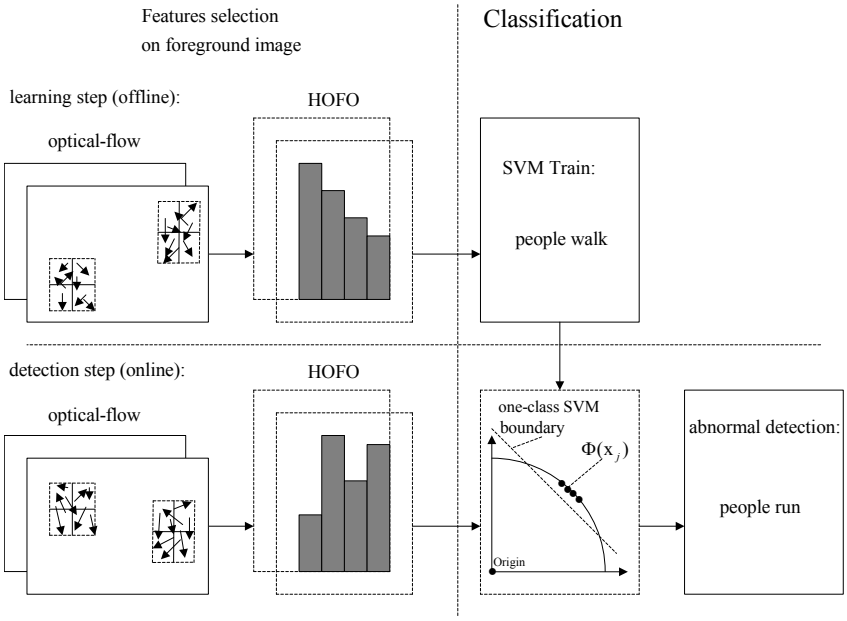


Figure 3.13: Major processing states of the proposed one-class SVM abnormal *blob* event detection method. HOFO of the *blob* is calculated.

The abnormal *blob* detection and localization conceptions are defined by depending on the implementation. Firstly, if the blobs of moving objects are provided, the abnormal action of the objects can be detected. Alternatively, the position of the object yielding an abnormal behavior in crowded scenes can be localized. The target that triggers the abnormal event is labeled automatically without human intervention, thus the target can be tracked.

3.3.3.2 Abnormal frame events detection method

The blob abnormal detection method can be adjusted to global frame visual abnormal event detection by taking the whole frame as one blob. The processes of feature descriptor computation and one-class SVM classification are similar as ones introduced in Section 3.3.3.3, but the descriptor changes from *blob* HOFO to *frame* HOFO. Moreover, for abnormal *frame* events detection, the precondition of one event could be defined as normal or abnormal is that it occurs during several consecutive frames. In other words, the normal or abnormal event is not punctual. Based on this premise, the short *abnormal* event clip which occurs intermittently at few frames in the long normal video sequence could be modified to *normal* state. Likewise, the *normal* event frames which are detected among the long consecutive sequence of abnormal frames could be altered to *abnormal*. A threshold N of the number of image frames is preset, the post-processing of the detection results is illustrated in Fig.3.14. If the number of abnormal states (negative predicted results) exceed the threshold N within normal states (positive predicted results), then the *normal* prediction labels are converted into *abnormal*. The performance of this state transition model is analyzed in Chapter 3. The abnormal *frame* detection results in Chapter 3 and Chapter 3 are obtained by SVM

classification method without applying the state transition model.

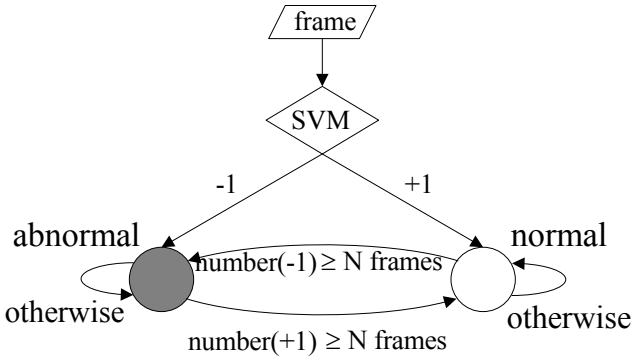


Figure 3.14: State transition model. *N* is the preset threshold number to adjust the detection result.

3.3.3.3 Abnormal frame events detection method based on foreground image

In case of a stationary camera, the foreground segregation becomes feasible by using change detection algorithm. In the following, we propose a fast implementation of the abnormal detection algorithm based on the foreground pixels.

Step 1: The first step consists of calculating the optical flow feature of the foreground image. The training frames are processed via optical flow method. And then the optical flow on the foreground is extracted. This procedure can be described as:

$$\{I_1, I_2, \ldots, I_n\} \longrightarrow \{OP_1^{\mathrm{FG}}, OP_2^{\mathrm{FG}}, \ldots, OP_n^{\mathrm{FG}}\}, \tag{3.14}$$

where $\{I_1, I_2, \ldots, I_n\}$ are the training normal frames, $\{OP_1^{\mathrm{FG}}, OP_2^{\mathrm{FG}}, \ldots, OP_n^{\mathrm{FG}}\}$ are the corresponding optical flows of the training foreground frames.

Step 2: The second step is calculating the HOFO of training foreground frames. The sketch map of choosing the features of the foreground image is shown in Fig.3.15. HOFO is computed on the global foreground image, the background area is not considered when the HOFO is being calculated. The proportion of consuming time between computing the HOFO of foreground patches and computing the HOFO of whole image is $\frac{A_{\mathrm{FG}}}{A_{img}}$, where A_{FG} is the area of the foreground, A_{img} is the area of the whole image. The foreground area can be regarded as the pixel number of the foreground. The step can be described as the following expression:

$$\{OP_1^{\mathrm{FG}}, OP_2^{\mathrm{FG}}, \ldots, OP_n^{\mathrm{FG}}\} \xrightarrow{HOFO} \{HOFO_1^{\mathrm{FG}}, HOFO_2^{\mathrm{FG}}, \ldots, HOFO_n^{\mathrm{FG}}\}, \tag{3.15}$$

where $\{OP_1^{\mathrm{FG}}, OP_2^{\mathrm{FG}}, \ldots, OP_n^{\mathrm{FG}}\}$ are optical flows of the training foreground frames, $\{HOFO_1^{\mathrm{FG}}, HOFO_2^{\mathrm{FG}}, \ldots, H$ are the HOFO descriptors of the training foreground frames.

The following classification steps are the same as the steps proposed previously in section , but the features of the frame change from *blob* HOFO descriptor to the *foreground frame* HOFO descriptor.

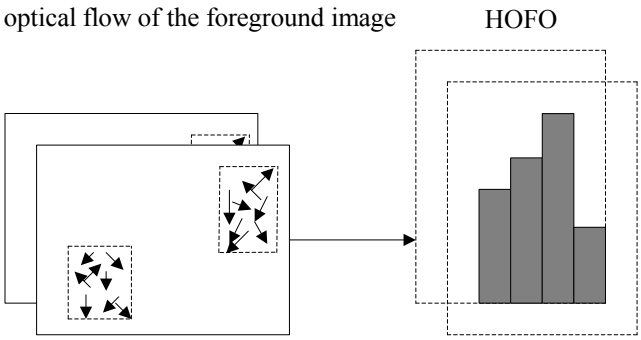


Figure 3.15: Feature selection. Compute the HOFO on the *foreground* images.

3.3.4 Experimental results

This section presents the results of experiments conducted to analyze the performance of the proposed HOFO descriptor and one-class SVM based method for abnormal blob event detection and abnormal frame event detection results.

3.3.4.1 Experimental results of abnormal blob events detection

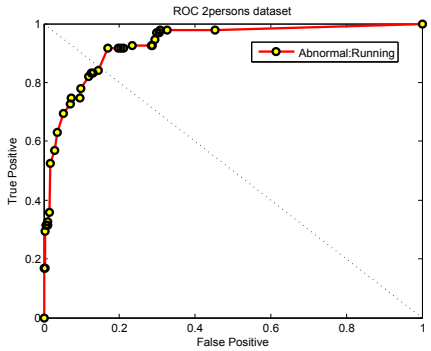
This section presents the results of abnormal blob events detection. The detection results of a scene with pedestrian movement parallel to the camera plane are show in Fig.3.16. The individual is walking or running in the scene. It simulates the abruptly changing velocity abnormal events scenes. The sequence is of low resolution, the people have a height about 30 pixels. The moving people are detected by background subtraction method. The samples for training and the normal samples for testing are obtained from the blobs that people are walking. The abnormal samples which correspond to the blob events needed to be detected are *blob* HOFO where people are running. Our method can distinguish the abnormal running blobs from the walking blobs. In receiver-operating characteristic (ROC) curve [Hanley 1982, Bradley 1997, Metz 1978], the true positive rate means that the running blob is classified as abnormal, while the false positive rate means that the walking blob is detected as abnormal. The detection accuracy of running people is 89.8%, the AUC is 0.9318.

The detection results of lawn scene and plaza scene of UMN dataset are shown in Fig.3.17. The objective of abnormal blob detection is to find all the abnormal blobs. The normal samples are the scenes where the persons are walking toward all the directions, these frames are chosen as training samples and normal testing samples, the abnormal scenes are where persons are running, these frames are chosen as abnormal testing frames. If the abnormal blob events are considered, the training samples are the HOFO of all the walking blobs, the abnormal testing samples are the HOFO descriptors of the running blobs. The results show that the abnormal detection algorithm of blob HOFO descriptor can obtain satisfactory detection results. The ROC of abnormal frame detection results based *original frame* HOFO (will be presented in Section 3.3.4.2) are also shown in the figures for comparing. In abnormal frame detection problem, the true positive means to classify

(a) Normal scenes for training

(b) Detect one person running

(c) Detect two persons walking

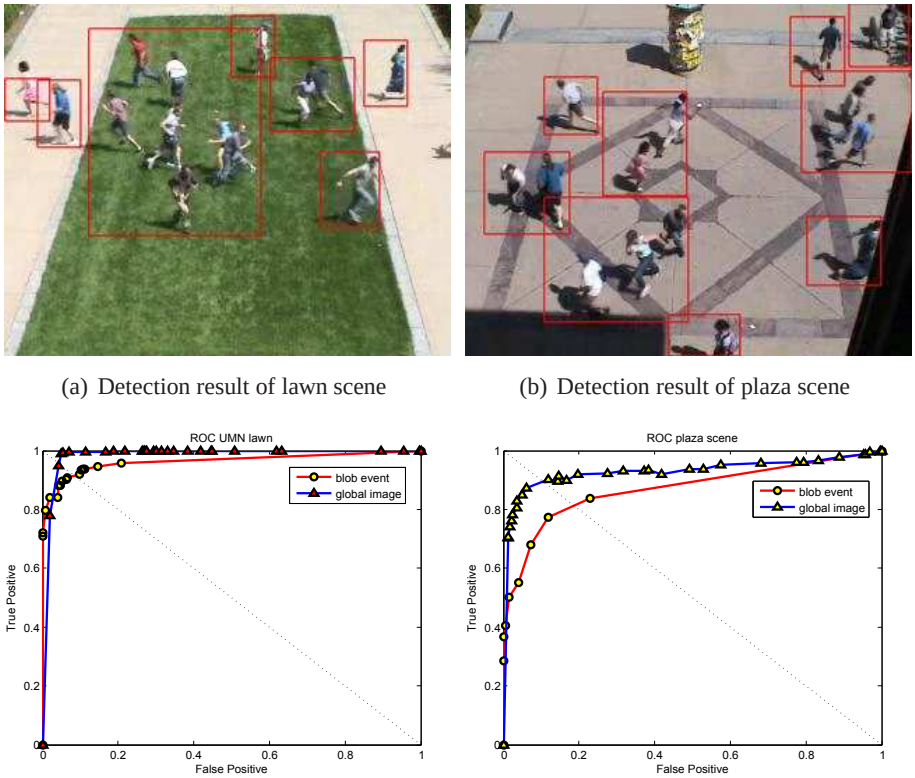


(d) ROC of 2persons dataset

Figure 3.16: Abnormal *blob* event detection results of two persons walking or running scene based on *blob* HOFO descriptor via one-class SVM. (a) The normal scenes for training, two persons are walking. (b) The detection result of one person is running. The red rectangle labels the abnormal blob, the person is running. The blue rectangle labels the normal blob, the person is walking. (c) The detection result of two persons are walking. (d) ROC curve of two persons walking and running dataset. The AUC is 0.9318

the frame where most of the persons are running as abnormal. In fact the blob detection method cannot label all the persons exactly by rectangle, sometimes the rectangle is on the background, or does not include all the parts of the human. These are the major reasons of lower value AUC of the blob based method. Nevertheless, the abnormal *blob* detection can obtain similar performance as the abnormal global *frame* detection by presetting a threshold of the percentage of blobs in one frame. For example, if 80% of the blobs on one frame are classified as abnormal, this frame is considered as abnormal frame. In the indoor scene of UMN dataset, the persons are almost conflicted each other and moving toward the same direction with similar velocities, the blob extraction cannot distinguish each person separately. Thus, our blob extraction based abnormal detection method is not applied to the indoor scene.

The detection results of local mall scenes in which people are running are shown in Fig.3.18. The abnormal blobs representing unusual speed are detected. The AUC of ab-



(a) Detection result of lawn scene

(b) Detection result of plaza scene

(c) ROC curve of lawn scene

(d) ROC curve of plaza scene

Figure 3.17: Abnormal *blob* event detection results of UMN dataset based on *blob* HOFO descriptor via one-class SVM. (a) Abnormal Detection results of scene lawn. The red rectangles label the abnormal running blobs. (b) Abnormal detection results of one scene plaza, all the persons are running. The red rectangles label the abnormal running blobs. (c) ROC curve of abnormal blob detection and abnormal frame detection in scene lawn. The AUC of blob detection is 0.9642. The AUC of frame detection is 0.9845. (d) ROC curve of abnormal blob detection and abnormal frame detection in scene plaza. The AUC of blob detection is 0.8698. The AUC of frame detection is 0.9284.

normal blobs detection results is 0.8868.

3.3.4.2 Experimental results of abnormal frame events detection and foreground frame events detection

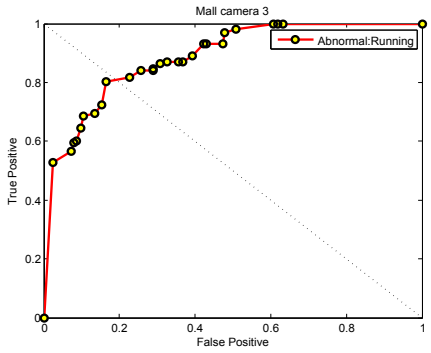
This subsection presents the results of experiments conducted to analyze the performance of the proposed method. UMN [UMN 2006] and PETS2009 [PETS 2009] datasets are adopted in our abnormal frame events detection experiments.

3.3.4.2.1 UMN dataset UMN dataset contains eleven video sequences of three different scenes (lawn, indoor and plaza) of crowded escape events. The detection results of the lawn scene and the plaza scene are shown in Fig.3.19 and Fig.3.20. The normal scene

(a) Normal scenes for training

(b) Detect one person running

(c) Detect one person running



(d) ROC mall scene

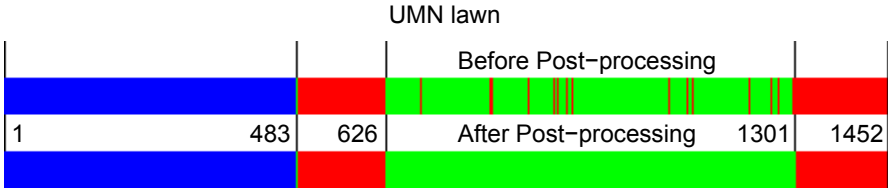
Figure 3.18: Abnormal *blob* event detection results of the mall scene based on *blob* HOFO descriptor via one-class SVM. (a) The normal scenes for training, two persons are walking. (b) The detection result of one person is running. The red rectangle labels the abnormal blob, the person is running. The blue rectangle labels the normal blob, the person is walking. (c) The detection result of one person is running. (d) ROC curve of mall scene. The AUC is 0.8868

is defined as individuals walking in different directions, the training samples and normal testing samples are selected from these frames. The abnormal scene is where the individuals are running, the abnormal testing samples are extracted from these frames. The results show that the abnormal detection algorithm of both the *original image* HOFO descriptor and the *foreground image* HOFO descriptor can obtain satisfactory detection performances. However, taking the HOFO of the *foreground image* as a feature saves the program running time.

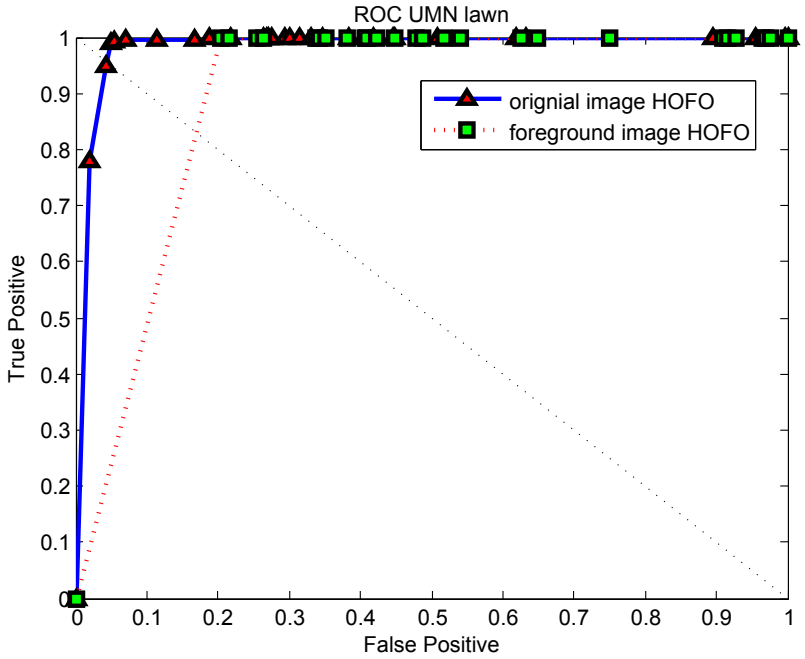
The detection results of the indoor scene are shown in Fig.3.21. The lower AUC value of the indoor scene is mainly due to the time lags of the frame labels. There are no people in the last few frames labeled as abnormal of each abnormal sequence. Whereas in the the training frames, there is no person in the upper half of the image. Because the HOFO descriptor shows the global moving information of the frame, the HOFO of training frame is similar to the HOFO of the abnormal frame without people. Our HOFO feature descriptor based classification method cannot distinguish this situation. However, this problem can

(a) Normal lawn scene

(b) Abnormal lawn scene



(c) Lawn scene results

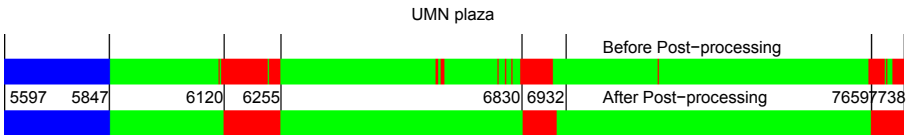


(d) ROC curve of lawn scene

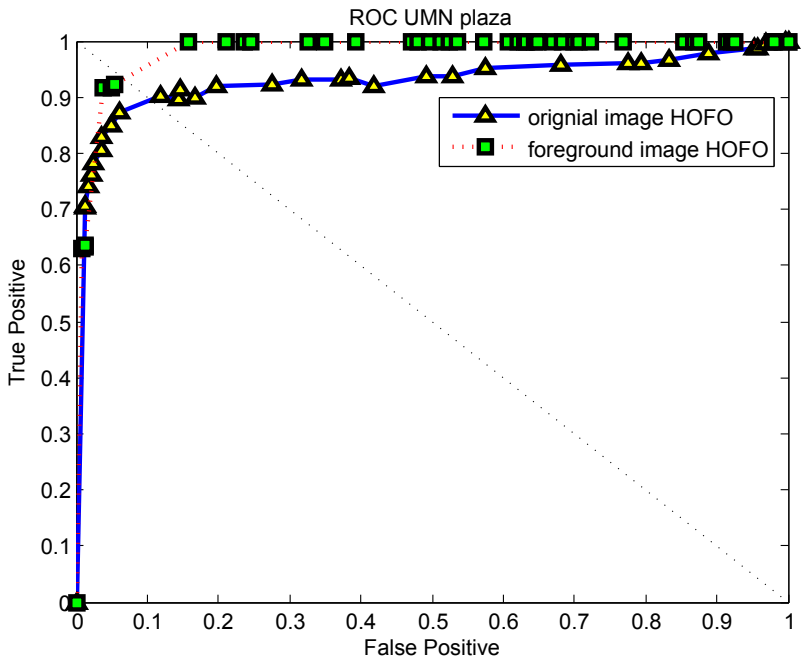
Figure 3.19: Abnormal *frame* event detection results of the lawn scene based on *original frame* HOFO and *foreground frame* HOFO via one-class SVM. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) The detection result bar represents the label of each frame based on the original frame HOFO. The upper bar shows the detection results before post-processing. The lower bar shows the results after applying state transition model. *Blue*, *green* and *red* color represents the training frames, normal frames, and abnormal frames respectively. Several pivotal frames are marked. (d) ROC curve of lawn scene results before applying the state transition model. The AUC of the *original frame* HOFO result is 0.9845. The AUC of the *foreground frame* HOFO result is 0.8975.

(a) Normal plaza scene

(b) Abnormal plaza scene



(c) Plaza scene result



(d) ROC curve of plaza scene

Figure 3.20: Abnormal *frame* event detection results of the plaza scene based on *original frame* HOFO and *foreground frame* HOFO via one-class SVM. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) The detection result bar represents the labels of each frame of the dataset based on the original frame HOFO. The upper bar shows the detection results before post-processing. The lower bar shows the results after applying state transition model. (d) ROC curve of plaza scene results before applying the state transition model. The AUC of the *original frame* HOFO result is 0.9284. The AUC of the *foreground frame* HOFO result is 0.9815.

be resolved by utilizing the foreground information. For example, if there are no moving objects in the frame, this frame is immediately classified as abnormal. In this paper, all the performance data are obtained from the results based on the HOFO feature descriptor classify algorithm.

The performances of our HOFO based method and of the state-of-the-art methods are shown in TABLE 3.2. The AUC value of our proposed method in the table is calculated from the detection results before applying the state transition model. The states of the frames, where the event is changing from normal to abnormal, are inherently ambiguous. These frames can be either be labeled as normal or abnormal. If the detection results of these ambiguous frames (about 15 frames, 1 second in surveillance video) are not considered, the AUC of our abnormal detection results after applying state transition model can approach 1.

Method	Area under ROC		
	lawn	indoor	plaza
Social Force [Mehran 2009]	0.96		
Optical Flow [Mehran 2009]	0.84		
NN [Cong 2011]	0.93		
SRC [Cong 2011]	0.995	0.975	0.964
STCOG [Shi 2010]	0.9362	0.7759	0.9661
HOFO (Ours)	0.9845	0.9037	0.9815

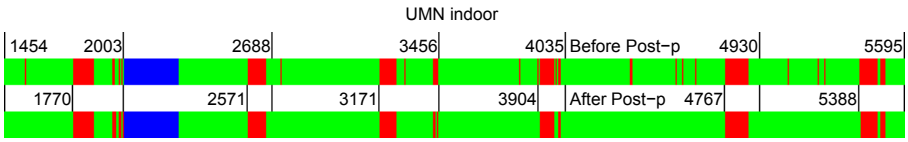
Table 3.2: The comparison of our proposed HOFO descriptor and one-class SVM based method with the state-of-the-art methods for abnormal *frame* event detection of UMN dataset. The AUC values of our HOFO descriptor based classified method are calculated from the detection results before applying the state transition model. The AUC can approach 1 if a state transition model is applied.

3.3.4.2.2 PETS dataset Because taking an HOFO of *foreground image* and *original image* as a feature descriptor has similar abnormal detection results, we only show the results based on the *original image* HOFO of the PETS2009 dataset [PETS 2009]. The detection results of the PETS scene (the sequence labeled as *Time14-17*) are shown in Fig.3.22. The training samples and the normal testing samples are extracted from the sequence (*Time14-55*) where the individuals are walking in different directions. The abnormal testing samples are the frames where the people are moving (walking or running) in one direction. The abnormal detection results before and after applying the state transition model are exhibited in Fig.3.23. The accuracy of abnormal detection results before state transition post-processing is 90.00%. By applying the state transition constraint, the detection results fluctuate less.

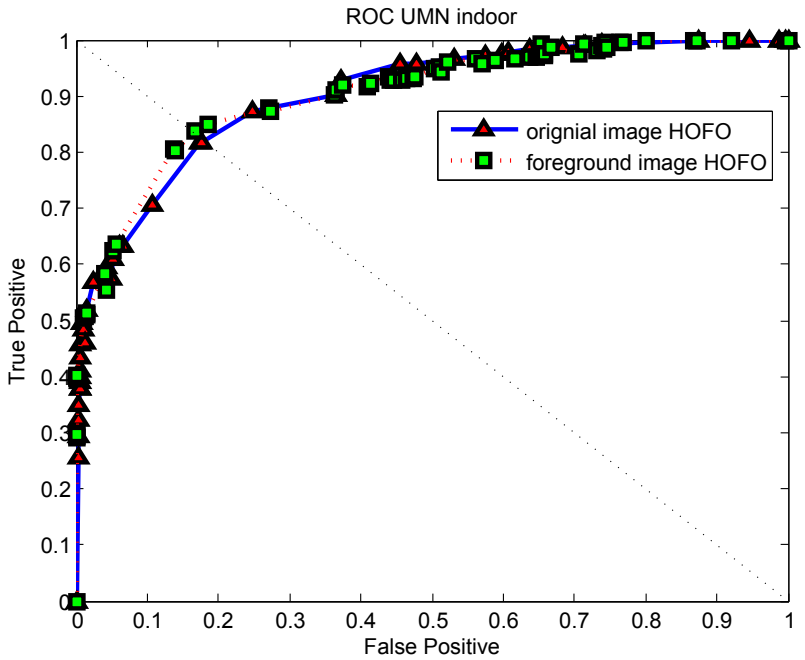
Fig.3.24 shows the detection results of sequence *Time14-16*, where individuals are walking or running in the same direction. A normal state corresponds to the frames where the individuals are walking, while an abnormal state corresponds to the frames where the

(a) Normal indoor scene

(b) Abnormal indoor scene

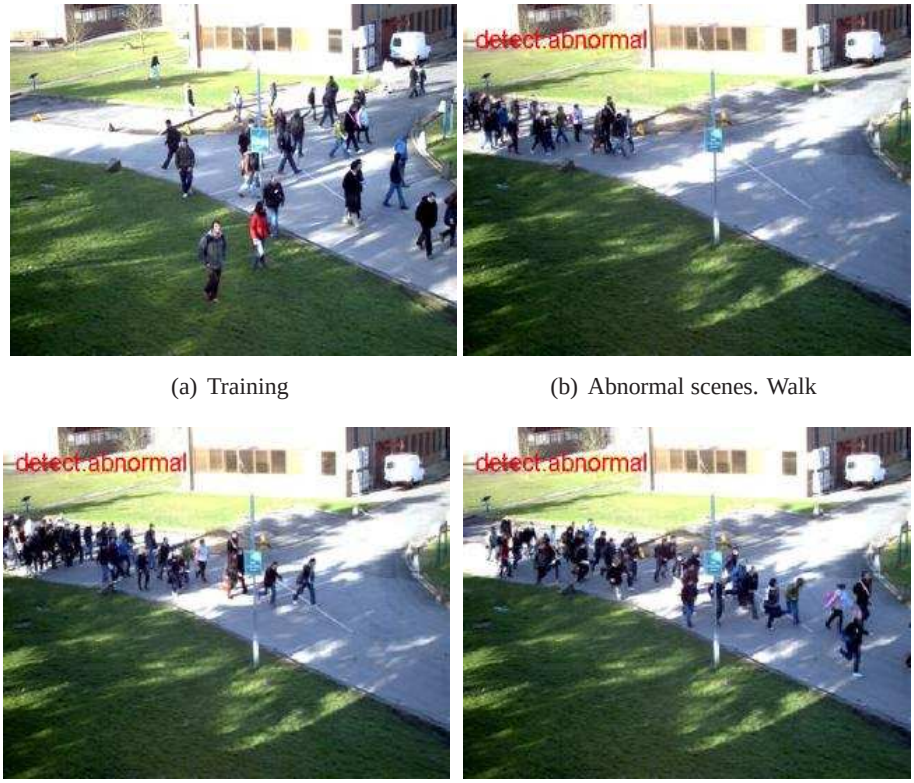


(c) Indoor scene result



(d) ROC curve of indoor scene

Figure 3.21: Abnormal *frame* event detection results of the indoor scene based on *original frame* HOFO descriptor and *foreground frame* HOFO descriptor via one-class SVM. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) The detection result bar represents the labels of each frame based on the original frame HOFO. The upper bar shows the detection results before post-processing. The lower bar shows the results after applying state transition model. *Blue*, *green* and *red* color represents the training frames, normal frames, and abnormal frames respectively. (d) ROC curve of indoor scene results before applying the state transition model. The AUC of the *original frame* HOFO is 0.9022. The AUC of the *foreground frame* HOFO is 0.9037.



(a) Training (b) Abnormal scenes. Walk

(c) Abnormal scenes. Move (d) Abnormal scenes. Run

Figure 3.22: Abnormal *frame* event detection results of *Time14-17* based on *original frame* HOFO descriptor via one-class SVM. (a) Training frames, individuals are walking toward different directions. (b) Abnormal frames, individuals are walking toward the identical direction. (c) Abnormal frames, individuals are moving (walking or running) toward the identical direction. (d) Abnormal frames, individuals are running toward the identical direction.

people are running. The training samples are chosen from the frames (*Time14-17*, *Time14-31*) where people are walking in the same direction. The detection results are illustrated in Fig.3.25. The accuracy of the results before applying state transition post-processing is 93.24%. False alarms are reduced by applying the state transition model.

The crowd splitting sequence (*Time14-31*) detection results are shown in Fig.3.26. Frames where there is one cohesive crowd are considered as normal, while frames where the crowd is splitting are considered as abnormal. Training samples are extracted from the frames (*Time14-16*) where people are walking in the same direction. Fig.3.26(c) shows the detection results of each frame. The accuracy of the results before state transition post-processing is 94.62%. The state transition model leads to a 13 frame delay of predicting an abnormal event, but the fluctuations between the *abnormal* and the *normal* state are reduced.

The crowd formation and evacuation sequence (*Time14-33*) detection results are presented in Fig.3.27. Crowd formation is defined by the scene in which the people are walk-

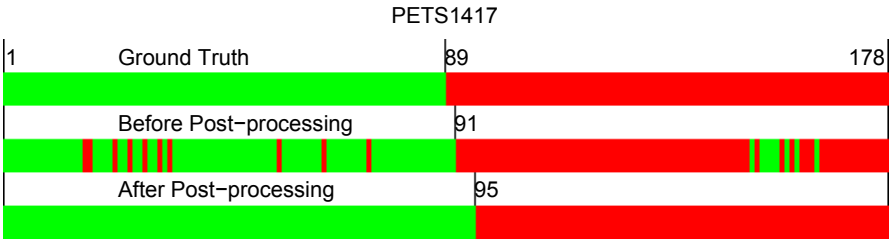
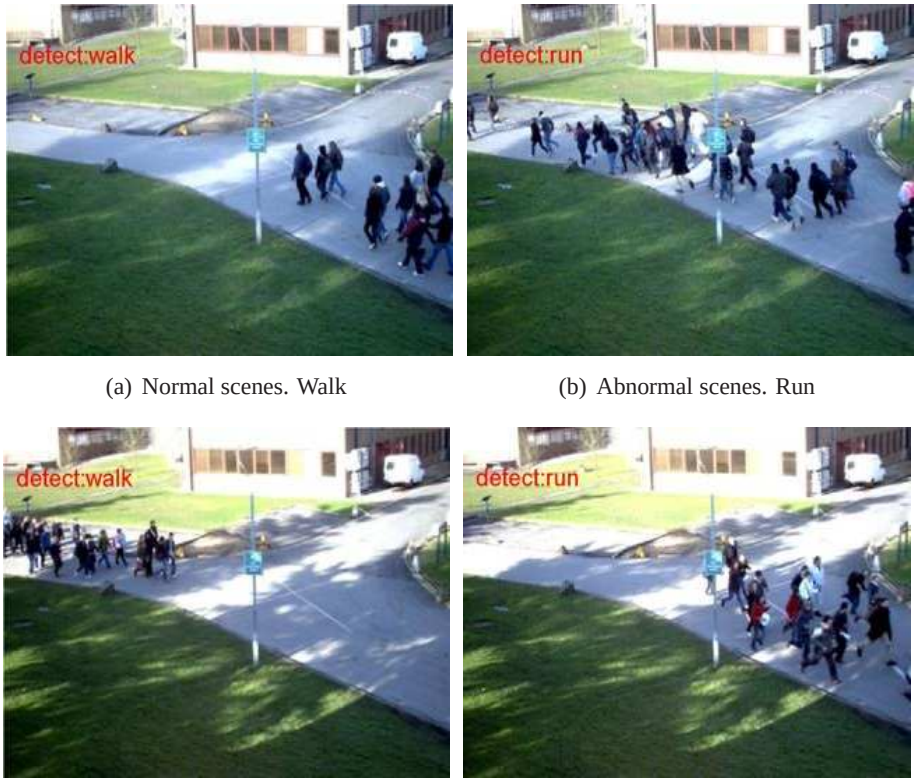


Figure 3.23: *Time14-17* results based on *original frame* HOFO descriptor via one-class SVM. *Green* color represents the normal frames, and *red* color corresponds with abnormal frames. 400 training frames (Frame 0^{th} to 399^{th}), and 89 normal testing frames (Frame 400^{th} to 488^{th}) are obtained from *Time14-55*. 89 abnormal testing frames (Frame 0^{th} to 89^{th}) are selected from *Time14-17*. The accuracy of detection results before state transition post-processing is 90.00%.

ing towards the convergence point. Evacuation refers to the scene in which the people are diverging. The essence of abnormal detection is to find the sample which differs from the training data, hence the outputs are two states, normal and abnormal. The frames where the people are loitering in small areas around a location are considered as normal, as shown in Fig.3.27(c). The other two situations including crowd formation and individual evacuation are considered as abnormal. The training frames are chosen from sequence (*Time14-55*) where people are walking in different directions. Because the orders of events are obtained in advance, the abnormal states before the normal events are classified as "gathering (crowd formation)", while the other abnormal events are labeled as "evacuation". If the abnormal detection mission, distinguishing running event from walking is taken into account: such as the example of sequence *Time14-16* shown in Fig.3.24, the two events, "gathering" and "evacuation", can be discriminated without the prior information of event order. Each frame is split into four parts *A*,*B*,*C* and *D*, as illustrated in Fig.3.27(a). The HOFO feature descriptor is calculated in each sub-image, respectively. Based on this image segmentation, the global frame abnormal detection task is decomposed into sub-frame events analysis. However, part *D* is not considered, for there are no people in this sub-image in the crowd formation period. Fig.3.28 presents the detection results of each frame. The individuals gather at the convergence point at different times, the earlier "gathering" events occurs in sub-frame *C*. The individuals assemble in sub-frame *C*, *B* and *A* at frames 73, 111 and 175, respectively. The rapid dispersion event occurs at almost the same time in these three sub-images, close to frame 341. The global frame detection accuracy of the results after state transition post-processing is 97.88%.

The local dispersion sequence (*Time14-27*) detection results are shown in Fig.3.29. As shown in Fig.3.29(a), each frame is split into five parts *A*,*B*,*C*,*D* and *E*, the cross-point is the convergence place of the individuals. Owning to the occlusion in part *A*, people loitering obscure the people dispersing, a precise part *E* is segmented out of *A*. Local dispersion is defined by the scene in which people in each part are walking in one direction, the opposite direction from the convergence point. Local dispersion is considered as an abnormal event, loitering is considered as a normal event. Training samples are chosen from the sequence



(a) Normal scenes. Walk (b) Abnormal scenes. Run

(c) Normal scenes. Walk (d) Normal scenes. Run

Figure 3.24: Abnormal *frame* event detection results of *Time14-16* based on *original frame* HOFO descriptor via one-class SVM. (a) Pedestrians are walking toward the identical direction, from right to left. (b) Pedestrians are running toward the identical direction, from right to left. (c) Pedestrians are walking toward the identical direction, from left to right. (d) Pedestrians are running toward the identical direction, from left to right.

(*Time14-55*) where people are walking in different directions. The detection result of each frame is shown in Fig.3.30. In sub-image *E*, the frames 92 to 106, 120 to 130, and 273 to 294 are classified as abnormal states, which are defined as local dispersion. Frames 107 to 119 in the sub-frame *E*, the optical flows of the moving are not detected for the occlusion. These frames are detected as normal states. In part *B*, the local dispersion is not easy to detect, as few individuals in this part are moving. The accuracy of the global frame detection results after state transition post-processing is 88.89%.

The experimental results on the sequences show that our proposed method can successfully discriminate panic-driven events and irregular moving queues. Our feature is based on the optical flow obtained by the HS method, whereas there are other methods that can compute precise optical flow. If a more precise optical flow can be obtained, the more robust abnormal detection results that our HOFO based method can provide than this paper. Nevertheless, based on the optical flow which is calculated by the HS method, our proposed method can give satisfactory abnormal detection results.

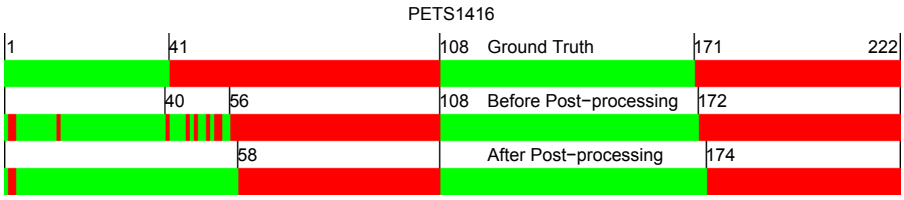
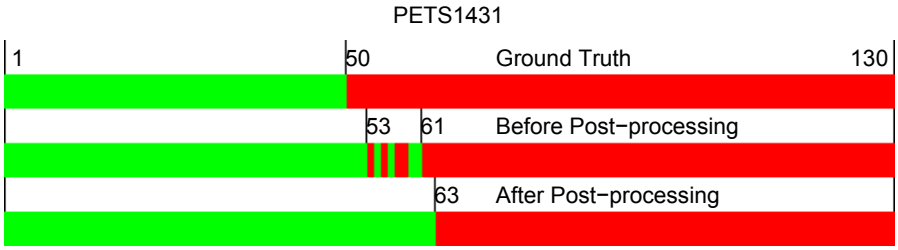


Figure 3.25: *Time14-16* results based on *original frame* HOFO descriptor via one-class SVM. *Green* color represents the normal frames, and *red* color corresponds with abnormal frames. Trains frames are chosen from *Time14-17* and *Time14-31*: 61 frames (Frame 0^{th} to 60^{th}) in *Time14-17* where pedestrians are walking from left to right, 50 frames (Frame 0^{th} to 49^{th}) in *Time14-31* where pedestrians are walking from right to left. The accuracy of detection results before state transition post-processing is 93.24%.

3.4 Conclusion

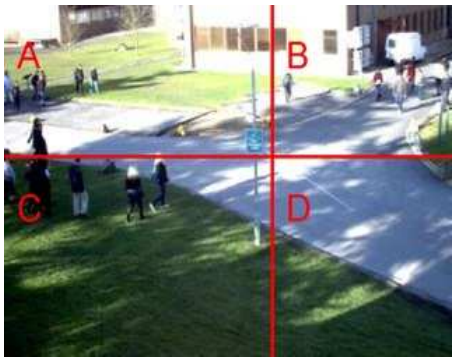
In this chapter, the abnormal frame detection based on block feature of optical flow is proposed. For analyzing the activity of the single person, the blob extraction method based on the foreground and the optical flow in a crowded scene is proposed. Also, an other descriptor based on the histogram of optical flow orientations (HOFO) is proposed to detect abnormal blobs and abnormal frames. Nonlinear one-class SVM algorithms are then used for classification. A fast implementation based on background subtraction is also proposed. The proposed detection algorithms have been tested on several video datasets yielding successful results in detecting abnormal events.

(a) Normal scenes. Cohesive crowd (b) Abnormal scenes. Crowds split



(c) PETS *Time14-31* results

Figure 3.26: Abnormal *frame* event detection results of *Time14-31* based on *original frame* HOFO descriptor via one-class SVM. (a) Cohesive crowd of persons. (b) Multiple diverging flows. (c) The detection result bar represents the labels of each frame. 41 training frames (Frame 0^{th} to 40^{th}) are obtained from *Time14-16*. The detection accuracy before state transition post-processing is 94.62%.



(a) Frame is split into 4 parts

(b) Abnormal scenes. Gather

(c) Normal scenes. Loiter

(d) Abnormal scenes. Evacuation

Figure 3.27: Abnormal *frame* event detection results of *Time14-33* based on *original* image HOFO descriptor via one-class SVM. (a) The frame is split into 4 parts, *A*,*B*,*C* and *D*. (b) Crowd formation. (c) Individuals are loitering. (d) Evacuation of the persons.

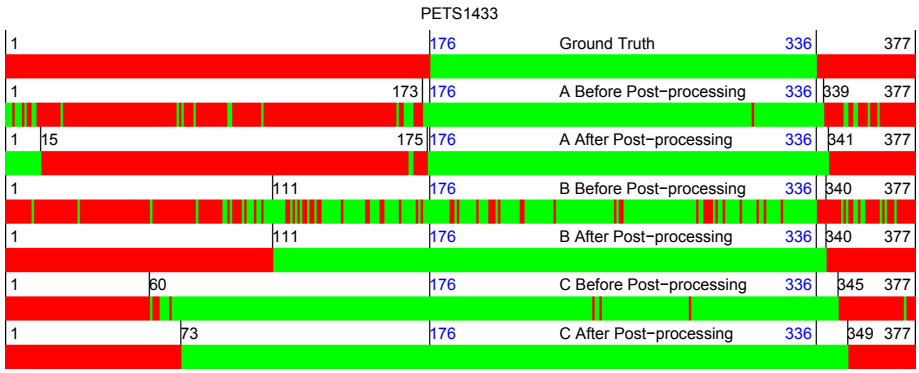
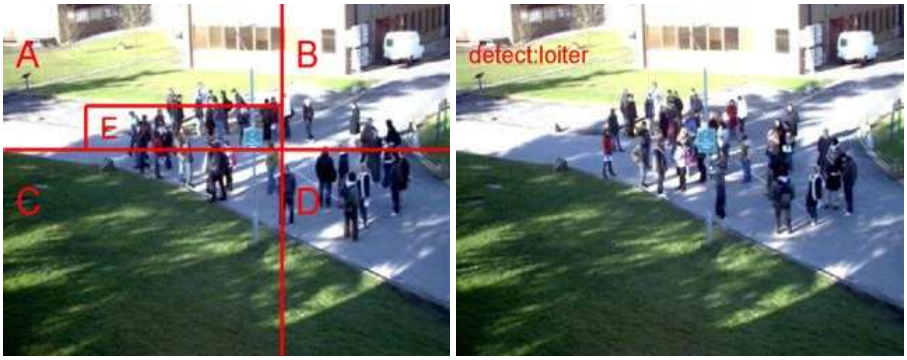


Figure 3.28: *Time14-33* results based on *original* image HOFO descriptor via one-class SVM. 269 training frames (Frame 81^{th} to 349^{th}) are obtained from *Time14-55*. The accuracy before applying state transition model of results, in Part *A* is 90.98%, in Part *B* is 81.96%, in Part *C* is 85.68%. The accuracy after applying state transition model of the global frame is 97.88%.



(a) Frame is split into 5 parts (b) Normal scenes. Loiter

(c) Normal scenes. Loiter (d) Abnormal scenes. Dispersion

Figure 3.29: Abnormal *frame* event detection results of *Time14-27* based on *original* image HOFO descriptor via one-class SVM. (a) The frame is split into 5 parts. (b) Individuals are loitering in small areas. (c) Another frame of individuals are loitering in small areas. (d) Local dispersion of crowds.

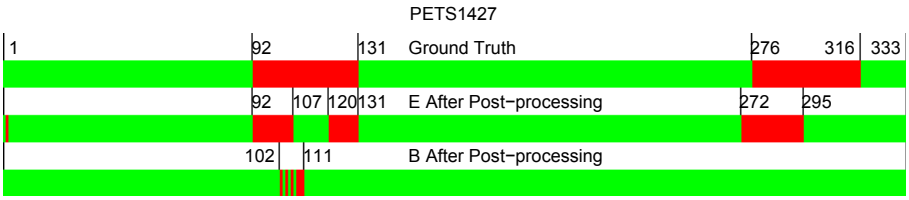


Figure 3.30: *Time14-27* results based on *original* image HOFO descriptor via one-class SVM. 269 training frames (Frame 81^{th} to 349^{th}) are chosen from *Time14-55*. In sub-image *B*, the abnormal state defined as local dispersion is detected at frame 102^{th},104^{th},106^{th},and from 108^{th} to 110^{th}. The accuracy after applying state transition model is 88.89%.

Chapter 4

Abnormal detection based on covariance feature descriptor

Contents

In this chapter, we propose a covariance matrix descriptor fusing both optical flow and intensity information of a blob or a whole image. This proposed descriptor is inspired by region covariance [Tuzel 2006] used for patch matching in a tracking problem and for object detection. One of the advantages of the covariance descriptor is its constant and low dimensionality whatever the number of considered pixels from which low-level features are extracted. As in the previous chapter, we use the one-class support vector machines (OC-SVM), as a model-free pattern recognition method to detect abnormal events. In the nonlinear one-class SVM, a multi-kernel strategy is also proposed to tune the importance of the partial features, in order to enhance improve the abnormal detection performances.

The rest of the chapter is organized as follows. In Section 4.1, the proposed covariance matrix descriptor encoding motion features and intensity features is introduced. In Section 4.2, we propose the multi-kernel strategy, and an overview of our visual-based abnormal blob or frame event detection method. In Section 4.3, we present the abnormal blob localization and abnormal frame detection results on benchmark datasets. Finally, Section 4.4 concludes the chapter.

4.1 Covariance Descriptor

The covariance matrix is proposed by O. Tuzel [Tuzel 2006] for describing gray or color blob image features. It has been successfully used in the object detection problem

[Tuzel 2007, Tuzel 2008], the face recognition problem [Pang 2008], and the tracking problem [Porikli 2006c]. The covariance descriptor is robust against noise, illumination distortions, and rotation [Porikli 2006a]. A fast construction of the covariance matrix is introduced in [Porikli 2006b]. The performance of different features constructing the covariance matrix descriptor has been analyzing in [Cortez-Cargill 2009]. We propose to construct covariance matrix descriptor based on the optical flow and the intensity to encode movement features both in a blob and in a global image. The covariance descriptor is defined as:

$$F(x, y, \ell) = \phi_\ell(I, x, y) \tag{4.1}$$

where I is an image (which can be gray, red-green-blue (RGB), etc.), F is a $W \times H \times d$ dimensional feature of image I, W is the image width, H is the image height, d is the number of used features, ϕ_ℓ is a mapping relating the image with the ℓ-th feature from the image I. For a given rectangular region R, the feature points can be represented as $d \times d$ covariance matrix:

$$C_R = \frac{1}{n-1} \sum_{k=1}^{n_p} (z_k - \mu)(z_k - \mu)^\top, \tag{4.2}$$

where μ is the mean of the points, C_R is the covariance matrix of the feature vector F, z_k is the feature vector of pixel k, n_p pixels are chosen. The diagonal entries of the covariance matrix represent the variance of each feature, the rest entries of the matrix represent the correlation between different features. The covariance C_R of a given region R does not have any information regarding the order and the number of points.

Based on the optical flow and the intensity, 13 different feature vectors F shown in TABLE 4.1 are proposed to construct the covariance descriptor. Where I is the intensity of the gray image, the optical flow is obtained from the gray image, u is the horizontal optical flow, v is the vertical optical flow; I_x, u_x, v_x and I_y, u_y v_y are the first derivatives of the intensity, horizontal optical flow and vertical optical flow in the x direction and y direction; I_{xx}, u_{xx}, v_{xx} and I_{yy}, u_{yy}, v_{yy} are the second derivatives of the corresponding features in the x direction and y direction; I_{xy}, u_{xy} and v_{xy} are the second derivatives in the y direction of the first derivatives in the x direction of the corresponding features. Fig.4.1 illustrates the covariance matrix feature of the blobs, for the k-th blob in i-th frame B_i^k, covariance matrix feature is C_i^k. The optical flow shows the inter-frame information, it describes the movement information. The intensity shows the intra-frame information, it encodes the appearance information. If the whole frame is taken as a big blob, the covariance matrix descriptor of i-th frame is C_i.

4.2 Abnormal blob detection and localization

Based on the covariance matrix descriptor, we introduce the abnormal blob detection method in this section by three parts. Firstly, one-class support vector machines (OC-SVM) is briefly introduced. The second part proposes the multi-kernel strategy for the covariance matrix descriptor. The third part is the description of the global strategy of the abnormal blob detection method via one-class SVM. If the global image is taken as one blob,

		Feature Vector F
optical flow	$F_1(4\times4)$	$[y\ x\ u\ v\ ]$
	$F_2(6\times6)$	$[y\ x\ u\ v\ u_x\ u_y]$
	$F_3(6\times6)$	$[y\ x\ u\ v\ v_x\ v_y]$
	$F_4(8\times8)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y]$
	$F_5(12\times12)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}]$
	$F_6(14\times14)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ u_{xy}\ v_{xy}]$
optical flow with intensity	$F_7(5\times5)$	$[y\ x\ u\ v\ I]$
	$F_8(9\times9)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ I]$
	$F_9(13\times13)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ I]$
	$F_{10}(15\times15)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ u_{xy}\ v_{xy}\ I]$
	$F_{11}(11\times11)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ I\ I_x\ I_y]$
	$F_{12}(17\times17)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ I\ I_x\ I_y\ I_{xx}\ I_{yy}]$
	$F_{13}(20\times20)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ u_{xy}\ v_{xy}\ I\ I_x\ I_y\ I_{xx}\ I_{yy}\ I_{xy}]$

Table 4.1: Features F used to form the covariance matrices. For example, $F_1(4\times4)$ means the covariance matrix (COV) descriptor is in size 4×4.

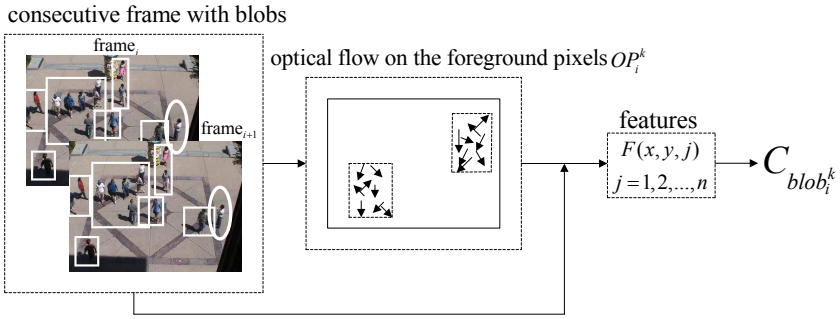


Figure 4.1: Computation of the covariance matrix (COV) descriptor of the blob.

the strategy of the abnormal blob detection method can also detect global abnormal frame events.

4.2.1 Nonlinear One-class SVM

The problem of non-linear one-class SVM [Schölkopf 2001, Canu 2005] can be presented as a constrained minimization one:

$$\min_{\omega,\xi,\rho} \ \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho, \tag{4.3}$$

$$\text{subject to: } \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle \geq \rho - \xi_i, \ \ \xi_i \geq 0. \tag{4.4}$$

The decision function in the data space $\mathcal{X}$ is defined as:

$$f(\boldsymbol{x}) = \mathbf{sgn}(\sum_{i=1}^{n} \alpha_i \kappa(\boldsymbol{x_i}, \boldsymbol{x}) - \rho), \tag{4.5}$$

where $\boldsymbol{x}$ is a vector in the input data space $\mathcal{X}$, κ is the kernel function implicitly mapping the data into a higher dimensional feature space where a linear classifier can be designed.

4.2.2 Kernel for Covariance Matrix Descriptor

For one-class SVM, the kernel κ of two covariance matrices must be computed. If proper parameters are given, the traditionally used kernel, such as the Gaussian, polynomial, and sigmoidal kernel, has similar performances [Schölkopf 2002]. We choose the Gaussian kernel defined by the following expression:

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}), \quad (\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{X} \times \mathcal{X}, \tag{4.6}$$

where the parameter σ indicates the scale factor where the data should be clustered, $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two vectors.

The covariance matrix is an element in a Lie Group G, where the distance measuring the dissimilarity of two elements is defined as:

$$d(\boldsymbol{X}_1, \boldsymbol{X}_2) = \| \log(\boldsymbol{X}_1^{-1} \boldsymbol{X}_2)\|, \tag{4.7}$$

$$\text{with } \|\boldsymbol{A}\| = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}, \tag{4.8}$$

where $\|\cdot\|$ is the Frobenius norm, a_{ij} is an element in the matrix $\boldsymbol{A}$, $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are the matrices in a Lie Group G. Thus, the Gaussian kernel in a Lie Group G is:

$$\kappa(\boldsymbol{X}_i, \boldsymbol{X}_j) = \exp(-\frac{\| \log(\boldsymbol{X}_i^{-1} \boldsymbol{X}_j)\|}{2\sigma^2}), \quad (\boldsymbol{X}_i, \boldsymbol{X}_j) \in G \times G. \tag{4.9}$$

The Baker Campbell Hausdorff formula [Hall 2003] in the theory of Lie Group is:

$$\log(\exp \boldsymbol{X} \exp \boldsymbol{Y}) = \sum_{n>0} \frac{(-1)^{n-1}}{n} \sum_{\substack{r_i+s_i>0 \\ 1 \le i \le n}} \frac{(\sum_{i=1}^{n}(r_i + s_i))^{-1}}{r_1! s_1! \cdots r_n! s_n!} [\boldsymbol{X}^{r_1} \boldsymbol{Y}^{s_1} \boldsymbol{X}^{r_2} \boldsymbol{Y}^{s_2} \ldots \boldsymbol{X}^{r_n} \boldsymbol{Y}^{s_n}]. \tag{4.10}$$

By using the first term of eq.(4.10), the approximate form of the Gaussian kernel in Lie Group is:

$$\kappa(\boldsymbol{X}_i, \boldsymbol{X}_j) = \exp(-\frac{\| \log(\boldsymbol{X}_i) - \log(\boldsymbol{X}_j)\|^2}{2\sigma^2}), \quad (\boldsymbol{X}_i, \boldsymbol{X}_j) \in G \times G, \tag{4.11}$$

where $\log(\boldsymbol{X})$ is a symmetrical matrix. The covariance descriptor $\boldsymbol{C}_R$ is of size $d \times d$, due to symmetry $\boldsymbol{C}_R$ has only $\frac{d^2+d}{2}$ different features. By choosing the $\frac{d^2+d}{2}$ upper triangular

and the diagonal elements of the matrix $\log(\boldsymbol{X})$ to construct a vector $\overline{\boldsymbol{x}}$, replacing $\log(\boldsymbol{X})$ in eq.(A.19) by $\overline{\boldsymbol{x}}$, the Gaussian kernel can be written as:

$$\kappa(\boldsymbol{X}_i, \boldsymbol{X}_j) = \exp(-\frac{\|\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}_j\|^2}{2\sigma^2}), \tag{4.12}$$

where $\overline{\boldsymbol{x}}_i$ is the vector constructed by elements of the upper triangular and the diagonal elements of the matrix $\log(\boldsymbol{X})$.

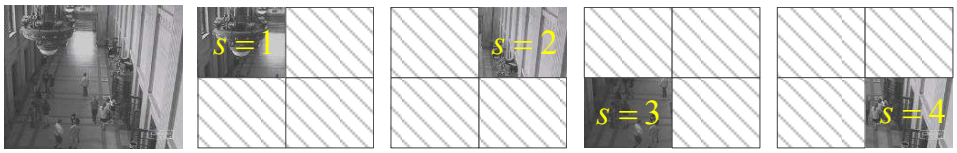
For constructing a more representative and discriminative feature descriptor, we split each frame into m parts. The multi-kernel strategy for our covariance matrix descriptor is defined by [Noumir 2012a, Rakotomamonjy 2008, Chen 2013]:

$$\kappa(\boldsymbol{X}_i, \boldsymbol{X}_j) = \sum_{s=1}^{m} \mu_s \kappa_s(\overline{\boldsymbol{x}}_i, \overline{\boldsymbol{x}}_j). \tag{4.13}$$

Eq.(A.21) is a kernel consisting of m basic kernels $\kappa_s, s = 1, \cdots, m$. Because each basic kernel satisfies Mercer condition, their summation is also a semi-positive definite kernel under the condition of non-negative μ_s. In this expression, the Gaussian kernel is adopted with:

$$\kappa_s(\overline{\boldsymbol{x}}_i, \overline{\boldsymbol{x}}_j) = \exp(-\frac{\|\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}_j\|^2_{[s]}}{2\sigma^2}). \tag{4.14}$$

The kernels $\kappa_s, s = 1, \cdots, m$ are Gaussian kernels. Each sample vector $\overline{\boldsymbol{x}}$ consists of m parts, $[\overline{\boldsymbol{x}}_1, \overline{\boldsymbol{x}}_2, \ldots, \overline{\boldsymbol{x}}_m]$. This kernel strategy is similar to filter the frame by using a mask. For example, a frame is split into 4 parts, as shown in Fig.4.2. If $s = 1$, the left-up part of the image is selected. We preset the weight μ_s according to the characteristic of the image to tune the importance of each sub-image. In the indoor scene, in the normal and the abnormal frames, there are no people in the upper half of the image. Thus, we set $\mu_{1,2} = 0.1$, $\mu_{3,4} = 0.4$ to reduce the importance of the sub-image where $s = 1$ and $s = 2$. In this case, since $\mu_s \geq 0$ and $\sum_{s=1}^{4} \mu_s = 1$, the resulting kernel belongs to the convex hull of the 4 considered kernels. By considering this combination, the resulting kernel out performs each kernel κ_s used individually.



(a) Image (b) $S = 1$ (c) $S = 2$ (d) $S = 3$ (e) $S = 4$

Figure 4.2: Filter the image by the mask to select a sub-image. (a) An original frame of the indoor scene. (b) $S = 1$, $\mu_1 = 0.1$, the left-upper part of the image is selected. (c) $S = 2$, $\mu_2 = 0.1$, the right-upper part. (d) $S = 3$, $\mu_3 = 0.4$, the left-lower part. (e) $S = 4$, $\mu_4 = 0.4$, the right-lower part.

4.3 Abnormal Events Detection and Localization Results

This section presents the results of experiments conducted to analyze the performance of the proposed method for abnormal blob localization, and global abnormal frame events detection. In the experiments below, if a frame is split into 4 parts, the frame feature consists of 4 covariance matrix descriptors, we mark it as "*4 covariances*", otherwise, we mark the feature as "*1 covariance*". If the multi-kernel strategy is used, we mark it as "*4 kernels*", otherwise, we mark the kernel strategy as "*1 kernel*".

4.3.1 Abnormal Blob Detection Results

The samples for training and the normal samples for testing are the blobs where people are walking. The abnormal samples correspond to the blobs where people are running. Our method can distinguish the abnormal running blobs from the walking blobs. In ROC curve, the true positive rate means that the running blob is classified as abnormal, while the false positive rate means that the walking blob is classified as abnormal.

The detection results of a scene of two pedestrians moving parallel to the camera plane are shown in Fig.4.3. It simulates the abnormal scenes where the velocity of the object changes. The sequence is of a low resolution, the people have a height about 30 pixels. The maximum AUC value is 0.8759.

The detection results of the lawn scene and the plaza scene in UMN dataset [UMN 2006] are shown in Fig.4.4. The maximum AUC value of the lawn scene is 0.9721, of the plaza scene is 0.8523. The results show that the abnormal detection algorithm of the *blob* covariance feature can obtain satisfactory detection results.

The detection results of mall scenes [Adam 2008] are shown in Fig.4.5. In one frame, there are walking people and also the running ones. The maximum AUC value is 0.8583.

The AUC of the detection results of different scenes and different covariance features are summarized in the TABLE 4.2. Generally, the features including both optical flow and intensity induce better detection results than the ones where only the optical flow is considering.

4.3.2 Abnormal Frame Detection Results

Taking the global *frame* as one blob, the abnormal *blob* detection method can be adjusted to detect abnormal *frame*. The detection results of UMN dataset [UMN 2006] and PETS2009 dataset [PETS 2009] are introduced below.

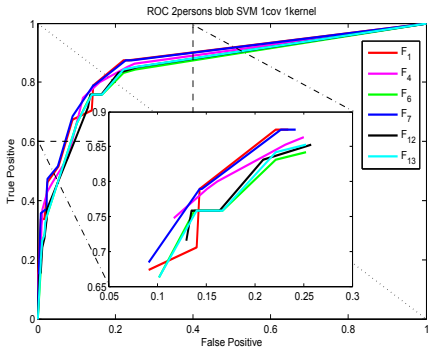
4.3.2.1 Abnormal Frame Detection Results of the UMN dataset

The UMN dataset includes eleven video sequences of three different scenes of crowded escape events. The detection results of lawn scene, plaza scene and indoor scene are shown in Fig.4.6, Fig.4.7 and Fig.4.8, respectively. The training samples and normal testing samples are the frames where the people are walking in different directions. The abnormal testing samples are the frames where the people are running. The *"1 covariance descriptor and 1 kernel"* strategy results are shown in TABLE 4.3, the *"4 covariance descriptors and*

(a) A training frame

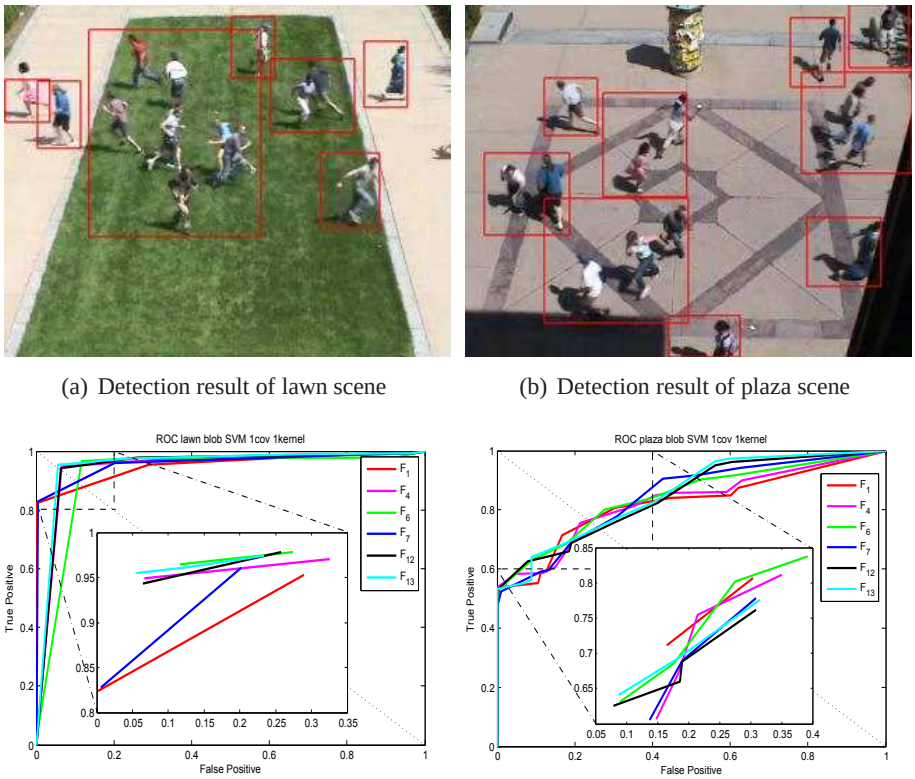
(b) A running person

(c) Optical flow image



(d) ROC via SVM

Figure 4.3: Abnormal *blob* event detection results of the two people walking or running scene based on *blob* covariance matrix descriptor via one-class SVM. (a) The normal scene for training, two people are walking. (b) The detection result. The red rectangle labels the abnormal blob, the person is running. The blue rectangle labels the normal blob, the person is walking. (c) The optical flow image of (b). A black border is added to show the image clearly. (d) ROC curve of different feature F results by using *"1 covariance descriptor and 1 kernel"*. The maximum AUC value is 0.8759.



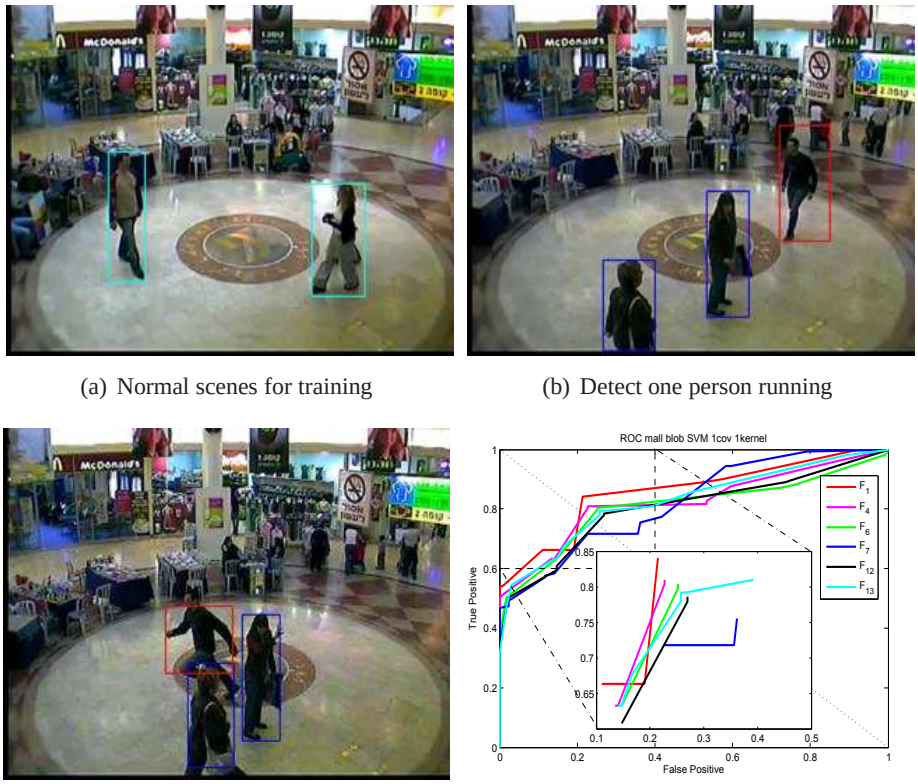
(a) Detection result of lawn scene

(b) Detection result of plaza scene

(c) ROC curve of lawn

(d) ROC curve of plaza

Figure 4.4: Abnormal *blob* event detection results of UMN dataset based on *blob* covariance matrix descriptor via one-class SVM, the abnormal *blob* event localization results of the lawn scene and the plaza scene. (a) The abnormal detection results of lawn scene. All the people are running. The red rectangles label the abnormal running blobs. (b) The abnormal detection results of plaza scene. (c) ROC curve of different feature F results of the lawn scene results by using *"1 covariance descriptor and 1 kernel"*. The maximum AUC value is 0.9721. (d) ROC curve of different feature F results of the plaza scene by using *"1 covariance descriptor and 1 kernel"*. The maximum AUC value is 0.8523.



(a) Normal scenes for training

(b) Detect one person running

(c) Detect one person running

(d) ROC curve of mall

Figure 4.5: Abnormal *blob* event detection results of the mall scene based on *blob* covariance matrix descriptor via one-class SVM. (a) The normal blobs for training, two people are walking. (b) The detection result. The red rectangles label the abnormal blobs, the people are running. The blue rectangles label the normal blobs, the people are walking. (c) Another abnormal *blob* event detection result. (d) ROC curve by using *"1 covariance descriptor and 1 kernel"*. The maximum AUC value is 0.8583.

Features		2persons	lawn	plaza	mall
Blob one-class SVM *1 covariance 1 kernel*					
optical flow	$F_1(4\times4)$	0.8739 ②	0.9504 ⑨	0.8200 ⑬	***0.8583*** ①
	$F_2(6\times6)$	0.8645 ⑦	0.9562 ⑥	0.8201 ⑫	0.8359 ③
	$F_3(6\times6)$	0.8700 ③	0.9533 ⑧	0.8289 ⑩	0.7934 ⑬
	$F_4(8\times8)$	0.8654 ⑥	0.9424 ⑫	0.8275 ⑪	0.8240 ⑦
	$F_5(12\times12)$	0.8523 ⑩	0.9649 ②	0.8430 ⑦	0.8066 ⑪
	$F_6(14\times14)$	0.8500 ⑫	0.9218 ⑬	0.8449 ④	0.8071 ⑩
optical flow with intensity	$F_7(5\times5)$	***0.8759*** ①	0.9591 ⑤	0.8439 ⑥	0.8217 ⑧
	$F_8(9\times9)$	0.8660 ⑤	0.9637 ③	0.9426 ⑧	0.8340 ④
	$F_9(13\times13)$	0.8521 ⑪	0.9441 ⑪	0.8442 ⑤	0.8248 ⑥
	$F_{10}(15\times15)$	0.8500 ⑬	0.9625 ④	0.8499 ③	0.8110 ⑨
	$F_{11}(11\times11)$	0.8665 ④	***0.9721*** ①	0.8380 ⑨	0.8404 ②
	$F_{12}(17\times17)$	0.8525 ⑨	0.9474 ⑩	0.8466 ②	0.8028 ⑫
	$F_{13}(20\times20)$	0.8546 ⑧	0.9541 ⑦	***0.8523*** ①	0.8266 ⑤

Table 4.2: AUC of abnormal *blob* event detection results based on *blob* covariance matrix descriptor constructed from different covariance features F via one-class SVM (OC-SVM) by using "*1 covariance descriptor and 1 kernel*". The biggest value of each scene is shown in bold and red color.

1 kernel" strategy results are shown in TABLE 4.4, the *"4 covariance descriptors and 4 kernels"* multi-kernel strategy results are shown in TABLE 4.5. The indoor scene is more difficult than the other two scenes, due to the instable illumination situation and the gloom circumstance. The camera is far away from the moving people. When some people come into or go out from the room, the illumination becomes much stronger. Our proposed abnormal detection method can handle a this bad illumination scene, and obtain satisfactory detection results.

By comparing the results of all these three senses in TABLE 4.3 and TABLE 4.4, we can see that splitting a frame into 4 parts can generally improve the performance of abnormal detection results. By comparing the results of "indoor" and "indoor♯" in TABLE 4.5, we can see by choosing suitable coefficients of the multi-kernel strategy to adapt the characteristic of the scene, the performances are much better in every feature.

By comparing the abnormal *blob* detection results in TABLE 4.2 and the abnormal *frame* detection results, we can see that abnormal *frame* detection performance is a little better than the abnormal *blob* detection performance. In fact the blob detection method cannot label all the people very exactly. The rectangle may be on the background, or does not include all the parts of the human. These are the major reasons of lower AUC value of the *blob* feature based method. Nevertheless, the abnormal *blob* detection can obtain similar performance as abnormal global *frame* detection by presetting a threshold of the percentage of blobs in one frame. For example, if 80% of the blobs in one frame are classified as abnormal, this frame is then considered as an abnormal frame. Thus, the abnormal *blob* detection has the results as the same as the ones when the covariance of a

Features		lawn	indoor	plaza
Frame one-class SVM *1 covariance 1 kernel*				
optical flow	$F_1(4\times4)$	0.9382 ⑫	0.7359 ⑬	0.9103 ⑬
	$F_2(6\times6)$	0.9474 ⑪	0.8381 ⑩	0.9148 ⑫
	$F_3(6\times6)$	0.9583 ⑩	0.8410 ⑨	0.9192 ⑪
	$F_4(8\times8)$	0.9656 ⑦	0.8483 ⑧	0.9367 ⑨
	$F_5(12\times12)$	0.9798 ②	0.8744 ⑥	0.9782 ②
	$F_6(14\times14)$	***0.9803*** ①	0.8752 ⑤	***0.9790*** ①
optical flow with intensity	$F_7(5\times5)$	0.9337 ⑬	0.8314 ⑪	0.9220 ⑩
	$F_8(9\times9)$	0.9617 ⑧	0.8529 ⑦	0.9419 ⑧
	$F_9(13\times13)$	0.9786 ④	0.8797 ④	0.9721 ④
	$F_{10}(15\times15)$	0.9789 ③	0.8145 ⑫	0.9734 ③
	$F_{11}(11\times11)$	0.9583 ⑨	0.9000 ③	0.9472 ⑦
	$F_{12}(17\times17)$	0.9758 ⑥	***0.9291*** ①	0.9549 ⑥
	$F_{13}(20\times20)$	0.9767 ⑤	0.9253 ②	0.9580 ⑤

Table 4.3: AUC of abnormal *frame* event detection results based on *frame* covariance matrix descriptor constructed from different features F via one-class SVM (OC-SVM) by using *"1 covariance descriptor and 1 kernel"* of the UMN dataset.

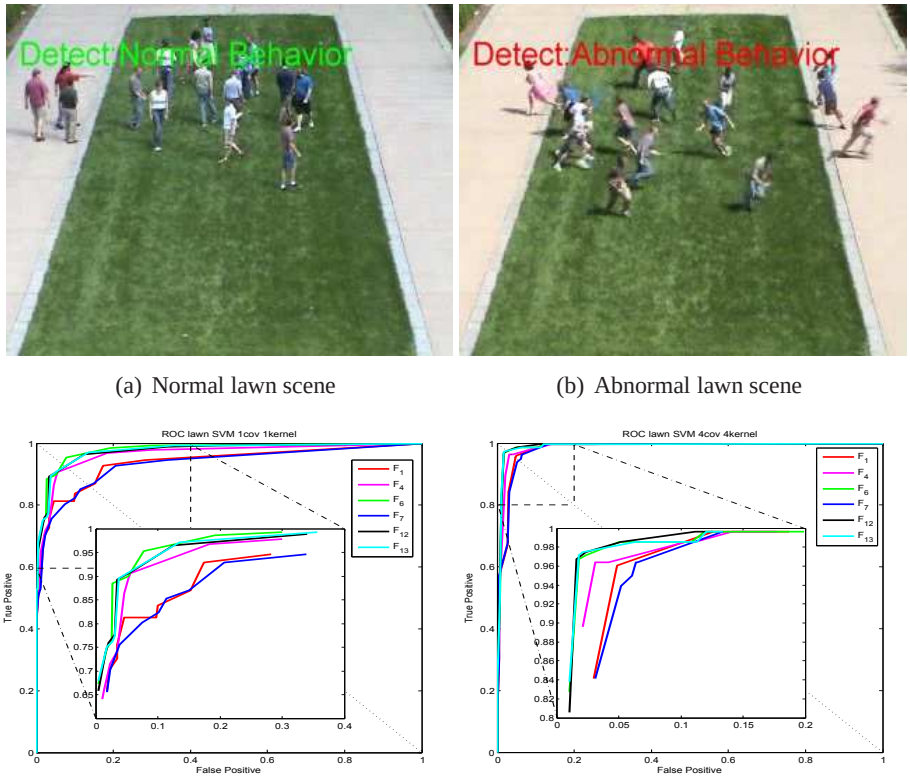
frame is chosen as a descriptor.

The performances of the covariance matrix descriptor based method and the state-of-the-art methods are shown in TABLE 4.6. The covariance matrix based multi-kernel learning strategy abnormal frame detection method obtains competitive performance. Our method is better than others except sparse reconstruction cost (SRC) [Cong 2011], which takes multi-scale HOF as a feature, classifies a testing sample by its sparse reconstruction cost, through a weighted linear reconstruction of the over-complete normal basis set. For a particular scene, the kernel coefficients in the multi-kernel strategy can be tuned to obtain a better performance. By using the integral image strategy [Tuzel 2006], the covariance matrix descriptor of the *blob* can be computed quickly from the global *frame* covariance. Because our abnormal detection method can detect abnormal global *frame* and abnormal *blob*, we can localize the *blob* in the abnormal *frame* conveniently.

4.3.2.2 Abnormal Frame Detection results of the PETS dataset

The covariance descriptor can not only encode the magnitude information of a frame, and also describe the direction. The detection results of *Time 14-17* scene are show in Fig.4.9. The training samples and normal testing samples are chosen from the sequence (*Time 14-55*), where the people are walking in different directions. The abnormal testing samples are chosen from the sequence (*Time 14-17*), where the people are walking or running in one direction. The proposed abnormal detection method detect the one direction movement, the maximum AUC value is 0.9662.

The detection results of crowd splitting sequence (*Time 14-31*) are shown in Fig.4.10.



(a) Normal lawn scene

(b) Abnormal lawn scene

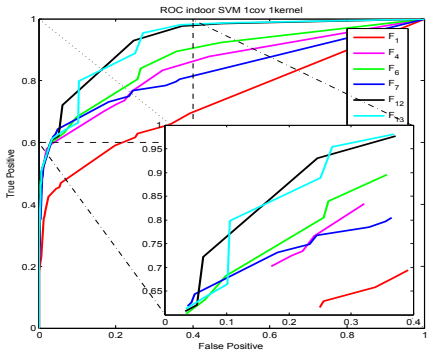
(c) ROC 1covariance 1kernel

(d) ROC 4covariances 4kernels

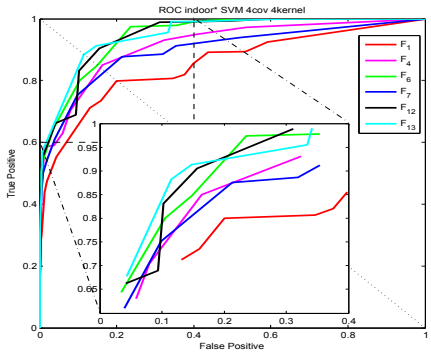
Figure 4.6: Abnormal *frame* event detection results of the indoor scene based on *original frame* covariance descriptor via one-class SVM. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) ROC curve of different feature F results by using *"1 covariance descriptor and 1 kernel"*. The maximum AUC value is 0.9803. (d) ROC curve by using *"4 covariance descriptors and 4 kernels"*, $\sum_{s=1}^{4} \mu_s \kappa_s$, $\mu_{1,2,3,4} = 0.25$. The maximum AUC value is 0.9900.

(a) Normal indoor scene

(b) Abnormal indoor scene

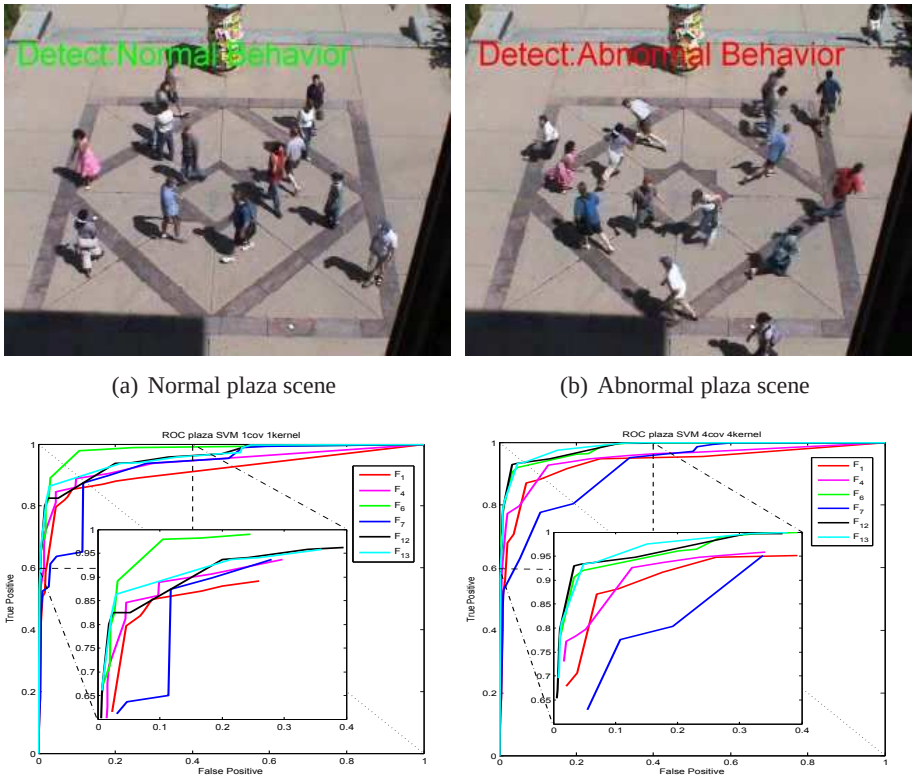


(c) ROC 1covariance 1kernel



(d) ROC 4covariances 4kernels

Figure 4.7: Abnormal *frame* event detection results of the indoor scene based on *original frame* covariance descriptor via one-class SVM. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) ROC curve by using *"1 covariance descriptor and 1 kernel"*. The maximum AUC value is 0.9291. (d) ROC curve by using *"4 covariance descriptors and 4 kernels"*, $\sum_{s=1}^{4} \mu_s \kappa_s$, $\mu_{1,2} = 0.1$, $\mu_{3,4} = 0.4$. The maximum AUC value is 0.9522.



(a) Normal plaza scene

(b) Abnormal plaza scene

(c) ROC 1covariance 1kernel

(d) ROC 4covariances 4kernels

Figure 4.8: Abnormal *frame* event detection results of the plaza scene based on *original frame* covariance descriptor via one-class SVM. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) ROC curve by using *"1 covariance descriptor and 1 kernel"*. The maximum AUC value is 0.9790. (d) ROC curve by using *"4 covariance descriptors and 4 kernels"*, $\sum_{s=1}^{4} \mu_s \kappa_s$, $\mu_{1,2,3,4} = 0.25$. The maximum AUC value is 0.9829.

Features		lawn	indoor	plaza
Frame one-class SVM *4 covariances 1kernle*				
optical flow	$F_1(4\times4)$	0.9868 ⑫	0.8473 ⑬	0.9372 ⑫
	$F_2(6\times6)$	***0.9920*** ①	0.8637 ⑩	0.9486 ⑪
	$F_3(6\times6)$	0.9905 ②	0.8801 ⑨	0.9498 ⑩
	$F_4(8\times8)$	0.9879 ⑨	0.8736 ⑩	0.9502 ⑨
	$F_5(12\times12)$	0.9888 ⑥	0.9072 ④	0.9738 ⑤
	$F_6(14\times14)$	0.9891 ④	0.9045 ⑤	0.9735 ⑥
optical flow with intensity	$F_7(5\times5)$	0.9868 ⑫	0.8676 ⑪	0.9417 ⑬
	$F_8(9\times9)$	0.9874 ⑩	0.8818 ⑧	0.9599 ⑧
	$F_9(13\times13)$	0.9889 ⑤	0.9102 ③	0.9775 ③
	$F_{10}(15\times15)$	0.9890 ③	0.8878 ⑦	0.9761 ④
	$F_{11}(11\times11)$	0.9873 ⑪	0.8943 ⑥	0.9639 ⑦
	$F_{12}(17\times17)$	0.9883 ⑦	***0.9151*** ①	***0.9818*** ①
	$F_{13}(20\times20)$	0.9882 ⑧	0.9148 ②	0.9810 ②

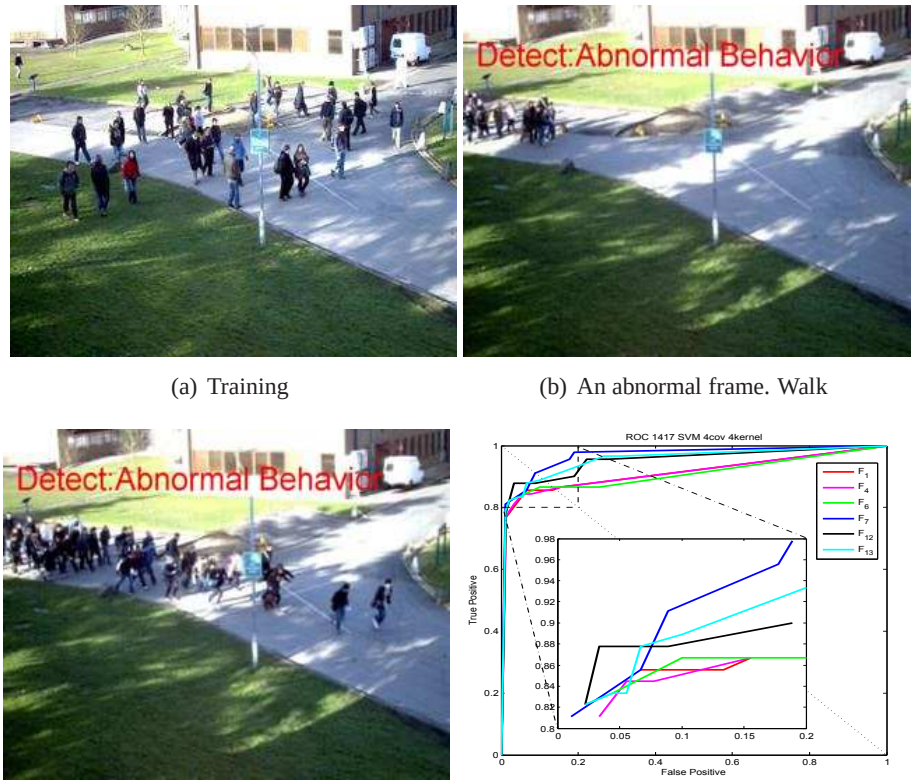
Table 4.4: AUC of abnormal *frame* event detection results based on *frame* covariance matrix descriptor constructed from different features F via one-class SVM (OC-SVM) by using *"4 covariance descriptors and 1 kernel"* of the UMN dataset.

Features		lawn	indoor	indoor ♯	plaza
Frame one-class SVM *4 covariances 4 kernels*					
optical flow	$F_1(4\times4)$	0.9828 ⑫	0.8381 ⑬	***0.9522*** ①	0.9374 ⑫
	$F_2(6\times6)$	0.9866 ⑨	0.8840 ⑪	0.9007 ⑬	0.9441 ⑪
	$F_3(6\times6)$	0.9870 ⑦	0.8971 ⑩	0.9136 ⑪	0.9454 ⑩
	$F_4(8\times8)$	0.9863 ⑩	0.9008 ⑧	0.9141 ⑩	0.9485 ⑨
	$F_5(12\times12)$	***0.9900*** ①	0.9344 ②	0.9422 ⑤	0.9783 ④
	$F_6(14\times14)$	0.9895 ④	0.9318 ③	0.9442 ④	0.9790 ③
optical flow with intensity	$F_7(5\times5)$	0.9817 ⑬	0.8714 ⑫	0.8976 ⑫	0.9153 ⑬
	$F_8(9\times9)$	0.9862 ⑪	0.9088 ⑦	0.9245 ⑧	0.9506 ⑧
	$F_9(13\times13)$	0.9899 ③	0.9309 ⑤	0.9416 ⑥	0.9763 ⑥
	$F_{10}(15\times15)$	0.9894 ⑥	0.8982 ⑨	0.9242 ⑨	0.9767⑤
	$F_{11}(11\times11)$	0.9870 ⑧	0.9298 ⑥	0.9289 ⑦	0.9555 ⑦
	$F_{12}(17\times17)$	0.9899 ②	0.9310 ④	0.9453 ③	0.9809 ②
	$F_{13}(20\times20)$	0.9895 ⑤	***0.9365*** ①	0.9484 ②	***0.9829*** ①

Table 4.5: AUC of abnormal *frame* event detection results of the UMN dataset by using *"4 covariance descriptors and 4 kernels"*, $\sum_{s=1}^{4}\mu_s\kappa_s$. The "indoor♯" means $\mu_{1,2}=0.1$, $\mu_{3,4}=0.4$. The other results obtained by using $\mu_{1,2,3,4}=0.25$.

Method	Area under ROC		
	lawn	indoor	plaza
Social Force [Mehran 2009]	0.96		
Optical Flow [Mehran 2009]	0.84		
NN [Cong 2011]	0.93		
SRC [Cong 2011]	0.995	0.975	0.964
STCOG [Shi 2010]	0.9362	0.7759	0.9661
COV SVM (Ours)	0.9920	0.9522	0.9829

Table 4.6: The comparison of our proposed covariance matrix descriptor and one-class SVM based method with the state-of-the-art methods for abnormal *frame* event detection of the UMN dataset.

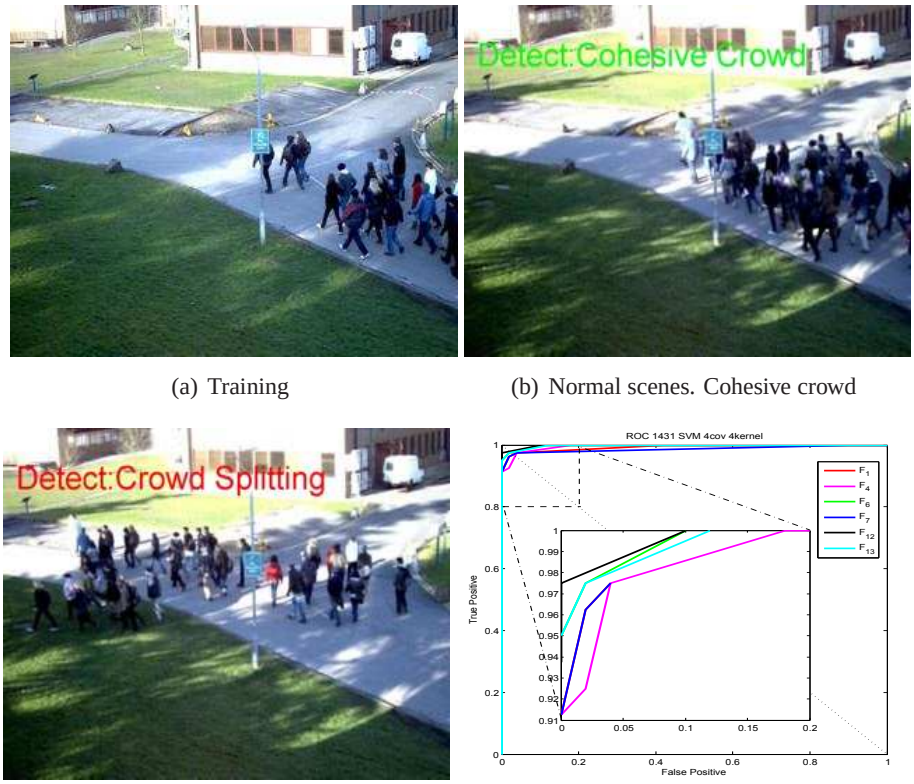


(a) Training (b) An abnormal frame. Walk

(c) An abnormal frame. Move (d) ROC 4covariances 4kernels

Figure 4.9: Abnormal *frame* event detection results of *Time14-17* based on *original frame* covariance matrix descriptor via one-class SVM. (a) A training frame (*Time 14-55*). The people are walking in different directions. (b) An abnormal frame (*Time 14-17*). The people are walking in the same direction. (c) An abnormal frame (*Time 14-17*). The people are moving (walking or running) in the same direction. (d) ROC curve by using *"4 covariance descriptors and 4 kernels"*. The biggest AUC value is 0.9662.

The training samples are chosen from the scene (*Time 14-16*), where there is one cohesive crowd. The normal and abnormal testing samples are chosen from sequence *Time 14-31*. The abnormal scene is the frames where the crowd is splitting. The maximum AUC value is 0.9988. The detection results of *Time 14-17* and *Time 14-31* are shown in TABLE 4.7. By using the multi-kernel learning strategy, the performance of the detection results are improved.



(a) Training (b) Normal scenes. Cohesive crowd

(c) Abnormal scenes. Crowds split (d) ROC 4covariances 4kernels

Figure 4.10: Abnormal *frame* event detection results of *Time14-31* based on *original frame* covariance matrix descriptor via one-class SVM. (a) A training frame. A people cohesive crowd (*Time 14-16*) in the frame. 41 training frames (0 to 40) are chosen from *Time14-16*. (b) A normal testing frame *Time14-31* . (c) A people cohesive crowd abnormal frame. Multiple diverging flows *Time14-31*. (c) ROC curve by using *"4 covariance descriptors and 4 kernels"*. The biggest AUC value is 0.9988.

4.4 Conclusion

The covariance matrix descriptor constructed by different features of the intensity and the optical flow is proposed to encode the moving information of a blob or a frame. The influence of the different features is analyzed by experiments. The covariance matrix descriptor can be computed conveniently from the frame to the blob by adopting integral image. A

Features		1417	1431	1417 $*$	1431 $*$	1417 ♯	1431 ♯
Frame one-class SVM							
		1 covariance 1 kernel		*4 covariances 1 kernel*		*4 covariances 4 kernels*	
optical flow	$F_1(4\times4)$	0.7357 ⑥	0.6341 ⑫	0.9275 ⑬	***0.9953*** ①	0.9136 ⑨	0.9934 ⑪
	$F_2(6\times6)$	0.7283 ⑪	0.6650 ⑨	0.9391 ⑪	0.9911 ⑦	0.9214 ⑧	0.9973 ⑧
	$F_3(6\times6)$	***0.7541*** ①	0.7291 ⑤	0.9378 ⑫	0.9900 ⑩	0.9059 ⑬	0.9960 ⑨
	$F_4(8\times8)$	0.7196 ⑬	0.7145 ⑥	0.9432 ⑧	0.9951 ②	0.9125 ⑪	0.9956 ⑩
	$F_5(12\times12)$	0.7388 ⑤	0.8256 ②	0.9412 ⑨	0.9905 ⑧	0.9135 ⑩	0.9981 ④
	$F_6(14\times14)$	0.7314 ⑦	***0.8258*** ①	0.9402 ⑩	0.9884 ⑫	0.9081 ⑫	0.9983 ②
optical flow with intensity	$F_7(5\times5)$	0.7396 ③	0.5464 ⑬	0.9463 ⑤	0.9923 ⑤	***0.9662*** ①	0.9874 ⑥
	$F_8(9\times9)$	0.7233 ⑫	0.6449 ⑩	***0.9490*** ①	0.9944 ③	0.9385 ⑤	0.9931 ⑫
	$F_9(13\times13)$	0.7396 ③	0.7886 ④	0.9453 ⑦	0.9901 ⑨	0.9235 ⑦	0.9974 ⑥
	$F_{10}(15\times15)$	0.7301 ⑧	0.7963 ③	0.9464 ④	0.9881 ⑬	0.9240 ⑥	0.9983 ②
	$F_{11}(11\times11)$	0.7294 ⑩	0.6448 ⑪	0.9460 ⑥	0.9935 ④	0.9546②	0.9914 ⑬
	$F_{12}(17\times17)$	0.7447 ②	0.6730 ⑧	0.9475 ②	0.9913 ⑥	0.9501 ④	***0.9988*** ①
	$F_{13}(20\times20)$	0.7301 ⑧	0.7070 ⑦	0.9474 ③	0.9898 ⑪	0.9546 ②	0.9980 ⑤

Table 4.7: AUC of abnormal *frame* event detection results based on *frame* covariance matrix descriptor constructed by different features F via one-class SVM (OC-SVM) of PETS dataset. "1417" and "1431" are the results by using *"1 covariance descriptor and 1 kernel"*. "1417 $*$" and "1431 $*$" are the results by using *"4 covariance descriptors and 1 kernel"*. "1417 ♯" and "1431 ♯" are the results by using *"4 covariance descriptors and 4 kernels"*, $\sum_{s=1}^{4}\mu_s\kappa_s$, $\mu_{1,2,3,4}=0.25$.

multi-kernel strategy is proposed to adapt the detection method to the characteristics of a particular scene, improving the detection results. The proposed method has been tested on several datasets, and it was shown that the proposed method is able to detect abnormal events both at the blob and the frame levels.

Chapter 5

Abnormal detection via online one-class SVM

Contents

In Chapter 3 and Chapter 4, the abnormal *blob* and *frame* event detection methods have been proposed. These methods are based on histograms of optical flow orientations (HOFO) descriptor or covariance matrix (COV) descriptor, and one-class support vector machines (OC-SVM) classification. SVM is usually trained in a batch model, i.e., all training data are given a priori and learning is conducted in one batch. If additional training data arrive later, the SVM must be retrained from scratch [Shilton 2005]. In the problem of abnormal event detection for videosurveillance, the normal sequence for training may last for a long time. It is impractical to train the whole big training set of normal samples as one batch. Moreover, if a new frames are added to a large training dataset, they will likely have only a minimal effect on the previous decision surface. Resolving the problem from scratch seems computationally wasteful. Considering these two aspects, the online strategy is adopted in our work to respect both the computational and memory requirements. Two online one-class SVM algorithms are introduced, the online support vector data description (online SVDD) and the online least squares one-class support vector machines (online LS-OC-SVM). The covariance matrix descriptor proposed in Section 4.1 is used in this chapter.

5.1 Abnormal detection via online support vector data description

In this section, we propose two strategies of abnormal event detection based on online support vector data description (SVDD). Before introducing these strategies, we first describe the online hypersphere one-class SVM classification method in the following.

5.1.1 Hypersphere one-class support vector machines

There are two frameworks for one-class SVM. One is the ν- support vector classifier (ν-SVC) introduced in [Schölkopf 2001], which is used in Chapter 3 and Chapter 4 for abnormal classification. The other is support vector data description (SVDD) which is presented in [Tax 2001, Tax 1999]. The SVDD method (considered in this chapter) computes a sphere shaped decision boundary with minimal volume around a set of objects. The center of the sphere $\boldsymbol{c}$ and the radius R are to be determined via the following optimization problem:

$$\min_{R,\xi,c} \quad R^2 + C\sum_{i=1}^{n}\xi_i, \tag{5.1}$$

$$\text{subject to: } \|\Phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0, \forall i, \tag{5.2}$$

where n is the number of training samples, ξ_i is the slack variable for penalizing the outliers. The hyperparameter C is the weight for restraining slack variables, it tunes the number of acceptable outliers. The nonlinear function $\Phi : \mathcal{X} \to \mathcal{H}$ maps a datum $\boldsymbol{x}_i$ into the feature space $\mathcal{H}$, it allows to solve a nonlinear classification problem by designing a linear classifier in the feature space $\mathcal{H}$. κ is the kernel function for computing dot products in $\mathcal{H}$, $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \langle\Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}')\rangle$. By introducing Lagrange multipliers, the dual problem associated with (5.2) is written by the following quadratic optimization problem:

$$\max_{\alpha} \quad \sum_{i=1}^{n}\alpha_i\kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{n}\alpha_i\alpha_j\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{5.3}$$

$$\text{subject to: } 0 \leq \alpha_i \leq C, \ \sum_{i=1}^{n}\alpha_i = 1, \ \boldsymbol{c} = \sum_{i=1}^{n}\alpha_i\Phi(\boldsymbol{x}_i). \tag{5.4}$$

The decision function is:

$$f(\boldsymbol{x}) = \mathbf{sgn}(R^2 - \sum_{i,j=1}^{n}\alpha_i\alpha_j\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) + 2\sum_{i=1}^{n}\alpha_i\kappa(\boldsymbol{x}_i, \boldsymbol{x}) - \kappa(\boldsymbol{x}, \boldsymbol{x})). \tag{5.5}$$

For the large training data, the solution cannot be obtained easily. An online strategy to train the data is used in our work. Let $\boldsymbol{c}_{\mathcal{D}}$ denotes a sparse model of the center $\boldsymbol{c}_n = \frac{1}{n}\sum_{i=1}^{n}\Phi(\boldsymbol{x}_i)$ by using a small subset of available samples which called dictionary:

$$\boldsymbol{c}_{\mathfrak{D}} = \sum_{i \in \mathfrak{D}} \alpha_i \Phi(\boldsymbol{x}_i), \tag{5.6}$$

where $\mathfrak{D} \subset \{1, 2, \ldots, n\}$, and let $N_{\mathfrak{D}}$ denotes the cardinality of this subset $\boldsymbol{x}_{\mathfrak{D}}$.

The distance between any mapped data $\Phi(\boldsymbol{x})$ respecting to the center $\boldsymbol{c}_{\mathfrak{D}}$ can be calculated by:

$$\|\Phi(\boldsymbol{x}) - \boldsymbol{c}_{\mathfrak{D}}\| = \sum_{i,j \in \mathfrak{D}} \alpha_i \alpha_j \, \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) - 2 \sum_{i \in \mathfrak{D}} \alpha_i \, \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + \kappa(\boldsymbol{x}, \boldsymbol{x}). \tag{5.7}$$

A modification of the original formulation of the one-class classification algorithm consisting of minimizing the approximation error $\|\boldsymbol{c}_n - \boldsymbol{c}_{\mathfrak{D}}\|$ is [Noumir 2012c, Noumir 2012b]:

$$\boldsymbol{\alpha} = \arg \min_{\alpha_i, i \in \mathfrak{D}} \|\frac{1}{n} \sum_{i=1}^{n} \Phi(\boldsymbol{x}_i) - \sum_{i \in \mathfrak{D}} \alpha_i \Phi(\boldsymbol{x}_i)\|^2. \tag{5.8}$$

The final solution is given by:

$$\boldsymbol{\alpha} = \boldsymbol{K}^{-1} \boldsymbol{\kappa}, \tag{5.9}$$

where $\boldsymbol{K}$ is the Gram matrix with (i, j)-th entry $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\boldsymbol{\kappa}$ is the column vector with entries $\frac{1}{n} \sum_{i=1}^{n} \kappa(\boldsymbol{x}_k, \boldsymbol{x}_i)$, $k \in \mathfrak{D}$.

In the online scheme, at each time step there is a new sample. Let $\boldsymbol{\alpha}_n$ denote the coefficients, $\boldsymbol{K}_n$ denote the Gram matrix, and $\boldsymbol{\kappa}_n$ denote the vector, at time step n. A criterion is used to determine whether the new sample can be included into the dictionary. A threshold μ_0 is preseted, for the datum $\boldsymbol{x}_t$ at time step t, the coherence-based sparsification criterion [Honeine 2012, Richard 2009] is:

$$\varepsilon_t = \max_{i \in \mathfrak{D}} \, |\kappa(\boldsymbol{x}_t, \boldsymbol{x}_{w_i})|, \tag{5.10}$$

First case: $\varepsilon_t > \mu_0$

In this case, the new data $\Phi(\boldsymbol{x}_{n+1})$ is not included into the dictionary. The Gram matrix $\boldsymbol{K}_{n+1} = \boldsymbol{K}_n$. $\boldsymbol{\kappa}_n$ changes online:

$$\boldsymbol{\kappa}_{n+1} = \frac{1}{n+1}(n\boldsymbol{\kappa}_n + \boldsymbol{b}) \tag{5.11}$$

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{K}_{n+1}^{-1} \boldsymbol{\kappa}_{n+1} = \frac{n}{n+1} \boldsymbol{\alpha}_n + \frac{1}{n+1} \boldsymbol{K}_n^{-1} \boldsymbol{b}. \tag{5.12}$$

where $\boldsymbol{b}$ is the column vector with entries $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{n+1})$, $i \in \mathfrak{D}$.

Second case: $\varepsilon_t \leq \mu_0$

In this case, the new data $\Phi(\boldsymbol{x}_{n+1})$ is included into the dictionary $\mathfrak{D}$. The Gram matrix $\boldsymbol{K}$ changes:

$$\boldsymbol{K}_{n+1} = \begin{bmatrix} \boldsymbol{K}_n & \boldsymbol{b} \\ \boldsymbol{b}^\top & \boldsymbol{\kappa}(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1}) \end{bmatrix}. \tag{5.13}$$

By using Woodbury matrix identity:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1} VA^{-1}, \tag{5.14}$$

$\boldsymbol{K}_{n+1}^{-1}$ can be calculated iteratively:

$$\boldsymbol{K}_{n+1}^{-1} = \begin{bmatrix} \boldsymbol{K}_n^{-1} & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{bmatrix} + \frac{1}{\kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1}) - \boldsymbol{b}^\top \boldsymbol{K}_n^{-1}\boldsymbol{b}} \times \begin{bmatrix} -\boldsymbol{K}_n^{-1}\boldsymbol{b} \\ 1 \end{bmatrix} \times \begin{bmatrix} -\boldsymbol{b}^\top \boldsymbol{K}_n^{-1} & 1 \end{bmatrix}. \tag{5.15}$$

The vector $\boldsymbol{\kappa}_{n+1}$ is updated from $\boldsymbol{\kappa}_n$,

$$\boldsymbol{\kappa}_{n+1} = \frac{1}{n+1}\begin{bmatrix} n\boldsymbol{\kappa}_n + \vec{b} \\ \kappa_{n+1} \end{bmatrix}, \tag{5.16}$$

$$\text{with } \kappa_{n+1} = \sum_{i=1}^{n+1} \kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_i). \tag{5.17}$$

Computing κ_{n+1} as eq.(5.17) needs to save all the samples $\{\boldsymbol{x}_{i=1}^{n+1}\}$ in memory. For conquering this issue, it can compute as $\kappa_{n+1} = (n+1)\kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1})$ by considering an instant estimation. The update of $\boldsymbol{\alpha}_{n+1}$ from $\boldsymbol{\alpha}_n$ is:

$$\begin{aligned} \boldsymbol{\alpha}_{n+1} =& \frac{1}{n+1}\begin{bmatrix} n\boldsymbol{\alpha}_n + \boldsymbol{K}_n^{-1}\boldsymbol{b} \\ 0 \end{bmatrix} \\ & - \frac{1}{(n+1)(\kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1}) - \boldsymbol{b}^\top \boldsymbol{K}_n^{-1}\boldsymbol{b})} \\ & \times \begin{bmatrix} \boldsymbol{K}_n^{-1}\boldsymbol{b} \\ 1 \end{bmatrix}\left(n\boldsymbol{b}^\top \boldsymbol{\alpha}_n + \boldsymbol{b}^\top \boldsymbol{K}_n^{-1}\boldsymbol{b} - \kappa_{n+1}\right). \end{aligned} \tag{5.18}$$

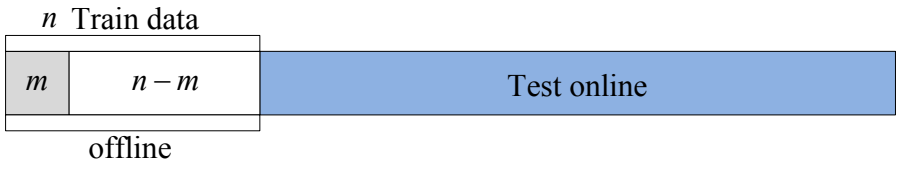
Based on eq.(5.18), we have an online implementation of the one-class SVM learning phase.

5.1.2 Abnormal Event detection

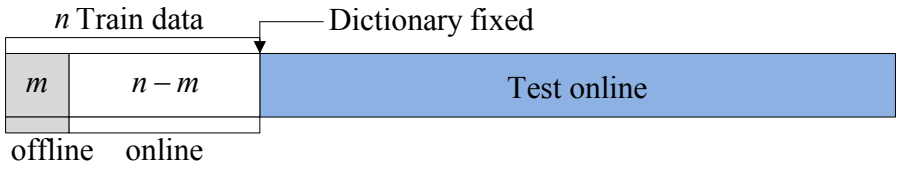
In an abnormal event detection problem, it is assumed that a set of training frames $\{I_1 \ldots I_n\}$ (the positive class) describing the normal behavior is obtained. The general architectures of online support vector data description (online SVDD) abnormal detection are introduced below.

The offline training strategy refers to the case where all the training samples are learnt as one batch, as shown in Fig.5.1(a). We propose two abnormal detection strategies, the difference between these two strategies is the time when the dictionary is fixed. These two strategies are shown in Fig.5.1(b) and (c). **Strategy 1** is shown in Fig.5.1(b). The

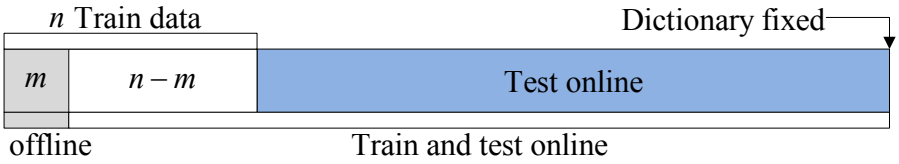
training data are learnt one-by-one. When the training period is finished, the dictionary and the classifier are fixed. Each test datum is classified based on the dictionary. Fig.5.1(c) illustrates **Strategy 2**. The training procedure is as the same as **Strategy 1**. But in the testing period, the dictionary is updated if the datum x_i satisfies the dictionary update condition. The details of these two strategies are explained below.



(a) Strategy offline



(b) Strategy 1



(c) Strategy 2

Figure 5.1: Offline and two online abnormal event detection strategies based on online support vector data description (SVDD). (a) Strategy offline. The training data are learnt as one batch offline. (b) Strategy 1. The dictionary is fixed when all the training data are learnt. (c) Strategy 2. The dictionary continues being updated through the testing period.

The abnormal *blob* events detection and abnormal *frame* events detection proposed in Chapter 3 and in Chapter 4 are in the same way in the one-class SVM classification processes, the difference is whether the HOFO descriptor or COV descriptor is calculated, in the blob or in the frame. Chapter 5 focuses on the online one-class SVM algorithm, so only COV descriptor is chosen, and the abnormal *frame* events detection task is considered.

5.1.2.1 Strategy 1

In Strategy 1, the dictionary is updated merely through the training period. The COV descriptor computation processes are the same as the ones in Chapter 4. After COV descriptor of each frame is calculated, the training and testing processes of online one-class SVM are introduced hereinbelow.

Step 1: The first step is calculating the covariance matrix descriptor of training frames based on the image intensity and the optical flow. This step can be generalized as:

$$\{(I_1, OP_1), (I_2, OP_2), \ldots, (I_n, OP_n)\} \longrightarrow \{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_n\}, \tag{5.19}$$

where $\{(I_1, OP_1), (I_2, OP_2), \ldots, (I_n, OP_n)\}$ are the image intensity and the corresponding optical flow of the 1^{st} to n^{th} frame. $\{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_n\}$ are the covariance matrix descriptors.

Step 2: The second step consists of applying one-class SVM on the small subset of extracted descriptor of the training normal frames to obtain the support vectors. Consider a subset $\{\boldsymbol{C}_i\}_{i=1}^{m}, 1 \leq m \ll n$ of data selected from the full training sample set $\{\boldsymbol{C}_i\}_{i=1}^{n}$, without loss of generality, assume that the first m examples are chosen. This set of m examples is called dictionary $\boldsymbol{C}_{\mathfrak{D}}$:

$$\{\boldsymbol{C}_1, \boldsymbol{C}_2 \ldots \boldsymbol{C}_m\}, 1 \leq m \ll n \overset{SVM}{\longrightarrow} \text{support vector } \{S\,p_1, S\,p_2, \ldots, S\,p_o\}, \tag{5.20}$$

where the set $\{\boldsymbol{C}_1, \boldsymbol{C}_2 \ldots \boldsymbol{C}_m\}$ is the first m covariance matrix descriptors of the training frames, it is the original dictionary $\boldsymbol{C}_{\mathfrak{D}}$. In one-class SVM, the majority of the training samples do not contribute to the definition of the decision function. The entries of a monitory subset of the training samples, $\{S\,p_1, S\,p_2, \ldots, S\,p_o\}, o \leq m$, are support vectors contributing to the definition of the decision function.

Step 3: After learning the dictionary $\boldsymbol{C}_{\mathfrak{D}}$ which includes the first $m, 1 \leq m \ll n$ samples, the training samples $\{\boldsymbol{C}_{m+1}, \boldsymbol{C}_{m+2}, \ldots, \boldsymbol{C}_n\}$ are learned online via the technique described in Section 5.1.1. This step can be generalized as:

$$\begin{aligned} &\{\boldsymbol{C}_{\mathfrak{D}}, \boldsymbol{C}_k\}, m < k \leq n \overset{SVM}{\longrightarrow} \\ &\text{support vector } \{S\,p_1, S\,p_2, \ldots, S\,p_p\}, o \leq p \leq n, \\ &\boldsymbol{C}_{\mathfrak{D}} := \boldsymbol{C}_{\mathfrak{D}} \cup \boldsymbol{C}_k, \quad \text{if} \quad \varepsilon_t \geq \mu_0, \end{aligned} \tag{5.21}$$

where $\boldsymbol{C}_{\mathfrak{D}}$ is the dictionary obtained through **Step 2**, $\boldsymbol{C}_k$ is a new sample in the remaining training dataset. According to the criterion introduced in Section 5.1.1, if the new sample $\boldsymbol{C}_k$ satisfies the dictionary updated condition, it will be included into the dictionary $\boldsymbol{C}_{\mathfrak{D}}$.

Step 4: Based on the dictionary and the classifier obtained from the training frames, the incoming frame sample $\boldsymbol{C}_{n+l}$ is classified. The workflow of **Strategy 1** is shown in Fig.5.2, and described by the following equation:

$$\begin{aligned} &f(\boldsymbol{C}_{n+l}) \\ &= \mathbf{sgn}(R^2 - \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa(\boldsymbol{C}_i, \boldsymbol{C}_j) + 2 \sum_{i} \alpha_i \kappa(\boldsymbol{C}_i, \boldsymbol{C}_{n+l}) - \kappa(\boldsymbol{C}_{n+l}, \boldsymbol{C}_{n+l})) \\ &= \begin{cases} 1 & f(\boldsymbol{C}_{n+l}) \geq 0 \\ -1 & f(\boldsymbol{C}_{n+l}) < 0. \end{cases} \end{aligned}$$

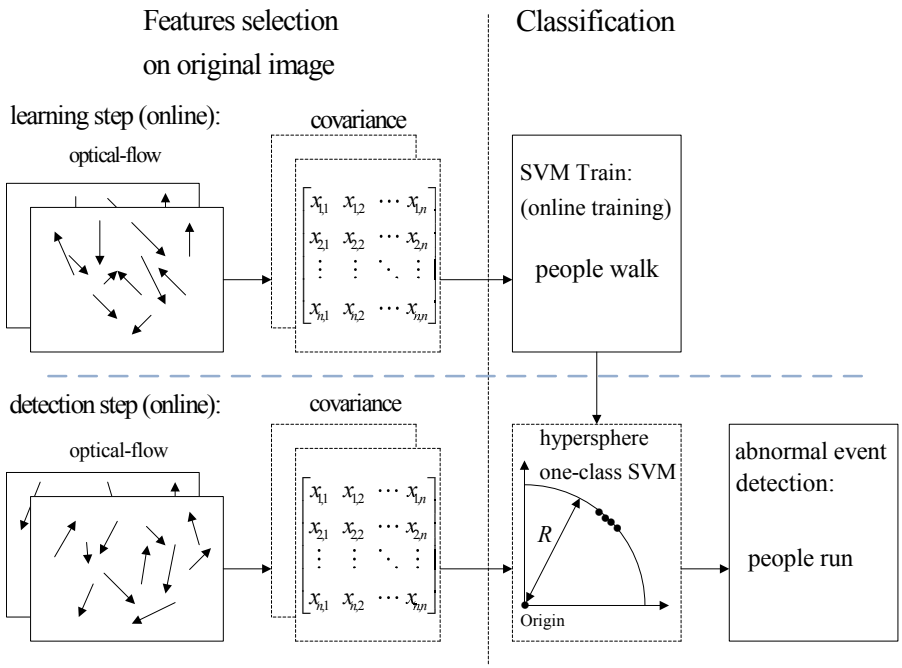


Figure 5.2: Major processing states of the proposed online support vector data description (SVDD) abnormal *frame* event detection method. The *frame* COV descriptor is computed.

5.1.2.2 Strategy 2

In this strategy, the dictionary is updated through both training and testing periods. The feature extraction step (**Step 1**) and the online training steps (**Step 2**, **Step 3**) are as the same as the ones presented in **Strategy 1**. The testing step is different. The new coming datum which is detected as normal, but satisfies dictionary update condition should be included into $\boldsymbol{C}_{\mathfrak{D}}$. The dictionary is needed to be updated through the testing period to include new samples.

Step 4-Strategy 2: If the incoming frame sample $\boldsymbol{C}_{n+l}$ is classified as normal ($f(\boldsymbol{C}_{n+l}) = 1$), the data is checked by the criterion described in Section 5.1.1. When the data satisfies the dictionary update criterion, this testing sample will be included into the dictionary. This step can be generalized by the following equation:

$$
\begin{aligned}
& f(\boldsymbol{C}_{n+l}) \\
&= \mathbf{sgn}(R^2 - \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa(\boldsymbol{C}_i, \boldsymbol{C}_j) + 2\sum_{i} \alpha_i \kappa(\boldsymbol{C}_i, \boldsymbol{C}_{n+l}) - \kappa(\boldsymbol{C}_{n+l}, \boldsymbol{C}_{n+l})) \\
&= \begin{cases} 1 & f(\boldsymbol{C}_{n+l}) \geq 0 \begin{cases} \varepsilon_t \geq \mu_0 \rightarrow \boldsymbol{C}_{\mathfrak{D}} := \boldsymbol{C}_{\mathfrak{D}} \cup \boldsymbol{C}_{n+l} \\ \varepsilon_t < \mu_0 \rightarrow \boldsymbol{C}_{\mathfrak{D}} := \boldsymbol{C}_{\mathfrak{D}} \end{cases} \\ -1 & f(\boldsymbol{C}_{n+l}) < 0. \end{cases}
\end{aligned}
$$

5.1.3 Abnormal Detection Results

This section presents the results of experiments conducted to analyze the performance of the proposed method. A competitive performance through both **Strategy 1** and **Strategy 2** of UMN [UMN 2006] dataset is presented. The normal samples for training or for normal testing are the frames where the people are walking in different directions. The samples for abnormal testing are the frames where people are running.

5.1.3.1 Abnormal Visual Events Detection–Strategy 1

The results of the proposed abnormal events detection method via **Strategy 1** online one-class SVM of UMN [UMN 2006] dataset are shown below.

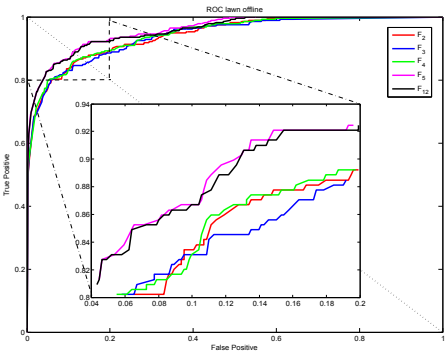
The detection results of lawn scene, indoor scene and plaza scene are shown in Fig.5.3, Fig.5.4 and Fig.5.5 respectively. Gaussian kernel for the Lie Group is used in these three scenes. Different value of σ and penalty factor C are chosen, the area under the ROC curve is shown as a function of these parameters [Hanley 1982]. The results show that taking covariance matrix as descriptor can obtain satisfactory performance for abnormal detection. And also, training the samples online can obtain similarly detection performance as training all the samples offline. Online one-class SVM is appropriate to detect abnormal visual events. There are 1431 frames in the lawn scene, 480 normal frames are used for training. In the offline strategy, all the 480 frames covariance matrices should be saved in the memory. In **Strategy 1**, 100 frames covariance matrices are considered as the dictionary firstly. When F_5-17×17 feature is adopted to construct the covariance descriptor, the variance of Gaussian kernel is $\sigma = 1$, the preset threshold of the criterion is $\mu_0 = 0.5$, the dictionary size increases from 100 to 101, the maximum accuracy of the detection results is 91.69%. In the indoor scene, there are 2975 normal frames and 1057 abnormal frames. In the plaza scene, there are 1831 normal frames and 286 abnormal frames. The processes of the experiments are similar to the ones of the lawn scene. When feature vector is F_5-17×17, $\sigma = 1$, $\mu_0 = 0.5$, the dictionary size of theses two scenes remain 100. The online strategy keeps the memory size almost unchanged when the size of training dataset increases.

5.1.3.2 Abnormal frame events detection–Strategy 2

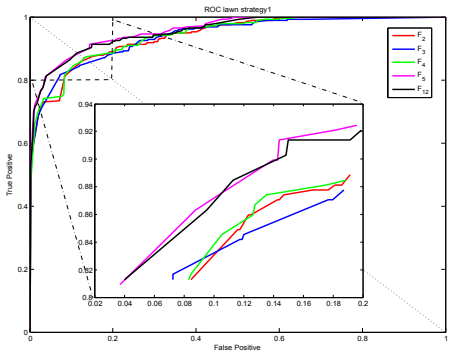
The results of the abnormal event detection method via **Strategy 2** of UMN dataset are shown as follows. In the experiment process of the lawn scene, 100 normal samples from the training samples are learnt firstly, and then other 380 training data are learnt online one-by-one. After these two training steps, we can obtain the basic dictionary from the training samples, and also the classifier. In the following testing step, the dictionary is updated if the sample satisfies the dictionary update criterion. When a new sample is coming, it is firstly detected by the previous classifier. If it is classified as anomaly, the dictionary and the classifier are not changed. Otherwise, if the sample is classified as a normal one, the sparse criterion introduced in Section 5.1.1 is used to check the correlation between the earlier dictionary and this new datum. It will be included into the dictionary when it satisfied the update condition. The dictionary will be updated through the whole testing period. The other two scenes, the indoor and plaza scene are handled by the same methods.

(a) Normal lawn scene

(b) Abnormal lawn scene

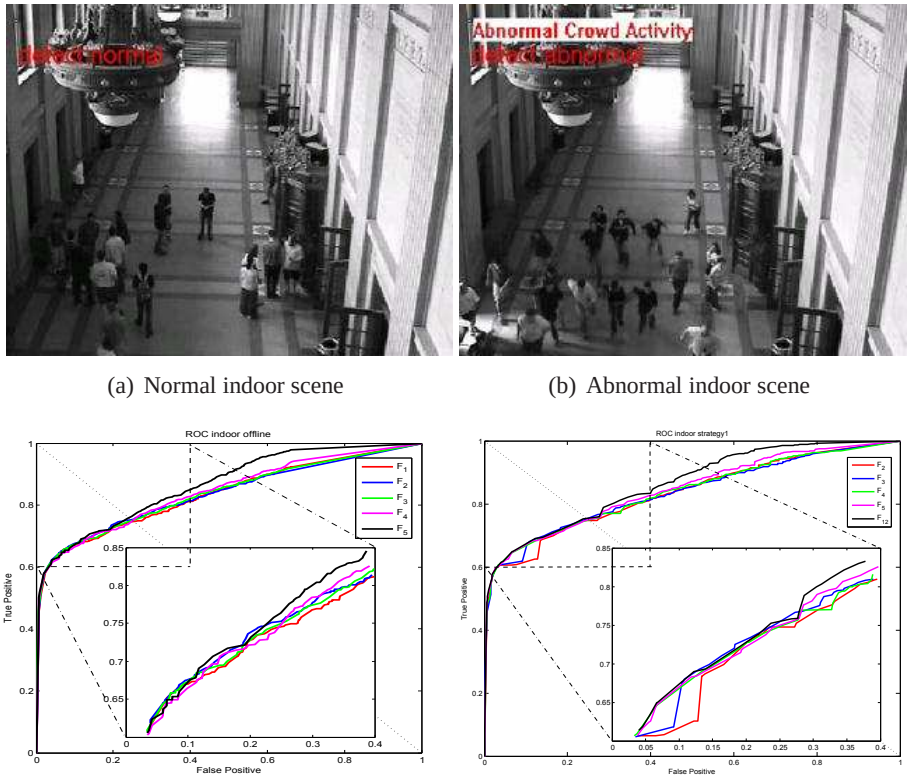


(c) ROC train offline



(d) ROC Strategy 1

Figure 5.3: Abnormal *frame* event detection results of the lawn scene based on *frame* covariance matrix descriptor via online support vector data description (online SVDD) **Strategy 1**. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) ROC curve of different features F of the lawn scene results via one-class SVM. All the training samples are learned together offline. The biggest AUC value is 0.9591. (d) ROC curve of different features F results via **Strategy 1** online one-class SVM. The biggest AUC value is 0.9581.



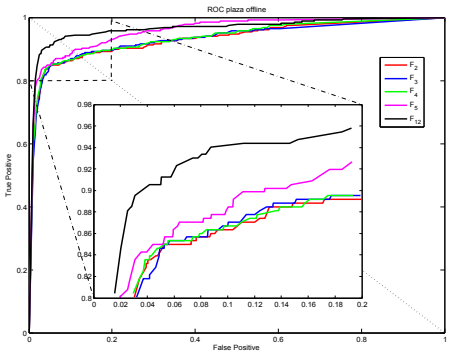
(a) Normal indoor scene (b) Abnormal indoor scene

(c) ROC train offline (d) ROC Strategy 1

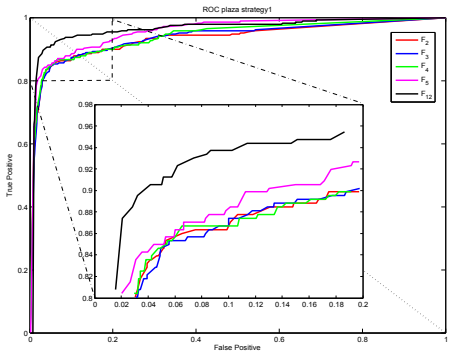
Figure 5.4: Abnormal *frame* event detection results of the indoor scene based on *frame* covariance matrix (COV) descriptor via online support vector data description (online SVDD) **Strategy 1**. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) ROC curve of different features F of the lawn scene results via one-class SVM. All the training samples are learned together offline. The biggest AUC value is 0.8649. (d) ROC curve of different features F results via **Strategy 1** online one-class SVM. The biggest AUC value is 0.8628.

(a) Normal plaza scene

(b) Abnormal plaza scene



(c) ROC train offline



(d) ROC Strategy 1

Figure 5.5: Abnormal *frame* event detection results of the plaza scene based on *frame* covariance matrix (COV) descriptor via online support vector data description (online SVDD) **Strategy 1**. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) ROC curve of different features F of the plaza scene results via one-class SVM. All the training samples are learned together offline. The biggest AUC value is 0.9649. (d) ROC curve of different features F results via **Strategy 1** online one-class SVM. The biggest AUC value is 0.9632.

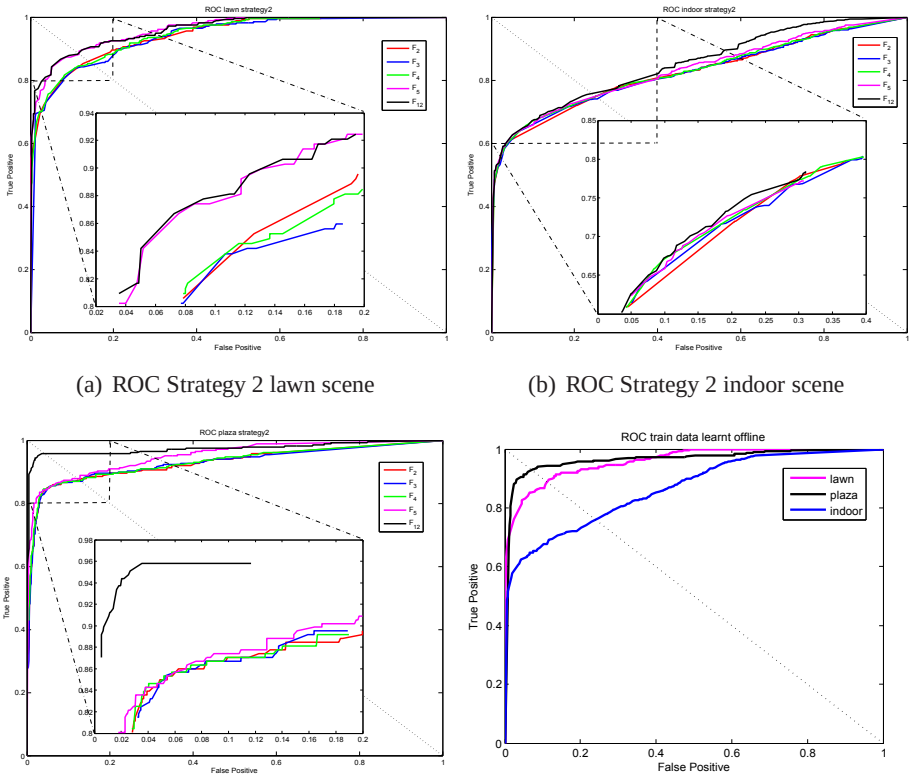
Features	Area under ROC		
	lawn	indoor	plaza
training samples are learned offline			
$F_2(6\times 6du)$	0.9426	0.8351	0.9323
$F_3(6\times 6dv)$	0.9400	0.8358	0.9321
$F_4(8\times 8)$	0.9440	0.8375	0.9359
$F_5(12\times 12)$	**0.9591**	0.8440	0.9580
$F_{12}(17\times 17)$	0.9567	**0.8649**	**0.9649**
Strategy 1			
$F_2(6\times 6du)$	0.9399	0.8328	0.9343
$F_3(6\times 6dv)$	0.9390	0.8355	0.9366
$F_4(8\times 8)$	0.9418	0.8377	0.9411
$F_5(12\times 12)$	**0.9581**	0.8457	0.9573
$F_{12}(17\times 17)$	0.9551	**0.8628**	**0.9632**
Strategy 2			
$F_2(6\times 6du)$	0.9427	0.8237	0.9288
$F_3(6\times 6dv)$	0.9370	0.8241	0.9283
$F_4(8\times 8)$	0.9430	0.8274	0.9312
$F_5(12\times 12)$	**0.9605**	0.8331	0.9505
$F_{12}(17\times 17)$	0.9601	**0.8495**	**0.9746**

Table 5.1: AUC of abnormal *frame* events detection results based on *frame* COV descriptor constructed by different features F via original support vector data description (SVDD), **Strategy 1** online hypersphere one-class SVM, and **Strategy 2** online hypersphere one-class SVM of UMN dataset. The biggest value of each method is shown in bold.

When F_5-17×17 feature is adopted, the variance of the Gaussian kernel is $\sigma = 1$, and the preset threshold of the criterion is $\mu_0 = 0.5$, the dictionary size of the lawn, indoor and plaza scene are increased from 100 to 106, 102 and 102, respectively. The ROC curve of detection results of these three scenes are shown in Fig. 5.6(a), (b) and (c). Besides the merit of saving memory of **Strategy 1**, **Strategy 2** also has the advantage of adaptation to the long duration sequence.

The results performances of offline strategy, **Strategy 1** and **Strategy 2** are shown in TABLE 5.1. The performances of these two strategies results are similar to that of the results when all training samples are learnt together. When $F_4(12\times 12)$ or $F_5(17\times 17)$ are chosen as the features to form covariance matrix descriptor, the results have the best performance. These two features are more abundant to include movement and intensity information.

The result performances of the covariance matrix descriptor based online one-class SVM method and the state-of-the-art methods are shown in TABLE 5.2. The covariance matrix based online abnormal frame detection method obtains competitive performance. In generally, our method is better then others except sparse reconstruction cost (SRC)



(a) ROC Strategy 2 lawn scene

(b) ROC Strategy 2 indoor scene

(c) ROC Strategy 2 plaza scene

(d) ROC train all data

Figure 5.6: ROC curve of abnormal *frame* events detection results of the lawn, indoor, and plaza scenes based on *frame* COV descriptor via online support vector data description (online SVDD) **Strategy 2**. (a) ROC curve of different features F results via **Strategy 2** of lawn scene. The biggest AUC value is 0.9605. (b) **Strategy 2** results of indoor scene. The biggest AUC value is 0.8495. (c) **Strategy 2** results of plaza scene. The biggest AUC value is 0.9746. (d) The ROC curve of best performance of lawn, indoor and plaza scene when the training samples are learnt offline. The biggest AUC value of lawn, indoor and plaza are 0.9591, 0.8649 and 0.9649.

Method	Area under ROC		
	lawn	indoor	plaza
Social Force [Mehran 2009]	0.96		
Optical Flow [Mehran 2009]	0.84		
NN [Cong 2011]	0.93		
SRC [Cong 2011]	0.995	0.975	0.964
STCOG [Shi 2010]	0.9362	0.7759	0.9661
COV online (Ours)	0.9605	0.8628	0.9746

Table 5.2: The comparison of our proposed *frame* covariance matrix descriptor and online support vector data description (online SVDD) based method with the state-of-the-art methods for abnormal *frame* event detection of UMN dataset.

[Cong 2011] in lawn scene and indoor scene. In that paper, multi-scale HOF is taken as a feature, and a testing sample is classified by its sparse reconstructor cost, through a weighted linear reconstruction of the over-complete normal basis set. But computation of the HOF might take more time than calculating covariance. By adopting the integral image [Tuzel 2006], the covariance matrix descriptor of the subimage can be computed conveniently. So the covariance descriptor can be appropriately used to analyze the partial movement.

5.2 Abnormal detection via online least squares one-class SVM

In this section, we propose a novel online classification method, namely online least squares one-class support vector machines (online LS-OC-SVM). The LS-OC-SVM extracts a hyperplane as an optimal description of training objects in a regularized least squares sense. The online LS-OC-SVM firstly learns from a training set with a limited number of samples to provide a basic normal model, and then updates the model through remaining data. In the sparse online scheme, the model complexity is controlled by the coherence criterion. And then, the online LS-OC-SVM is adopted to handle the abnormal event detection problem.

5.2.1 Least squares one-class support vector machines

Least squares SVM (LS-SVM) was proposed by Suykens in [Suykens 1999, Suykens 2002]. By using the quadratic loss function, Choi proposed least squares one-class SVM (LS-OC-SVM) [Choi 2009]. LS-OC-SVM extracts a hyperplane as an optimal description of training objects in a regularized least squares sense. It can be written as the following objective function:

$$\begin{aligned} &\min_{\boldsymbol{w},\xi,\rho} \ \frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + \frac{1}{2}C\sum_{i=1}^{n}\xi_i^2 \\ &\text{subject to: } \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle = \rho - \xi_i. \end{aligned} \tag{5.22}$$

The condition for the slack variables in OC-SVM, $\xi_i \geq 0$, is no longer in need. The variable, ξ_i, represents an error caused by a training object, $\boldsymbol{x}_i$, with respect to the hyperplane. The definitions of the other parameters in eq.(5.22) are the same as the ones in OC-SVM. The associated Lagrange is:

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}^{n}\alpha_i\left(\boldsymbol{w}^\top\Phi(\boldsymbol{x}_i) - \rho + \xi_i\right). \tag{5.23}$$

Setting derivatives of eq.(5.23) with respective to primal variables, $\boldsymbol{w}$, ξ_i, ρ and α_i, to zero, we have the following stationarity conditions:

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \qquad \Rightarrow \boldsymbol{w} = \sum_{i=1}^{n}\alpha_i\Phi(\boldsymbol{x}_i), \tag{5.24}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \qquad \Rightarrow C\xi_i = \alpha_i, \tag{5.25}$$

$$\frac{\partial L}{\partial \rho} = 0 \qquad \Rightarrow \sum_{i=1}^{n}\alpha_i = 1, \tag{5.26}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \qquad \Rightarrow \boldsymbol{w}^\top\Phi(\boldsymbol{x}_i) + \xi_i - \rho = 0. \tag{5.27}$$

Substituting eq.(5.24)–(5.26) into (5.27) yields:

$$\sum_{i,j=1}^{n}\alpha_i\Phi^\top(\boldsymbol{x}_i)\Phi(\boldsymbol{x}_j) + \frac{\alpha_i}{C} - \rho = 0. \tag{5.28}$$

For all $i = 1, 2, \ldots, n$, we can rewrite eq.(5.28) in matrix form as:

$$\begin{bmatrix} \boldsymbol{K} + \frac{\boldsymbol{I}}{C} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ -\rho \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \tag{5.29}$$

where $\boldsymbol{K}$ is the Gram matrix with (i, j)-th entry $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $\boldsymbol{I}$ is the identity matrix with the same dimension as Gram matrix $\boldsymbol{K}$ and $\boldsymbol{\alpha}$ is the column vector with i-th entry α_i for training sample $\boldsymbol{x}_i$. $\mathbf{1}$ and $\mathbf{0}$ are all-one and all-zero column vectors, respectively, with compatible lengths. The parameters, $\boldsymbol{\alpha}$ and ρ, could be obtained by:

$$\begin{bmatrix} \boldsymbol{\alpha} \\ -\rho \end{bmatrix} = \begin{bmatrix} \boldsymbol{K} + \frac{\boldsymbol{I}}{C} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}^{-1}\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}. \tag{5.30}$$

The hyperplane is then described by:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n}\alpha_i\kappa(\boldsymbol{x_i}, \boldsymbol{x}) - \rho = 0. \tag{5.31}$$

The distance, $dis(\boldsymbol{x})$, of a datum, $\boldsymbol{x}$, with respect to the hyperplane is calculated by:

$$dis(\boldsymbol{x}) = \frac{|f(\boldsymbol{x})|}{\|\boldsymbol{\alpha}\|} = \frac{|(\sum_{i=1}^{n}\alpha_i\kappa(\boldsymbol{x}_i, \boldsymbol{x}) - \rho)|}{\|\boldsymbol{\alpha}\|}, \tag{5.32}$$

where $\boldsymbol{x}_i$ is a training sample, $\|\boldsymbol{\alpha}\|$ is the two-norm of vector $\boldsymbol{\alpha}$. An object with a low $dis(\boldsymbol{x})$ value lies close to the hyperplane thus resembles the training set better than other objects with high $dix(\boldsymbol{x})$ values. The distance, $dis(\boldsymbol{x})$, is used as a proximity measure to determine the normal and abnormal class of the data [Choi 2009].

5.2.2 Online least squares one-class support vector machines

In an online learning scheme, the training data continuously arrive. We thus need to tune hyperparameters in the objective function and the hypothesis class in an online manner [Diehl 2003]. Let $\boldsymbol{\alpha}_n$, $\boldsymbol{K}_n$ and $\boldsymbol{I}_n$ denote the coefficient, Gram matrix and identity matrix at the time step, n, respectively. The parameters of LS-OC-SVM $[\boldsymbol{\alpha}_n \ -\rho_n]^\top$ at the time step, n, could be calculated as:

$$\begin{bmatrix} \boldsymbol{\alpha}_n \\ -\rho_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C} & \mathbf{1}_n \\ \mathbf{1}_n^\top & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0}_n \\ 1 \end{bmatrix}. \tag{5.33}$$

In order to proceed, recall the matrix inverse identity for matrices A, B, C and D with suitable sizes [Honeine 2012]:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -A^{-1}B \\ 1 \end{bmatrix} \times (D - CA^{-1}B)^{-1} \times [-CA^{-1} \quad 1]. \tag{5.34}$$

The matrix, $\boldsymbol{K}_n$, with diagonal loading $\frac{\boldsymbol{I}_n}{C}$ can be calculated recursively with respect to time step n by:

$$\left[\boldsymbol{K}_{n+1} + \frac{\boldsymbol{I}_{n+1}}{C}\right]^{-1} \tag{5.35}$$

$$= \left[\begin{array}{c|c} \boldsymbol{K}_n + \frac{\boldsymbol{I}}{C} & \boldsymbol{\kappa}_{n+1} \\ \hline \boldsymbol{\kappa}_{n+1} & \kappa_{n+1} + \frac{1}{C} \end{array}\right]^{-1} \tag{5.36}$$

$$= \begin{bmatrix} \left(\boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C}\right)^{-1} & \mathbf{0}_n \\ \mathbf{0}_n^\top & 0 \end{bmatrix} + \frac{1}{\left(\kappa_{n+1} + \frac{1}{C}\right) - \boldsymbol{\kappa}_{n+1}\left(\boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C}\right)^{-1}\boldsymbol{\kappa}_{n+1}}$$

$$\begin{bmatrix} -\left(\boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C}\right)^{-1}\boldsymbol{\kappa}_{n+1} \\ 1 \end{bmatrix} \begin{bmatrix} -\boldsymbol{\kappa}_{n+1}^\top \left(\boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C}\right)^{-1} & 1 \end{bmatrix}. \tag{5.37}$$

where $\boldsymbol{\kappa}_{n+1}$ is the column vector with i-th entry $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{n+1})$, $i \in \{1, 2, \ldots, n\}$, and $\kappa_{n+1} = \kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1})$. Based on eq.(5.33) and (5.35), we arrive at an online implementation of LS-OC-SVM.

5.2.3 Sparse online least squares one-class support vector machines

The procedures for calculating the parameters, $\boldsymbol{\alpha}$ and ρ, of LS-OC-SVM in Section 5.2.2 lose sparseness, due to the quadratic loss function in the objective function eq.(5.22). This formulation is inappropriate for large-scale data and unsuitable for online learning, as the

number of training samples grows infinitely [Noumir 2012c]. We propose a sparse solution to provide a robust formulation. A dictionary is adopted to address the sparse approximation problem [Tropp 2004].

Instead of eq.(5.24), where $\boldsymbol{w}$ is expressed with all available data, we intend to approximate it by adopting a dictionary in a sparse way. Consider a dictionary, $\boldsymbol{x}_{\mathfrak{D}}$, $\mathfrak{D} \subset \{1, 2, \ldots, n\}$, of size D with elements $\boldsymbol{x}_{w_j}$, $j \in \mathfrak{D}$. Instead of eq.(5.24), we approximate $\boldsymbol{w}$ with these D dictionary elements:

$$\boldsymbol{w} = \sum_{j=1}^{D} \beta_j \Phi(\boldsymbol{x}_{w_j}). \tag{5.38}$$

The hyperplane becomes:

$$f(\boldsymbol{x}) = \sum_{j=1}^{D} \beta_j \kappa(\boldsymbol{x}, \boldsymbol{x}_{w_j}) - \rho = 0. \tag{5.39}$$

In sparse online LS-OC-SVM, the distance, $dis_{\mathfrak{D}}(\boldsymbol{x})$, of a datum, $\boldsymbol{x}$, to the hyperplane is:

$$dis_{\mathfrak{D}}(\boldsymbol{x}) = \frac{|\sum_{j=1}^{D} \beta_i \kappa(\boldsymbol{x}, \boldsymbol{x}_{w_j}) - \rho|}{\|\boldsymbol{\beta}\|}, \tag{5.40}$$

where $\boldsymbol{x}_{w_j}$ is a dictionary element and $\boldsymbol{\beta}$ is the column vector with the entries, β_j. Replacing eq.(5.38) into Lagrange Function (5.23), we have:

$$L = \frac{1}{2}\boldsymbol{\beta}^\top K_{\mathfrak{D}} \boldsymbol{\beta} - \rho + \frac{C}{2}\sum_{i=1}^{n} \xi_i^2 - \sum_{i=1}^{n} \alpha_i (\sum_{j=1}^{D} \beta_j \Phi^\top(\boldsymbol{x}_{w_j}) \Phi(\boldsymbol{x}_i) + \xi_i - \rho). \tag{5.41}$$

Taking the derivatives of the Function (5.41) with respect to primal variables, $\boldsymbol{\beta}$, ξ_i, ρ and α_i, yields:

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = 0 \qquad \Rightarrow K_D \boldsymbol{\beta} = K_D^\top(\boldsymbol{x})\boldsymbol{\alpha}, \tag{5.42}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \qquad \Rightarrow C\xi_i = \alpha_i, \tag{5.43}$$

$$\frac{\partial L}{\partial \rho} = 0 \qquad \Rightarrow \sum_{i=1}^{n} \alpha_i = 1, \tag{5.44}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \qquad \Rightarrow \sum_{j=1}^{D} \beta_j \kappa(\boldsymbol{x}_{w_j}, \boldsymbol{x}_i) + \xi_i - \rho = 0. \tag{5.45}$$

The matrix form for Condition (5.45) is written:

$$\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{\beta} + \boldsymbol{\xi} - \rho = 0. \tag{5.46}$$

Replacing Conditions (5.42) and (5.43) into (5.46) leads to:

$$\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x})\boldsymbol{\alpha} + \frac{\boldsymbol{\alpha}}{C} - \rho = 0. \tag{5.47}$$

Combining Equations (5.44) and (5.47), the equation for computing coefficients $\begin{bmatrix}\boldsymbol{\alpha} & -\rho\end{bmatrix}^{\top}$ becomes:

$$\begin{bmatrix}\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) + \frac{\boldsymbol{I}}{C} & \mathbf{1} \\ \mathbf{1}^{\top} & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{\alpha} \\ -\rho\end{bmatrix} = \begin{bmatrix}\mathbf{0} \\ 1\end{bmatrix}. \tag{5.48}$$

After providing these relations with the dictionary, we now discuss the dictionary construction. The coherence criterion is adopted to characterize a dictionary in sparse approximation problems. It provides an elegant model reduction criterion with a less computationally-demanding procedure [Noumir 2012c, Tropp 2004, Richard 2009]. The coherence of a dictionary is defined as the largest correlation between the elements in the dictionary, *i.e.*,

$$\mu = \max_{i,j\in\mathfrak{D},i\neq j} |\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)|. \tag{5.49}$$

In the online case, the coherence between a new datum and the current dictionary is calculated by:

$$\varepsilon_t = \max_{j\in\mathfrak{D}} |\kappa(\boldsymbol{x}_t, \boldsymbol{x}_{w_j})|, \tag{5.50}$$

where $\boldsymbol{x}_{w_j}$ is the element in the dictionary, $\boldsymbol{x}_{\mathfrak{D}}$. Presetting a threshold, μ_0, the new arrival sample, $\boldsymbol{x}_t$, at the time step, t, is tested with the coherence criterion to judge whether the dictionary remains unchanged or is incremented by including the new element. For n training samples, the subset, which includes m ($1 \leq m \ll n$) samples, is considered the initial dictionary. Then, each remaining sample is tested with eq.(5.50) to determine the relation between itself and the previous dictionary. If $\varepsilon_t \leq \mu_0$, it will be included into the dictionary. Concretely, the algorithm is performed with two cases described herein below.

First case: $\varepsilon_t > \mu_0$

In this case, at time step $n+1$, the new data, $\boldsymbol{x}_{n+1}$, is not included into the dictionary. The Gram matrix, $\boldsymbol{K}_{\mathfrak{D}}$, with the entries, $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i, j \in \{1, 2, \ldots, D\}$, is unchanged. When a new sample, $\boldsymbol{x}$, arrives, we need to compute:

$$\left[\begin{bmatrix}\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x}) \\ \boldsymbol{\kappa}^{\top}\end{bmatrix}\boldsymbol{K}_{\mathfrak{D}}^{-1}\begin{bmatrix}\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})^{\top} & \boldsymbol{\kappa}\end{bmatrix} + \frac{\boldsymbol{I}}{C}\right]^{-1} = \begin{bmatrix}\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) + \frac{\boldsymbol{I}}{C} & \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{\kappa} \\ \boldsymbol{\kappa}^{\top}\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) & \boldsymbol{\kappa}^{\top}\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{\kappa} + \frac{\boldsymbol{I}}{C}\end{bmatrix}^{-1}, \tag{5.51}$$

where at time step $n+1$, $\boldsymbol{\kappa}$ is the column vector with entries $\kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{w_j})$, $j \in \{1, 2, \ldots, D\}$. $\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})$ is the matrix with the (i, j)-th entry $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{w_j})$, $i \in \{1, 2, \ldots, n\}$, $j \in \{1, 2, \ldots, D\}$.

Second case: $\varepsilon_t \leq \mu_0$

In this case, the new data, $\boldsymbol{x}_{n+1}$, is added into the dictionary, $\boldsymbol{x}_{\mathfrak{D}}$. Then, the Gram matrix should be changed by:

$$\overline{\boldsymbol{K}}_{\mathfrak{D}} = \begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}} & \boldsymbol{d} \\ \boldsymbol{d}^{\top} & d \end{bmatrix}, \tag{5.52}$$

where $\overline{\boldsymbol{K}}_{\mathfrak{D}}$ is the Gram matrix of the dictionary, including the new arrival dictionary sample, $\boldsymbol{x}_{n+1}$, and $\boldsymbol{K}_{\mathfrak{D}}$ is the Gram matrix of the dictionary at the last time step, n. Let $\boldsymbol{x}_{\mathfrak{D}} = \{\boldsymbol{x}_{w_1}, \boldsymbol{x}_{w_2}, \ldots, \boldsymbol{x}_{w_D}\}$ denote the dictionary at time step n; $\boldsymbol{d}$ is the column vector with entries $d_j = \kappa(\boldsymbol{x}, \boldsymbol{x}_{w_j})$, $j \in \{1, 2, \ldots, D\}$, and $d = \kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1})$.

By adopting the matrix inverse identity eq.(5.34), we have:

$$\overline{\boldsymbol{K}}_{\mathfrak{D}}^{-1} = \begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}}^{-1} + \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^{\top} & c \end{bmatrix}, \tag{5.53}$$

where:

$$c = \frac{1}{d - \boldsymbol{d}^{\top}\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{d}}, \tag{5.54}$$

$$\boldsymbol{A} = c\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{d}\boldsymbol{d}^{\top}\boldsymbol{K}_{\mathfrak{D}}^{-1}, \tag{5.55}$$

$$\boldsymbol{b} = -c\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{d}. \tag{5.56}$$

Because the dictionary changes, the value of $\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})$ and also $\left[\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) + \frac{\boldsymbol{I}}{C}\right]^{-1}$ should be updated. Let the S denote the updated $\left[\boldsymbol{K}_{D}(\boldsymbol{x})\boldsymbol{K}_{D}^{-1}\boldsymbol{K}_{\mathcal{D}}^{\top}(\boldsymbol{x}) + \frac{\boldsymbol{I}}{C}\right]^{-1}$ at time step $n+1$; we have:

$$S = \left[\begin{bmatrix}\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x}) & \boldsymbol{q}\end{bmatrix} \overline{\boldsymbol{K}}_{\mathfrak{D}}^{-1} \begin{bmatrix}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) \\ \boldsymbol{q}^{\top}\end{bmatrix} + \frac{\boldsymbol{I}}{C}\right]^{-1} \tag{5.57}$$

$$\begin{aligned} =&[\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})^{\top} + \frac{\boldsymbol{I}}{C} + \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{A}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) + \\ &\boldsymbol{q}\boldsymbol{b}^{\top}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) + \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{b}\boldsymbol{q}^{\top} + c\boldsymbol{q}\boldsymbol{q}^{\top}]^{-1}. \end{aligned} \tag{5.58}$$

where at time step $n+1$, $\boldsymbol{q}$ is the column vector with entries $q_i = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_{D+1})$, $i \in \{1, 2, \ldots, n\}$, and $\boldsymbol{x}_{D+1}$ is the new arrival datum $\boldsymbol{x}_{n+1}$, which is included into the dictionary. The matrix inverse in eq.(5.57) can be calculated by using four-times Woodbury identity:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}, \tag{5.59}$$

with proper choices of matrices A, U, C and V, such that U and V should be chosen as two vectors, and A should be chosen as a scaler. Thus, the inverse, $(C^{-1} + VA^{-1}U)$, is a scaler; eq. (5.57) can be calculated very efficiently. For instance, for computing the inverse, including the term, $(\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{b}\boldsymbol{q}^{\top})$, we regard two vectors, $(\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{b})$ and $\boldsymbol{q}^{\top}$, as vector U and V, respectively, while C in Equation (5.59) is one.

Once knowing S, using eq.(5.51) to add the new $\boldsymbol{\kappa}$ with entries $\kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{w_j})$, $j \in \{1, 2, \ldots, D, D+1\}$, $\boldsymbol{x}_{w_j}$ is an element of the dictionary.

5.2.4 Abnormal Event Detection detection method

In an abnormal event detection problem, it is assumed that a set of training frames, $\{I_1, I_2, \ldots, I_n\}$ (the positive class), describing the normal behavior is obtained. The abnormal detection strategies relative to the online algorithms proposed in Section 5.2.2 and Section 5.2.3 are introduced below.

5.2.4.1 Online LS-OC-SVM Strategy

The general architecture of the abnormal event detection method via online least squares one-class SVM (online LS-OC-SVM) proposed in Section 5.2.2 is summarized in Algorithm 2; the flowchart is shown in Fig. 5.7 and explained below.

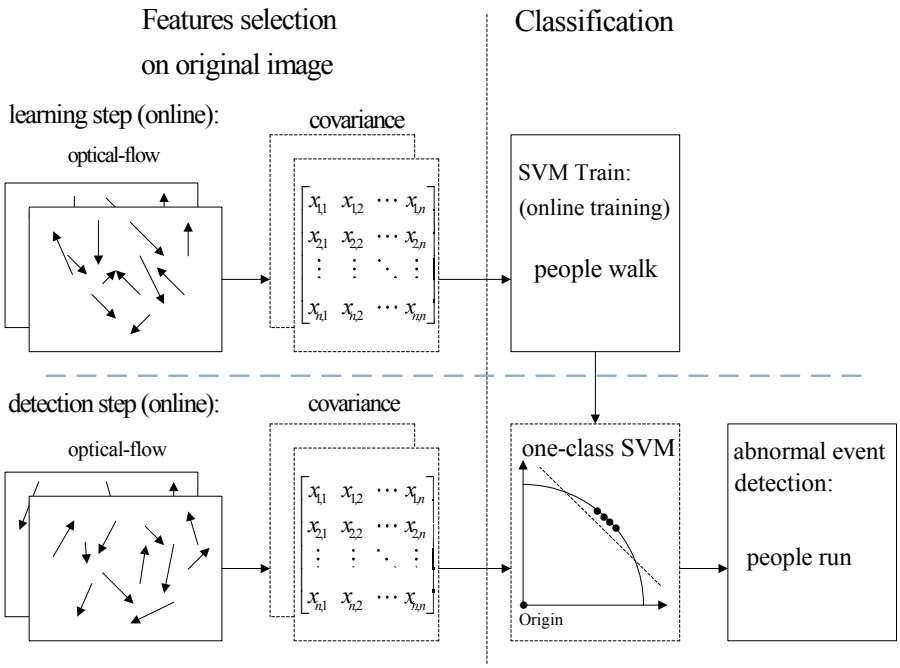


Figure 5.7: Major processing states of the proposed abnormal frame event detection method based on *frame* covariance matrix descriptor via one-class SVM.

The feature descriptor computation processes are the same as before. The training and testing processes of LS-OC-SVM are explained bellow. The two strategies proposed in Section 5.1.2.1 and Section 5.1.2.2 are also suitable in online OC-LS-SVM and sparse online OC-LS-SVM algorithms. In this section, we introduce the learning processes on the training samples.

Step 1: The first step consists of calculating the covariance matrix descriptor of the training frames. This step can be generalized as:

$$\{OP_1, OP_2, \ldots, OP_n\} \longrightarrow \{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_n\}, \tag{5.65}$$

Algorithm 2 Visual abnormal event detection via online least squares one-class support vector machine (LS-OC-SVM) and sparse online LS-OC-SVM.

Require

n training frames $\{I_i\}_{i=1}^{n}$ and the corresponding optical flow $\{OP_i\}_{i=1}^{n}$.
Compute the covariance matrix of each frame.

$$\{OP_1, OP_2, \ldots, OP_n\} \longrightarrow \{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_n\} \tag{5.60}$$

(a) *Online strategy:* Applying LS-OC-SVM on the small subset of training samples to calculate the coefficient matrix.

$$\{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_m\}, 1 \leq m \ll n \xrightarrow{online} \text{coefficient matrix } \Big[\boldsymbol{K}\Big] \ \Big[\boldsymbol{\alpha} \ \ -\rho\Big]^{\top} \tag{5.61}$$

(b) *Sparse online strategy:* Applying LS-OC-SVM to train the initial dictionary, $C_{\mathfrak{D}}$, offline.

$$\boldsymbol{C}_{\mathfrak{D}} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_m\}, 1 \leq m \ll n \xrightarrow{offline} \text{coefficient matrix } \Big[\boldsymbol{K}\Big] \ \Big[\boldsymbol{\beta} \ \ -\rho\Big]^{\top} \tag{5.62}$$

(a) *Online strategy:* Applying online LS-OC-SVM on the remaining samples to calculate the coefficient matrix.

$$\{\boldsymbol{C}_{m+1}, \boldsymbol{C}_{m+2} \ldots \boldsymbol{C}_n\}, \Big[\boldsymbol{K}\Big] \xrightarrow{online} \text{coefficient matrix } \Big[\boldsymbol{K}\Big] \ \Big[\boldsymbol{\alpha} \ \ -\rho\Big]^{\top} \tag{5.63}$$

(b) *Sparse online strategy:* Applying sparse online LS-OC-SVM on the remaining samples to calculate the coefficient matrix and to update the dictionary.

$$\begin{aligned} &\{\boldsymbol{C}_{\mathfrak{D}}, \boldsymbol{C}_k\}, m < k \leq n \xrightarrow{sparse\ online} \text{coefficient matrix } \Big[\boldsymbol{\beta} \ \ -\rho\Big]^{\top}, \\ &\begin{cases} \boldsymbol{C}_{\mathfrak{D}} := \boldsymbol{C}_{\mathfrak{D}} \cup \boldsymbol{C}_k, & \text{if} \quad \varepsilon_t \geq \mu_0, \\ \boldsymbol{C}_{\mathfrak{D}} := \boldsymbol{C}_{\mathfrak{D}}, & \text{if} \quad \varepsilon_t < \mu_0. \end{cases} \end{aligned} \tag{5.64}$$

Each frame $\boldsymbol{C}_{n+l}$ is classified via LS-OC-SVM.

where $\{OP_1, OP_2, \ldots, OP_n\}$ are the image optical flows of the $1st$ to $n{-}th$ frames; $\{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_n\}$ are the covariance matrix descriptors.

Step 2: The second step is applying LS-OC-SVM on a small subset of the training samples to calculate the coefficient parameters, $\boldsymbol{\alpha}$ and ρ, in eq. (5.29). Consider a subset $\{\boldsymbol{C}_i\}_{i=1}^{m}$, $1 \leq m \ll n$ of data selected from the training set $\{\boldsymbol{C}_i\}_{i=1}^{n}$. Without loss of generality, assume that the first m frames are chosen. These m samples are trained offline. This step can be described in the following equation:

$$\{\boldsymbol{C}_1, \boldsymbol{C}_2 \ldots \boldsymbol{C}_m\}, 1 \leq m \ll n \xrightarrow{offline} \text{coefficient matrix } \Big[\boldsymbol{K}\Big] \Big[\boldsymbol{\alpha} \quad -\rho\Big]^\top, \tag{5.66}$$

where $\Big[\boldsymbol{K}\Big]$ and $\Big[\boldsymbol{\alpha} \quad -\rho\Big]^\top$ are defined in eq. (5.29).

Step 3: After learning the first m samples, the coefficient matrices, $\boldsymbol{K}$ and $\Big[\boldsymbol{\alpha} \quad -\rho\Big]^\top$, are obtained. The online LS-OC-SVM method (Section 5.2.2) is applied to learn the remaining $n-m$ samples $\{\boldsymbol{C}_{m+1}, \boldsymbol{C}_{m+2} \ldots \boldsymbol{C}_n\}$. This step can be expressed as:

$$\{\boldsymbol{C}_{m+1}, \boldsymbol{C}_{m+2} \ldots \boldsymbol{C}_n\}, \Big[\boldsymbol{K}\Big] \xrightarrow{online} \text{coefficient matrix } \Big[\boldsymbol{K}\Big] \Big[\boldsymbol{\alpha} \quad -\rho\Big]^\top. \tag{5.67}$$

Step 4: Based on the coefficient matrix, $\Big[\boldsymbol{\alpha} \quad -\rho\Big]^\top$, the distance of the training samples $\{\boldsymbol{C}_i\}_{i=1}^{n}$ and the incoming test sample, $\boldsymbol{C}_{n+l}$, with respect to the decision plane is computed. By comparing the distances of the samples, an abnormal event is detected:

$$dis(\boldsymbol{C}_{n+l}) = \frac{|(\sum_{i=1}^{n} \alpha_i \kappa(\boldsymbol{C}, \boldsymbol{C}_i) - \rho)|}{\|\boldsymbol{\alpha}\|} \tag{5.68}$$

$$= \begin{cases} 1 & \text{if } f(\mathbf{C}_{n+l}) \geq T_{dis}, \\ -1 & \text{if } f(\mathbf{C}_{n+l}) < T_{dis}, \end{cases} \tag{5.69}$$

where $\boldsymbol{C}_{n+l}$ is the covariance matrix descriptor of the $(n+l)-th$ frame needed to be classified, and $\boldsymbol{C}_i$ is the sample of the training data. "1" corresponds to an abnormal frame; "-1" corresponds to a normal frame. T_{dis} is the threshold of the distance, it is the maximum distance of the training samples to the hyperplane.

5.2.4.2 Sparse online LS-OC-SVM strategy

The abnormal event detection via sparse online least squares one-class SVM (sparse online LS-OC-SVM) is introduced below. A subset of the samples is chosen to form the dictionary, $\boldsymbol{C}_{\mathfrak{D}}$, making a sparse representation of the training data. The initial dictionary, $\boldsymbol{C}_{\mathfrak{D}}$, is learned offline. Each remaining training sample is learned one-by-one online. Meanwhile, it is checked to be included, or not, into the dictionary. The test datum is classified based on the dictionary. The feature extraction step (Step 1) and the detection step (Step 4) are the same as the ones presented in Section 5.2.4.1. Owing to the dictionary, the training steps are different.

Step 2-sparse: The second step is applying LS-OC-SVM to train the initial dictionary offline. The first m samples are the initial dictionary denoted as $\boldsymbol{C}_{\mathfrak{D}}$. This step can be generalized as:

$$\boldsymbol{C}_{\mathfrak{D}} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \ldots, \boldsymbol{C}_m\}, 1 \leq m \ll n \xrightarrow{offline} \text{coefficient matrix } \begin{bmatrix}\boldsymbol{K}\end{bmatrix} \begin{bmatrix}\boldsymbol{\beta} & -\rho\end{bmatrix}^{\top}. \quad (5.70)$$

Step 3-sparse: After learning the initial dictionary, $\boldsymbol{C}_{\mathfrak{D}}$, including the first m ($1 \leq m \ll n$) samples, the remaining training samples, $\{\boldsymbol{C}_{m+1}, \boldsymbol{C}_{m+2}, \ldots, \boldsymbol{C}_n\}$, are learned via sparse online LS-OC-SVM described in Section 5.2.3. This step can be described in the following equations:

$$\begin{aligned} &\{\boldsymbol{C}_{\mathfrak{D}}, \boldsymbol{C}_k\}, m < k \leq n \xrightarrow{sparse\ online} \\ &\text{coefficient matrix } \begin{bmatrix}\boldsymbol{\beta} & -\rho\end{bmatrix}^{\top} \\ &\begin{cases} \boldsymbol{C}_{\mathfrak{D}} := \boldsymbol{C}_{\mathfrak{D}} \cup \boldsymbol{C}_k & \text{if} \quad \varepsilon_t \geq \mu_0 \\ \boldsymbol{C}_{\mathfrak{D}} := \boldsymbol{C}_{\mathfrak{D}} & \text{if} \quad \varepsilon_t < \mu_0, \end{cases} \end{aligned} \quad (5.71)$$

where $\boldsymbol{C}_{\mathfrak{D}}$ is the dictionary and $\boldsymbol{C}_k$ is a new incoming remaining sample in the training dataset. According to the coherence criterion introduced in Section 5.2.3, if the new sample, $\boldsymbol{C}_k$, satisfies the dictionary updated condition, it will be included into the dictionary, $\mathbf{C}_{\mathfrak{D}}$.

5.2.5 Abnormal Event Detection Results

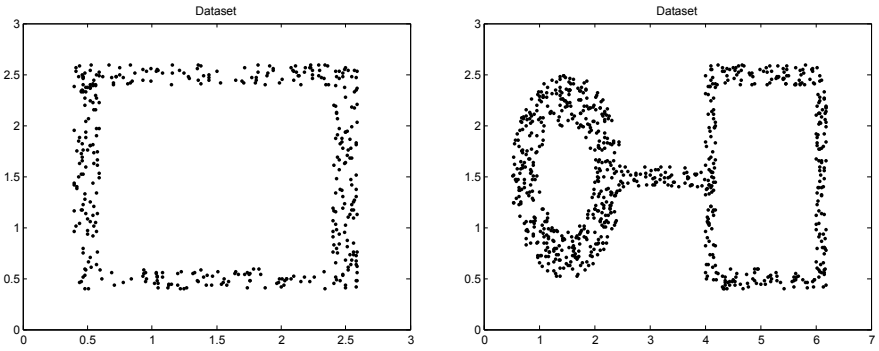
This section presents the results of experiments conducted to illustrate the performance of the two proposed classification algorithms, online least square one-class SVM (online LS-OC-SVM) and sparse online least square one-class SVM (sparse online LS-OC-SVM). The two-dimensional synthetic distribution dataset and the University of Minnesota (UMN) [UMN 2006] dataset are used.

5.2.5.1 Synthetic Dataset via Online LS-OC-SVM and Sparse Online LS-OC-SVM

Two synthetic data, "square" and "ring-line-square" [Hoffmann 2007], are used. The "square" consists of four lines, 2.2 in length and 0.2 in width. In the area of these lines, 400 points were randomly dispersed with a uniform distribution. The "ring-line-square" distribution is composed of three parts: a ring with an inner diameter of 1.0 and an outer diameter of 2.0, a line of 1.6 in length and 0.2 in width, and a square the same as dataset "square" introduced above. 850 points are randomly dispersed with a uniform distribution. These two data are shown in Fig.5.8.

The first sample is used for initializing the online LS-OC-SVM proposed in Section 5.2.2; the 399 remaining samples in "square" and 849 remaining samples in "ring-ling-square" are learned in the online manner.

Via the sparse online LS-OC-SVM method proposed in Section 5.2.3, the first sample is trained offline, and this sample is considered the initial dictionary. Then, each arrival



(a) Dataset square (b) Dataset ring-line-square

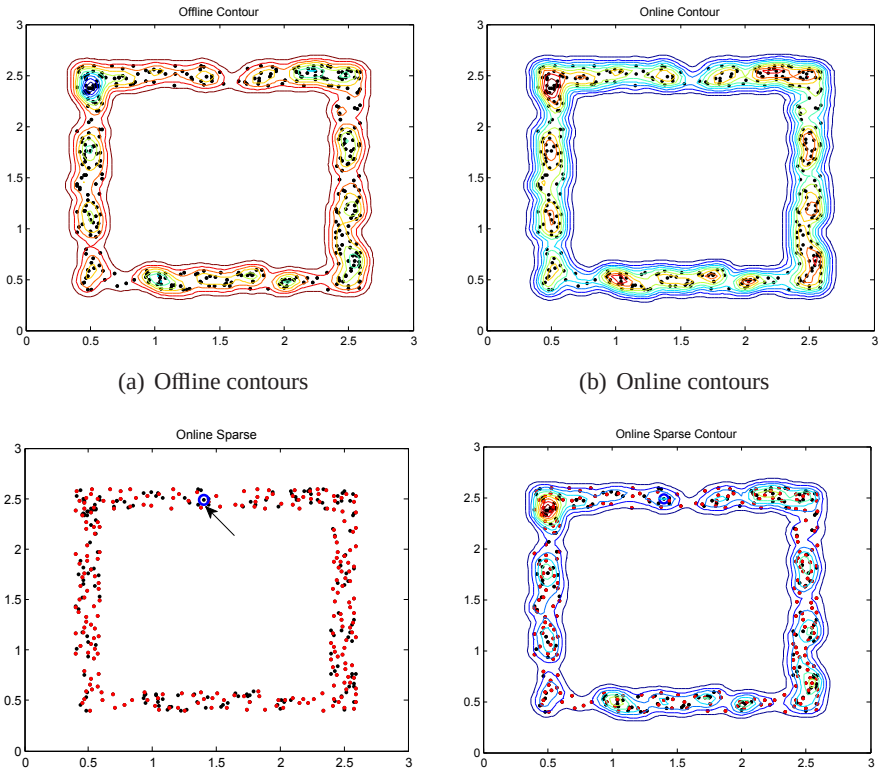
Figure 5.8: Synthetic datasets. (a) Dataset square. (b) Dataset ring-line-square.

sample in 399 remaining samples in "square" and 849 remaining samples in "ring-ling-square" are checked by the coherence criterion to determine whether the dictionary should be retained or updated by including the new element.

The distances are shown in contours illustrating the boundary. The contours of "square" and "ring-line-square" are shown in Fig. 5.9 and 5.10, respectively. Gaussian kernel was used in these two data, with bandwidth $\sigma = 0.065$. The preset threshold of the coherence criterion is $\mu_0 = 0.08$. The detection results obtained by these two online training algorithms are the same as the ones when training data were learned in a batch model.

5.2.5.2 Abnormal Visual Event Detection via Online LS-OC-SVM

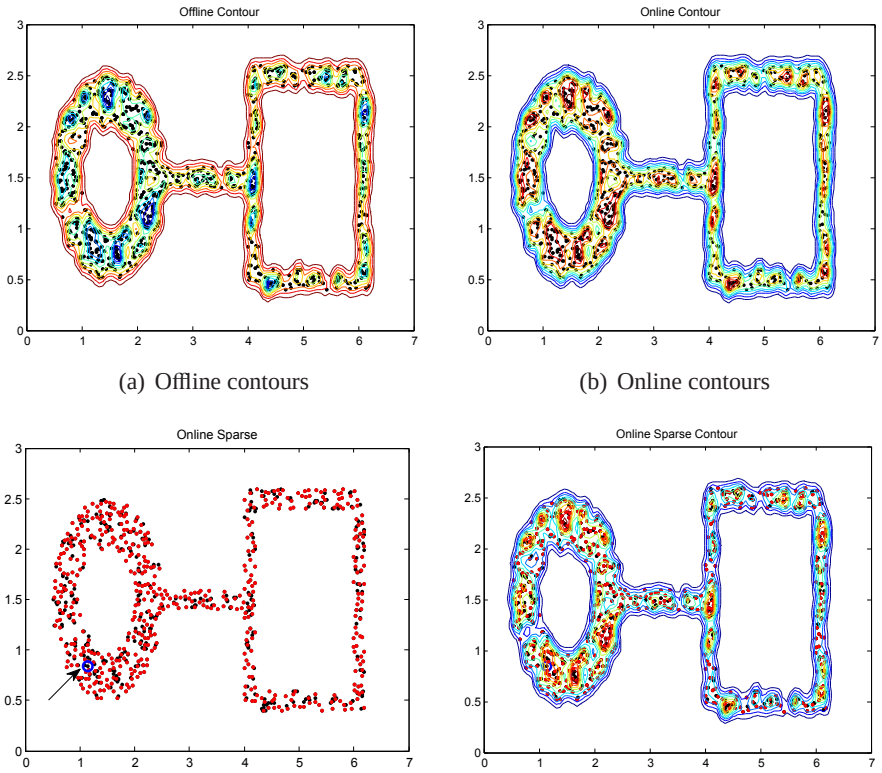
UMN dataset [UMN 2006] results via online LS-OC-SVM which is proposed in *Section 5.2.2* are shown below. The detection results of lawn scene, indoor scene and plaza scene are shown in Fig.5.11, Fig.5.12 and Fig.5.13, respectively. A Gaussian kernel for the covariance matrix in the Lie group is used. Various values of the variance, σ, in the Gaussian function and the penalty factor, C, are chosen to form the receiver operating characteristic (ROC) curve. In the indoor scene, time lags of the frame labels lead to the lower area under the ROC curve (AUC) value. In the last few frames, labeled as abnormal of abnormal sequences, there are no people, while, in the training samples, there are no people in the upper half of the image. The covariance of the training frame is similar to the covariance of the abnormal frame without people. Our covariance feature descriptor-based classification method cannot distinguish between these two situations. However, this issue can be resolved by utilizing the foreground information. For example, if there are no moving objects in the frame, this frame is immediately classified as abnormal. The results of these three scenes show that the covariance descriptor can distinguish between normal and abnormal events. The performance of online LS-OC-SVM is almost the same as that of the offline method.



(a) Offline contours (b) Online contours

(c) Sparse online data (d) Sparse online contours

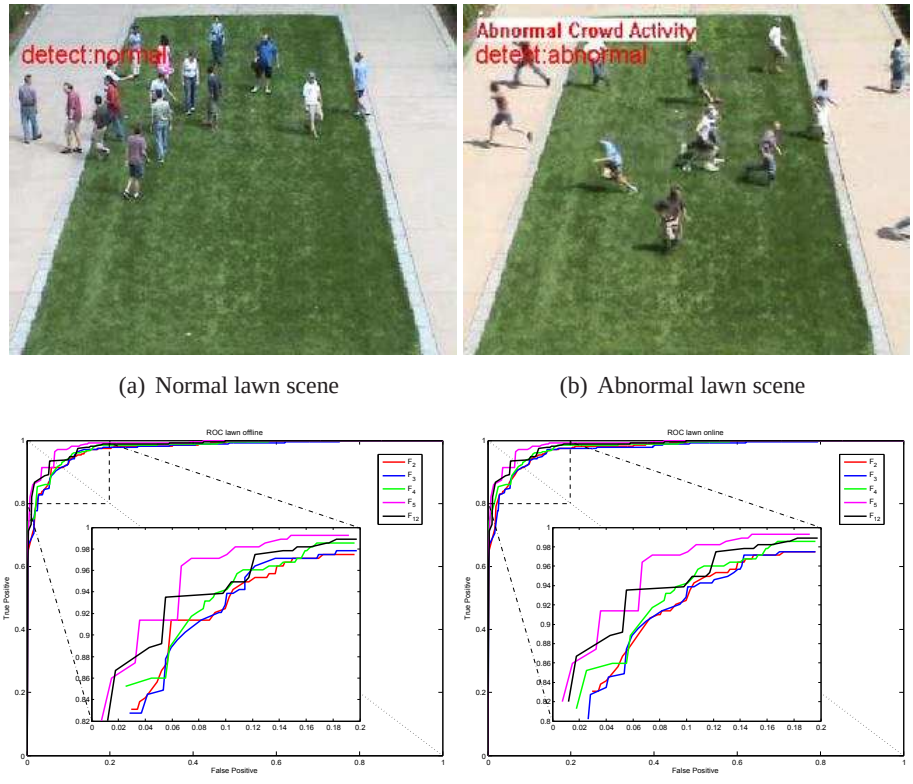
Figure 5.9: Offline, online least squares one-class SVM and sparse online least squares one-class SVM results of *'square'* dataset. The figure might be viewed better electronically, in color and enlarged. (a) The contours of the distances when all the date are trained as one batch offline. (b) The contours of the distances when the data are trained via online LS-OC-SVM. (c) The blue circle (pointed out by the arrow) shows the original dictionary. The red points show the *232* new data which are included into the dictionary via sparse online LS-OC-SVM. (d) The contours of the distances when the data are trained via sparse online LS-OC-SVM.



(a) Offline contours (b) Online contours

(c) Sparse online data (d) Sparse online contours

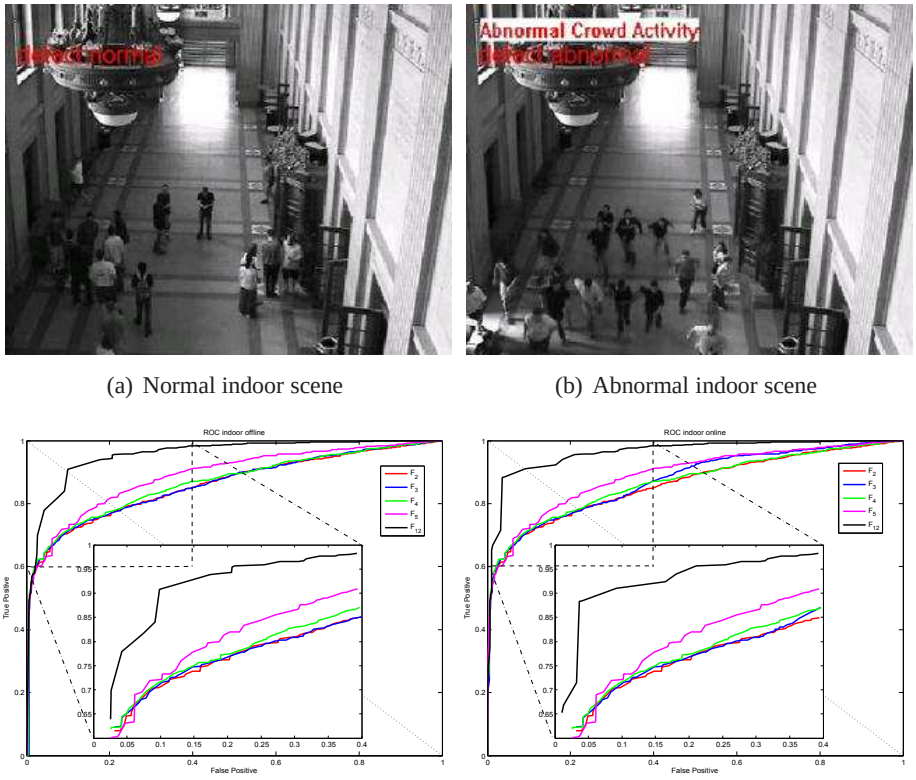
Figure 5.10: Offline, online least squares one-class SVM and sparse online least squares one-class SVM results of *'ring-line-square'* dataset. (a) The contours of the distances when all the date are trained as one batch offline. (b) The contours of the distances when the data are trained via online LS-OC-SVM. (c) The blue circle (pointed out by the arrow) shows the original dictionary. The red points show the *534* new data which are included into the dictionary via sparse online LS-OC-SVM. (d) The contours of the distances when the data are trained via sparse online LS-OC-SVM.



(a) Normal lawn scene (b) Abnormal lawn scene

(c) ROC train all data (d) ROC online

Figure 5.11: Abnormal *frame* event detection results of the lawn scene based on *frame* COV descriptor via online least squares one-class SVM. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) ROC curve of different features F of the lawn scene results via one-class SVM. All the training samples are learned together offline. The biggest AUC value is 0.9874. (d) ROC curve of different features F results via online LS-OC-SVM. The biggest AUC value is 0.9874.



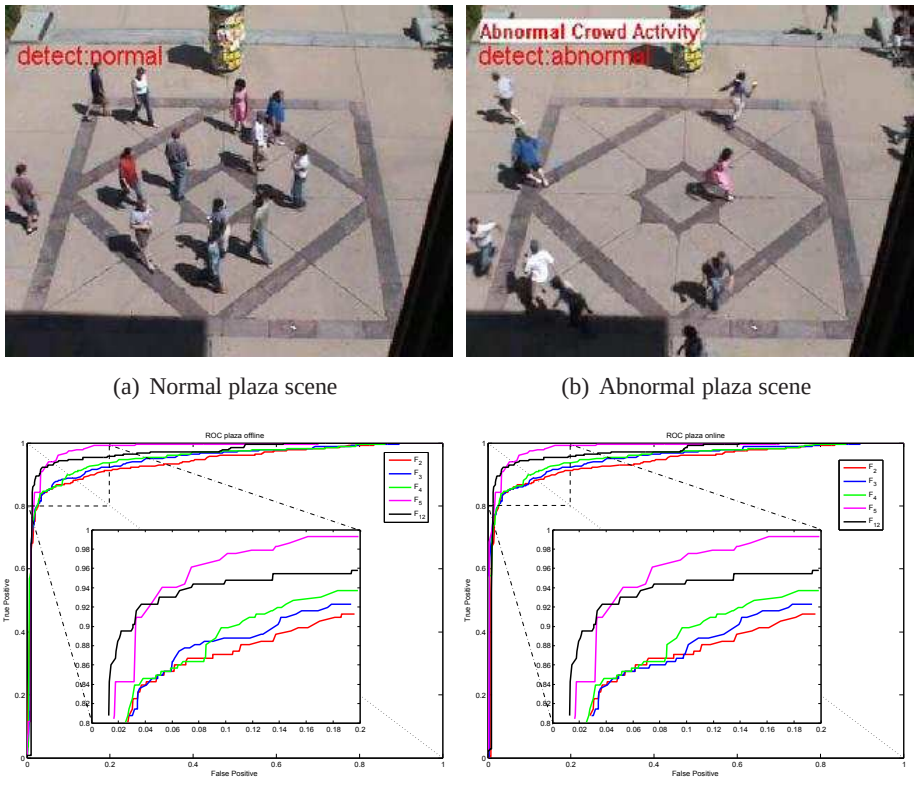
(a) Normal indoor scene

(b) Abnormal indoor scene

(c) ROC train all data

(d) ROC online

Figure 5.12: Abnormal *frame* event detection results of the indoor scene based on *frame* COV descriptor via online least squares one-class SVM. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) ROC curve of different features F of the lawn scene results via one-class SVM. All the training samples are learned together offline. The biggest AUC value is 0.9548. (d) ROC curve of different features F results via online LS-OC-SVM. The biggest AUC value is 0.9619.



(a) Normal plaza scene (b) Abnormal plaza scene

(c) ROC train all data (d) ROC online

Figure 5.13: Abnormal *frame* event detection results of the plaza scene based on *frame* COV descriptor via online least squares one-class SVM. (a) The detection result of one normal frame. (b) The detection result of one abnormal panic frame. (c) ROC curve of different features F of the plaza scene results via one-class SVM. All the training samples are learned together offline. The biggest AUC value is 0.9800. (d) ROC curve of different features F results via online LS-OC-SVM. The biggest AUC value is 0.9839.

5.2.5.3 Abnormal visual events detection via sparse online LS-OC-SVM

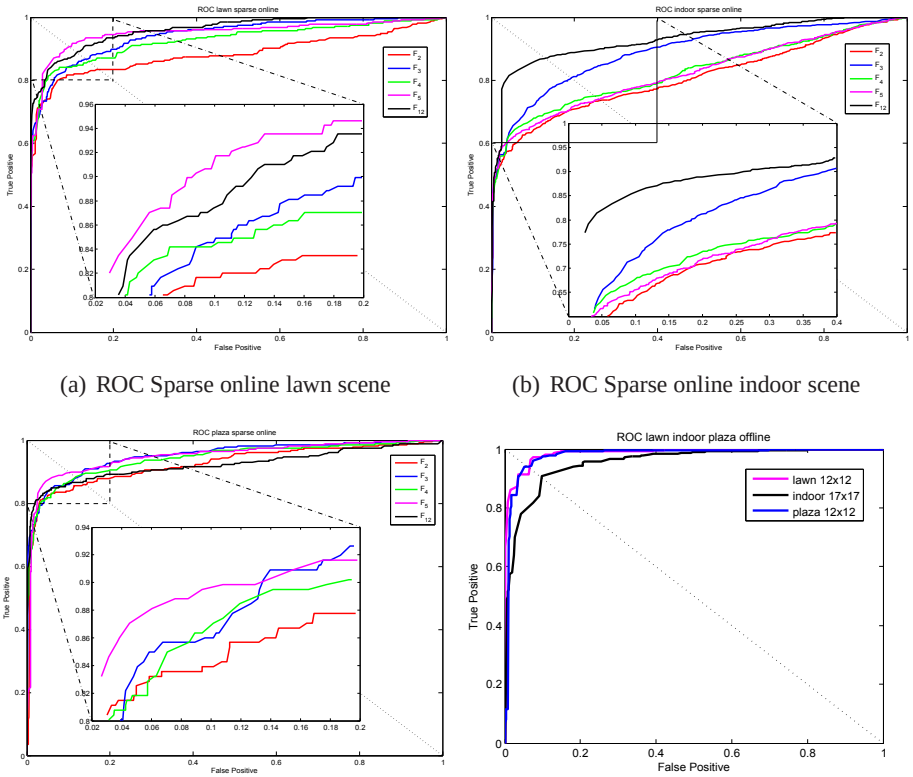
The UMN dataset abnormal events detection results via sparse online LS-OC-SVM which is proposed in *Section 5.2.3* are shown below. Take the lawn scene as the examples, the 1^{st} normal sample from the training samples is included into the dictionary firstly, and then other remaining training samples are learnt online by the sparse online LS-OC-SVM method. If the newly arrival sample satisfies the subspace sparse control criterion, the dictionary and the classifier is updated through the training period. The ROC curve of detection results of lawn scene, indoor scene and plaza scene are shown in Fig.5.14(a), (b) and (c) respectively.

The resulting performances when all training samples are learned offline via one-class SVM (OC-SVM), learned via least squares one-class SVM (LS-OC-SVM), learned via online least squares one-class SVM (online LS-OC-SVM) and learned via sparse online least squares one-class SVM (sparse LS-OC-SVM), are shown in Table 5.3. The LS-OC-SVM algorithm obtains better performance than the original OC-SVM. The performances of online and sparse online strategy results are similar to the resulting performances when all training samples are learned offline. The sparse online strategy can be computed efficiently and can adapt to the memory requirement.

The resulting performances of the covariance matrix descriptor-based online least squares one-class SVM method, and of state-of-the-art methods, are shown in Table 5.4. The covariance matrix-based online abnormal frame detection method obtains competitive performance. In generally, our sparse online LS-OC-SVM method is better than others, except sparse reconstruction cost (SRC) [Cong 2011]. In that paper, multi-scale histogram of optical flow (HOF) was taken as a feature and a testing sample was classified by its sparse reconstruction cost, through a weighted linear reconstruction of the over-complete normal basis set. However, the computation of the HOF takes more time than the computation of covariance. By adopting the integral image [Tuzel 2006], the covariance matrix descriptor of the subimage can be computed conveniently. The covariance descriptor can appropriately be used to analyze partial image movement. In [Cong 2011], the whole training dataset was saved in the memory in advance; then, the dictionary was chosen as an optimal subset for reconstructing. Our sparse online LS-OC-SVM strategy enables one to train the classifier with sequential inputs. This property makes our proposed method extremely suitable to handle large volumes of training data, while the method in [Cong 2011] fails to work due to lack of memory.

5.3 Conclusion

In this chapter, we proposed two online abnormal detection methods. The first method is based on the online nonlinear one-class SVM classification method. The second method is based on online least squares one-class SVM (online LS-OC-SVM) and sparse online least squares one-class SVM (sparse online LS-OC-SVM). Online LS-OC-SVM learns training samples sequentially; sparse online LS-OC-SVM incorporates the coherence criterion to form the dictionary for a sparse representation of the detector. The proposed detection algorithms have been tested on a synthetic dataset and a real-world video dataset yielding



(a) ROC Sparse online lawn scene

(b) ROC Sparse online indoor scene

(c) ROC Sparse online plaza scene

(d) ROC offline

Figure 5.14: ROC curve of abnormal *frame* events detection results of the lawn, plaza, and indoor scenes based on *frame* COV descriptor via sparse online least squares one-class SVM. (a) ROC curve of different features F results via sparse online LS-OC-SVM of lawn scene. The biggest AUC value is 0.9609. (b) Sparse online LS-OC-SVM results of indoor scene. The biggest AUC value is 0.9287. (c) Sparse online LS-OC-SVM results of plaza scene. The biggest AUC value is 0.9515. (d) The ROC curve of best performance of lawn, plaza and indoor scene when the training samples are learnt offline. The biggest AUC value of lawn, plaza and indoor are 0.9874, 0.9800 and 0.9548.

Features	Area under ROC		
	lawn	indoor	plaza
training samples are learned offline			
$F_2(6\times 6du)$	0.9755	0.8605	0.9422
$F_3(6\times 6dv)$	0.9738	0.8603	0.9489
$F_4(8\times 8)$	0.9788	0.8662	0.9538
$F_5(12\times 12)$	**0.9874**	0.8900	**0.9800**
$F_{12}(17\times 17)$	0.9832	**0.9548**	0.9680
Online LS One-class SVM			
$F_2(6\times 6du)$	0.9755	0.8616	0.9403
$F_3(6\times 6dv)$	0.9720	0.8730	0.9517
$F_4(8\times 8)$	0.9795	0.8670	0.9563
$F_5(12\times 12)$	**0.9874**	0.8904	**0.9839**
$F_{12}(17\times 17)$	0.9833	**0.9619**	0.9699
Sparse Online LS One-class SVM			
$F_2(6\times 6du)$	0.8840	0.8077	0.9245
$F_3(6\times 6dv)$	0.9435	0.8886	**0.9515**
$F_4(8\times 8)$	0.9269	0.8266	0.9428
$F_5(12\times 12)$	0.9510	0.8223	0.9501
$F_{12}(17\times 17)$	**0.9609**	**0.9287**	0.9229

Table 5.3: AUC of abnormal *frame* event detection results based on *frame* covariance matrix descriptor constructed by different features F via least squares one-class SVM (LS-OC-SVM) (*Section5.2.1*), online LS-OC-SVM (*Section5.2.2*,*Section5.2.4.1*), and sparse online LS-OC-SVM (*Section5.2.3*,*Section5.2.4.2*) of UMN dataset. The biggest value of each method is shown in bold.

successful results in detecting abnormal events.

Method	Area under ROC		
	lawn	indoor	plaza
Social Force [Mehran 2009]	0.96		
Optical Flow [Mehran 2009]	0.84		
NN [Cong 2011]	0.93		
SRC [Cong 2011]	0.995	0.975	0.964
STCOG [Shi 2010]	0.9362	0.7759	0.9661
LS-SVM (Ours)	0.9874	0.9548	0.9800
Online (Ours)	0.9874	0.9619	0.9839
Sparse Online(Ours)	0.9609	0.9287	0.9515

Table 5.4: The comparison of our proposed *frame* covariance matrix descriptor, online least squares one-class SVM (online LS-OC-SVM) and sparse online least squares one-class SVM (sparse online LS-OC-SVM) based methods with the state-of-the-art methods for abnormal *frame* event detection of UMN dataset.

CHAPTER 6

Conclusions and Perspectives

Contents

6.1 Contributions

Abnormal detection is a key component in intelligent video surveillance. In this thesis, our contributions are summarized as follows. Firstly, we adopt optical flow as the basic movement information, the block of the optical flow is constructed as mid-level feature descriptor. One-class support vector machines (OC-SVM) after learning one category of positive samples (normal samples), yields a decision function for detecting abnormal frames. Secondly, histograms of optical flow orientation (HOFO) is proposed as a new feature descriptor encoding the movement information. Thirdly, a covariance matrix descriptor fusing the optical flow information and the intensity is also proposed as an input to the classification algorithm. By adopting the integral image, the covariance can be efficiently computed at a frame level or at blob level. Fourthly, as the abnormal detection is usually applied on a long video sequence, two on-line abnormal detection methods are proposed. One is based on the support vector data description (SVDD), with a dictionary-based sparsification. Two strategies are proposed to construct and update the dictionary. Another on-line abnormal detection method, based on least squares one-class support vector machine (LS-OC-SVM) with a sparse formulation, is also proposed.

6.2 Perspectives

In crowded scenes, in one camera view, the people are overlapped with others. It is difficult to detect people from the occluded group. This situation can be improved by multi-cameras. By fusing the information from the multi-views, the people can be separated, if the person could be captured by a camera. The camera calibration technology could be used in this situation.

The object selection strategies for extracting the blobs should also be more robust. In this thesis, we used the background subtraction method, but this method is not very stable under lighting changes and unstable cameras. Other feature selection strategies which does

not depend on consequent frames, such as SIFT (scale-invariant feature transform) feature detection, should be tested and integrated to enhance the abnormal detection.

The feature descriptor can be improved by including the temporal information. In this thesis, the optical flow encodes a type of temporal information between successive frames. Other types of temporal features, such as 3-dimension histograms and temporal-spatial blocks, should be considered. Also, the semantic event models representing the sub-events and state event models representing the relationship of the sub-events by directional graph can be used to improve the discriminative capability of the abnormal detector.

The abnormal event detection can be combined with other computer vision techniques, such as single people action recognition, face detection, and texture analysis. For instance, if an abnormal event occurs, some people will be labeled and individually tracked. Also, the faces can be detected and recognized, and the texture of the clothes can be analyzed. In other words, abnormal event detection could be considered as a pre-processing step, other procedures to find deeper information for surveillance are to be post-deployed.

The voice information in the video sequence is also to be considered for abnormal event detection. The sound should be fused with video to detect and recognize events.

From a methodological side, advances in machine learning theory could improve the performance of video event detection. The kernel methods, online learning, sparse representation and deep learning theories can be used to enhance the learning and classification. As the amount of video streams will continuously grow, big data research will be helpful for dealing with video event detection problems.

Appendix A

Résume de Thèse en Français

Comme la demande de Ecole Doctorale de l'Université de Technologie de Troyes, cette appendice est un résumé substantiel en Français de 20 à 30 pages, pour les mémoires rédigés en Anglais.

A.1 Introduction

L'un des principaux domaines de recherche en vision par ordinateur est la surveillance visuelle. Le défi scientifique dans ce domaine comprend la mise en œuvre de systèmes automatiques pour obtenir des informations détaillées sur le comportement des individus et des groupes. En particulier, la détection de mouvements anormaux de groupes d'individus nécessite une analyse sophistiquée des images vidéo.

La détection d'événements anormaux, étudiée dans le cadre de cette thèse, est basée sur la conception d'un descripteur caractérisant les informations de mouvement et la conception de méthodes de classification non linéaire. Dans cette thèse, trois types de caractéristiques sont étudiés : flux optique global, les histogrammes des orientations du flux optique (HOFO) et le descripteur de covariance (COV). Sur la base de ces descripteurs, des algorithmes se basant sur les machines à vecteurs support (SVM) mono-classe sont utilisés pour détecter des événements anormaux. Ensuite, deux stratégies en ligne de SVM mono-classe ont été proposées pour une implémentation en temps réel des algorithmes de détection. La Fig.A.1 montre quelques exemples illustratifs des travaux qui sont menés dans le cadre de cette thèse.

A.2 Détection sur la base du flux optique et des histogrammes d'orientation

Dans cette section, nous introduisons les descripteurs se basant sur des caractéristiques de flux optique, et sur les histogrammes des orientations du flux optique (HOFO). La méthode d'extraction de blob anormal dans une scène vidéo est aussi décrite dans cette section.

A.2.1 Détection d'anormalies sur la base du flux optique

Comme l'action peut être caractérisée par la direction et l'amplitude du mouvement de l'objet dans la scène, on utilise le flux optique pour extraire des caractéristiques de bas

(a) Normale scène de la place

(b) Normale scène du intérieur

(c) Scène du mall

(d) Anormale scène de la place

(e) PETS

(f) Scène des deux personnes

Figure A.1: Des exemples des scènes normaux et anormaux. (a) Toutes les personnes se déplacent normalement dans un lieu public (jeux de données UMN). (b) Se déplacent normalement dans une gare (jeux de données UMN). (c) Une personne se déplace d'une manière anormale alors que toutes les autres personnes ont un mouvement normal. (d,e,f) Des scènes où il y a des mouvements anormaux que ce soit au niveau du groupe ou au niveau des individus.

ù

niveau. Le flux optique peut fournir des informations importantes sur la disposition spatiale des objets et le degré de changement de cette structure spatiale [Horn 1981]. Il s'agit de la distribution de la vitesse apparente de déplacement des modèles de brillance d'une image. Horn et Schunck ont proposé un algorithme de calcul du flux optique en introduisant une contrainte globale de régularité. La méthode Horn-Schunck (HS) combine un terme de données avec un terme spatial. Le terme de données exploite les informations sur les variations des caractéristiques de bas niveau de l'image et le terme spatial pénalise les disparités du champ du flux optique. Le flux optique est calculé en minimisant l'énergie globale fonctionnelle suivante:

$$E = \int\int [(I_x u + I_y v + I_t)^2 + \alpha^2(\|\nabla u\|^2 + \|\nabla v\|^2)]dxdy, \tag{A.1}$$

où I_x,I_y et I_t sont les dérivés de l'intensité d'image le long de x, y et t, u et v sont les composantes horizontale et verticale du flux optique, α est le paramètre de régularisation. Les équations de Lagrange sont utilisées pour minimiser la fonctionnelle, ce qui donne :

$$\begin{cases} I_x(I_x u + I_y v + I_t) - \alpha^2 \triangle u = 0 \\ \\ I_y(I_x u + I_y v + I_t) - \alpha^2 \triangle v = 0, \end{cases} \tag{A.2}$$

sous la contrainte que :

$$\begin{cases} \triangle u(x,y) = \overline{u}(x,y) - u(x,y) \\ \\ \triangle v(x,y) = \overline{v}(x,y) - v(x,y), \end{cases} \tag{A.3}$$

où $\overline{u}$ et $\overline{v}$ sont les moyennes pondérées de u et v calculées dans une zone autour de la position du pixel. Le flux optique est calculée dans un schéma itératif tel que représenté ci-dessous:

$$\begin{cases} u^{k+1} = \overline{u}^k - \frac{I_x(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2} \\ \\ v^{k+1} = \overline{v}^k - \frac{I_y(I_x\overline{u}^k + I_y\overline{v}^k + I_t)}{\alpha^2 + I_x^2 + I_y^2}, \end{cases} \tag{A.4}$$

où k désigne l'itération de l'algorithme. Un seul pas de temps a été pris de telle sorte que les calculs sont basés sur seulement deux images successives.

Dans la suite, nous décrivons le système global proposé pour détecter des événements anormaux en se basant sur le flux optique. Supposons que les frames $\{I_1, I_2, \ldots, I_n\}$ sont considérés comme des événements normaux. Dans le problème de détection d'anomalies, il est supposé que les données d'une seule classe, la classe positive (ou la scène normale), sont disponibles. Le cadre du SVM mono-classe est alors bien adaptée à la spécificité de ce problème de détection d'événement normaux où seuls les échantillons de scènes normaux sont disponibles. L'architecture générale de la méthode de détection est présentée dans la Fig.A.2.

Ci-dessous, on décrit les principales étapes de l'algorithme proposé:

Étape 1: La première étape consiste à calculer les caractéristiques de flux optique d'image à échelle de gris à. Chaque image est traitée via Horn-Schunck (HS) pour obtenir les caractéristiques en mouvement à chaque pixel. Cette étape peut ¨otre présentée comme suit:

$$\{I_1, I_2, \ldots, I_n\} \xrightarrow{HS} \{OP_1, OP_2, \ldots, OP_n\}, \tag{A.5}$$

où $\{I_1, I_2, \ldots, I_n\}$ sont les images originales et $\{OP_1, OP_2, \ldots, OP_n\}$ sont le flux optique correspondant.

Étape 2: La procédure SVM mono-classe est utilisée pour classer les échantillons de caractéristiques de images vidéo entrants. Trois stratégies sont proposées pour l'obtention des caractéristiques de l'image. L'image d'esquisse pour le choix des caractéristiques est représenté dans Fig.A.3.

Méthode 1: Il s'agit de prendre le flux optique au niveau de chaque pixel de l'image sous forme d'échantillons de caractéristiques, comme le montre la Fig.A.3(a). La séquence vidéo dans notre travail est étiquetée comme étant normal ou anormal . Ces étiquettes sont utilisées pour l'évaluation des performances. Les données d'entrée pour les SVM mono-classe sont extraites des images normaux. Ceci consiste à prendre le flux optique $OP_{i,j,k}$ comme fonction $F_{i,j,k}$ pour (i, j)-th pixel sur le cadre k. Pour chaque point de coordonnées

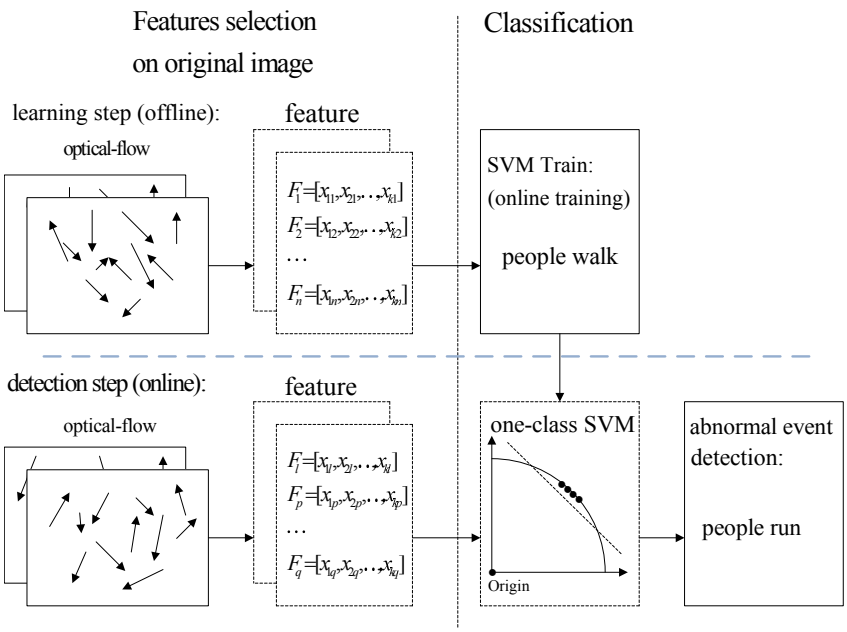
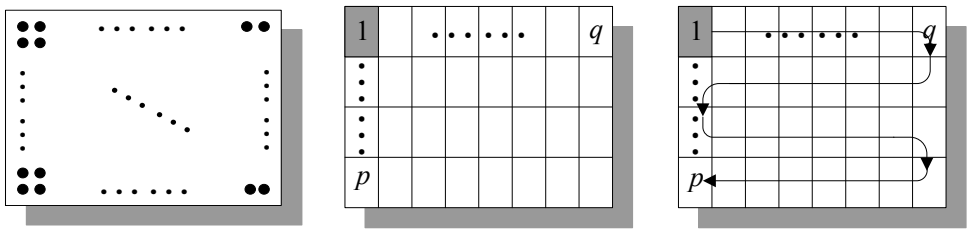


Figure A.2: Architecture du système global de détection d'anomalies se basant sur le flux optique et l'algorithme SVM mono-classe.

cartésiennes (i, j) des n images d'entrée, nous pouvons obtenir la formation échantillons $F_{i,j,1\ldots n}$, $n \geq 1$, puis calculer les vecteurs de support. Sur la base des vecteurs de support, les échantillons entrants $F_{i,j,n+1\ldots m}$ à coordonner (i, j) sont détectés. Pour l'ensemble de l'image, les événements anormaux sont détectés pixel par pixel.

Méthode 2: Il s¡¯agit de prendre le flux optique de tous les points dans le bloc des échantillons. Dans cette stratégie, l'image est segmentée en plusieurs blocs, comme le montre la Fig.A.3(b), l'image est divisée en blocs de $p \times q$, p est le nombre de blocs à la verticale (hauteur) et q est le nombre de blocs à l'horizontale de l'image. La hauteur du bloc est h pixels, la longueur du bloc est w pixels, il y a des points $h \times w$ dans le bloc. La fonctionnalité de bloc au i-th ligne et de la colonne j-th dans le cadre de k-th est à noter que $F^{\text{block}}_{i,j,k}$. Pour chaque bloc, la fonction F^{block} est organisée par le flux optique de tous les points de la forme $\{OP_1, OP_2, OP_3, \cdots, OP_{h\times w}\}$. Pour les flux vidéo, prendre les fonctions de bloc à des images normaux que les échantillons de formation pour les SVM une classe, puis des événements anormaux sont détectés block-by-block.

Méthode 3: L'image est également divisée en blocs, mais les échantillons sont tous les blocs de l'image d'éntrée, comme illustré sur la Fig.A.3(c). D'une manière similaire à la *Méthode 2*, nous décomposons l'image en $p \times q$ blocs, la taille de chaque bloc es-t $h \times w$. À l'image de k, l'échantillon caractéristique de tous les blocs de ce cadre est $\{F^{\text{block}}_{1,1,k}, F^{\text{block}}_{1,2,k}, \ldots, F^{\text{block}}_{p,q,k}\}$, un vecteur de dimension $(p \times q \times k) \times (h \times w)$. Pour obtenir les données de formation à l'image normale de 1-e à n-ième, un vecteur de dimension $(p\times q\times k)\times(h\times w)$. Pour la détection, l'échantillon d'essai est la caractéristique d'un bloc.



(a) Pixel par pixel (b) Bloc par bloc (c) Bloc^{all} par bloc

Figure A.3: Trois stratégies pour choisir les caractéristiques de flux optique. (a) Choisir les caractéristiques pixels par pixel. (b) Choisir les caractéristiques bloc-par-bloc. (c) Choisir tous les blocs dans le cadre de l'échantillon d'apprentissage, et test en bloc.

A.2.2 Extraction et détection de blob anormaux

Dans le cas d'une caméra fixe, la segmentation d'objet en mouvement grâce à des méthodes de soustraction de fond. Cependant, l'extraction de blobs est peu efficace à cause de chevauchements éventuels de plusieurs objets en mouvement dans la scène. Comme le montre la Fig.A.4(a), la personne à l'intérieur du premier rectangle est confondue avec une personne voisine. Comme les mouvements de ces personnes sont différents, nous proposons, dans cette thèse une méthode pour améliorer l'extraction des blobs en se basant sur le flux optique. La méthode est résumée dans l'algorithme 1, et illustrée dans la Fig.A.4(c).

(a) (b) (c)

Figure A.4: Les blobs avant et après la méthode d'extraction proposé. (a) 2 blobs extraits sur la base du gabarit de premier plan. (b) 3 blobs extraits par la méthode d'extraction de blob proposé, qui est basé sur le modèle de premier plan et du flux optique. (c) L'image du flux optique de la Fig.(a)(b).

On présente, dans la suite, les détails de la méthode proposée pour l'extraction de blobs en exploitant le flux optique.

Étape 1: La première étape consiste à l'étiquetage des composantes connexes d'une image de premier plan binaire. Représentent B^k_{FG} pour le blob de k-th à l'image de premier plan. Comme il ya généralement des occlusions des gens, certains rectangles contiennent plusieurs objets. Comme le montre la Fig.3(a), le 1-er rectangle comprend deux personnes.

Étape 2: La deuxième étape est l'étiquetage des blobs en fonction du flux optique. Si la taille du blob de premier plan est plus grand qu'un seuil de pré-réglage T_{blb}, le flux

Algorithm 3 Extraction de Blob.

Require:

Image de premier plan *FG*, flux optique *OP*

1: Étiquter les blobs dans *FG*, le blob à l'image de premier plan B^k_{FG} est obtient.
2: **if** Taille de blob à $FG \geq$ seuil de préréglage T_{blb} **then**
3: Le flux optique I_{OP} dans le blob est pris en compte.
4: Les flux optiques avec des amplitudes et des directions similaires sont regroupés.
5: Supprimer groupe de redondance par NMS algorithme, blob B^i_{OP} est obtient. La région $B_{\mathrm{RM}} = B_{\mathrm{FG}} - B_{\mathrm{OP}}$ restante.
6: Traverse B_{RM} par un rectangle référence de taille prédéfinie. NMS algorithme permet de choisir le blob B^j_{RM} du blob enregistré B_{RM}.
7: Remplacer blob B^k_{FG} par blob $B^i_{\mathrm{OP}} + B^j_{\mathrm{RM}}$.
8: Les blobs de l'image sont extraits.

optique dans ce domaine est pris en compte pour affiner l'extraction de blob. T_{blb} est réglé par rapport à la scène. Dans la scène du centre commercial, la taille de l'image est 240×320, T_{blb} est fixée à 50×100. Comme l'action de la population peut être représentée par la direction et et l'amplitude du mouvement, le flux optique est choisi comme étant la description de scène. L'algorithme de flux optique introduit par Sun *et al.* [Sun 2010] est utilisé dans notre travail. Il s'agit d'une méthode modifiée de la formulation de Horn et Schunck [Horn 1981] permettant une plus grande précision en utilisant des poids selon la distance spatiale, la luminosité, l'état de l'occlusion, et la médiane de filtrage.

Étape 3: La troisième étape consiste à appliquer la suppression non-maximale (NMS) algorithme [Neubeck 2006] pour sélectionner le blob B^i_{OP}. La somme des directions de tous les pixels de la blob est utilisée comme le poids des NMS.

Étape 4: La quatrième étape est l'étiquetage de la région B_{RM} restante, qui est dans le B_{FG} sauf le B^i_{OP}. Ceci consiste à traverser la région restante par un rectangle référence de taille prédéfinie, avec la même taille qu'à l'**Étape 2**. L'algorithme NMS permet de choisir le blob B^j_{RM} du blob enregistré $\{B^{j'}_{\mathrm{RM}}\}$.

Le blob plan B^k_{FG} est remplacé par le blob B^i_{OP} et la partie restante blob B^j_{RM}. Comme le montre la Fig.A.4, le rectangle 1-er dans Fig.A.4(a) est divisé en 3-ème et 4-ème rectangle en Fig.A.4(b).

A.2.3 Détection d'anomalies avec les histogrammes d'orientation du flux optique

Afin de coder les informations de mouvement dans un frame de l'image, nous avons considéré des histogrammes de l'orientation des flux optiques au niveau de plusieurs blocs qu parcourent toute l'image avec un chevauchement de plusieurs pixels. Ensuite, après normalisation, ces histogrammes sont concaténés pour former le vecteur descripteur HOFO. La Fig.A.5 illustre le calcul du descripteur HOFO de l'image originale et de l'image de premier plan. Chaque bloc est divisé en cellules où l'histogramme des orientations du flux optique est calculé.

Les procédures du calcul de HOFO dans le frame d'origine (sans soustraction de fonds) et dans le l'image de premier plan sont similaires. Le descripteur HOFO est calculé à chaque bloc, puis accumulé dans un vecteur global notée fonction F_k pour le cadre de k. Fig.A.6 et Fig.A.7 montre le calcul de HOFO. Les flux optique horizontale et verticale (u et v champs) sont répartis en 9 intervalles d'orientation, sur un horizon de 0°-360°. Le HOFO est calculé avec une proportion de recouvrement fixé à 50% de deux blocs contigus.

Un bloc contient $b^h \times b^w$ cells de $c^h \times c^w$ pixels, où b^h et b^w sont les nombre de cellules dans la direction y et x , respectivement, en coordonnées cartésiennes, c^h est la hauteur de la cellule et c^w est la largeur de la cellule. L'analyse des blocs de HOFO conjointement locales permet de considérer le comportement dans le cadre mondial. En d'autres termes, la concaténation de cellules HOFO nous permet de modéliser l'interaction entre les mouvements des blocs locaux.

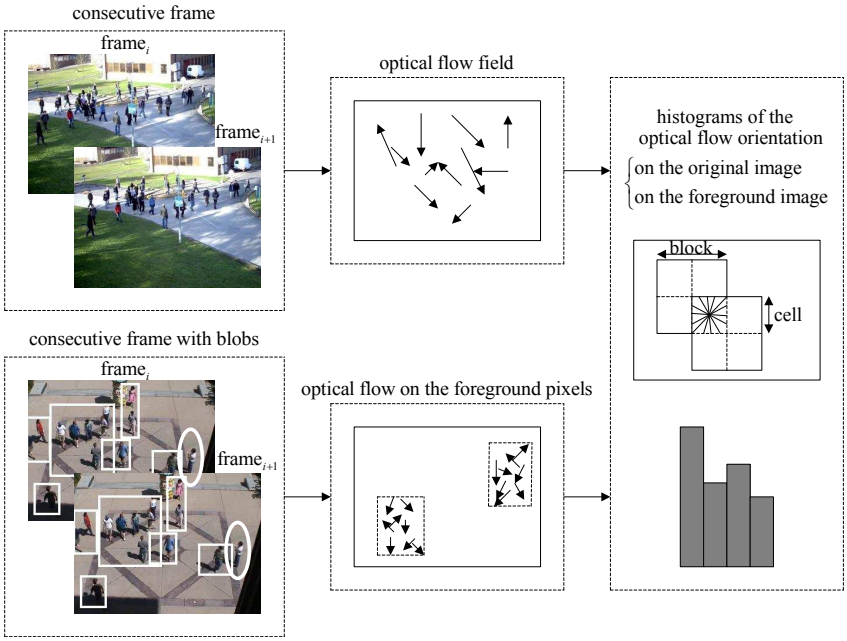


Figure A.5: Histogrammes des orientations de flux optique (HOFO) de la cadre d'origine, et de la cadre de premier plan obtenu après l'application de la soustraction du fond.

Supposons qu'un ensemble de blobs $\{B_i^{m_i'}\}$ de l'ensemble de l'image $\{I_1^{n^{\text{trn}}+n^{\text{tst}}}\}$, $1 \leq i \leq (n^{\text{trn}} + n^{\text{tst}}), 1 \leq m_i' \leq m_i$ décrivant la formation (normal) et de tester le comportement de blob (normal et anormal) de la scène donnée est disponible, n^{trn} est le nombre des cadres de formation, n^{tst} est le nombre de cadres de test, m_i est le nombre de blobs dans la carde i, m_i' est l'indice du blob, $B_i^{m_i'}$ est le m' blob dans la carde i. Le comportement du blob anormal est défini comme un événement qui s'écarte de l'ensemble des événements des blobs normaux. L'architecture générale de la détection de blobs anormaux par SVM mono-classe est expliquée ci-dessous.

Étape 1: La première étape consiste à calculer les caractéristiques de flux optique d'une image à échelle de gris.

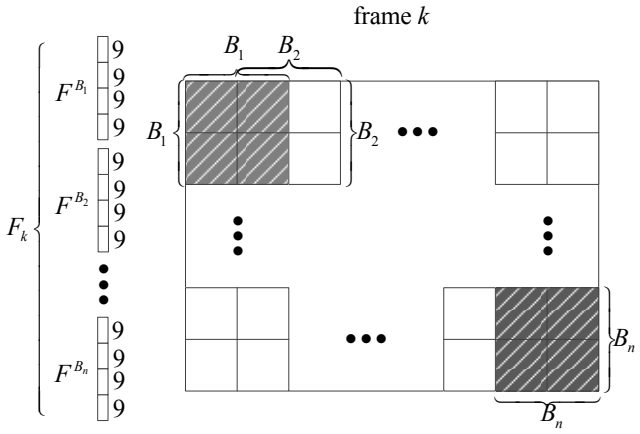


Figure A.6: Histogrammes d'orientation de flux optique (HOFO) de calcul de la k cadre.

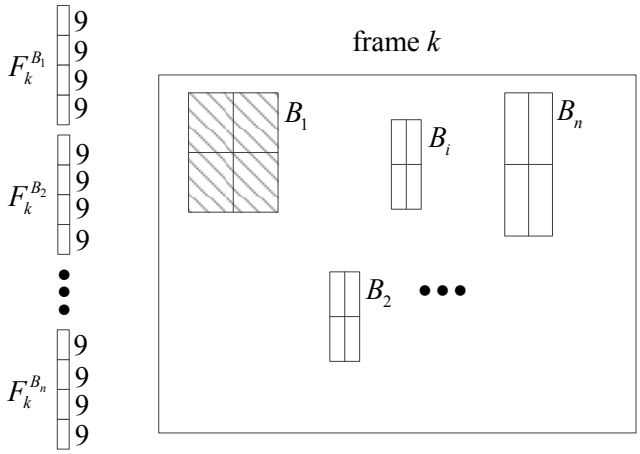


Figure A.7: Histogrammes de flux optique orientations (HOFO) calcul de la blob en la k cadre.

$$\{I_1, I_2, \ldots, I_{n^{\text{trn}}+n^{\text{tst}}}\} \tag{A.6}$$
$$\longrightarrow \{(FG_1, OP_1), \ldots, (FG_{n^{\text{trn}}+n^{\text{tst}}}, OP_{n^{\text{trn}}+n^{\text{tst}}})\} \tag{A.7}$$
$$\longrightarrow \{(B_1^1, \ldots, B_1^{m_1}), \ldots, (B_{n^{\text{trn}}+n^{\text{tst}}}^1, \ldots, B_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}})\} \tag{A.8}$$
$$\longrightarrow \{(OP_1^1, \ldots, OP_1^{m_1}), (OP_2^1, \ldots, OP_2^{m_2}), \ldots, (OP_{n^{\text{trn}}+n^{\text{tst}}}^1, \ldots, OP_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}})\}, \tag{A.9}$$

où I_i est le cadre de i, (FG_i, OP_i) sont l'image de premier plan et flux optique de la cadre i, $\{B_i^1, B_i^2, \ldots, B_i^{m_i}\}$ sont les 1 au m blobs dans le cadre de i, m_i est le nombre des blobs, $\{OP_i^1, \ldots, OP_i^{m_i}\}$ sont le flux optique correspondant des blobs.

Étape 2: La deuxième étape est le calcul de la fonction de matrice de covariance des blobs.

$$\begin{aligned}&\{(OP_1^1, B_1^1, \ldots, OP_1^{m_1}, B_1^{m_1}), \ldots, (OP_{n^{\text{trn}}+n^{\text{tst}}}^1, B_{n^{\text{trn}}+n^{\text{tst}}}^1, \ldots, OP_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}}, B_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}})\}\\ &\longrightarrow \{(HOFO_1^1, \ldots, HOFO_1^{m_1}), \ldots, (HOFO_{n^{\text{trn}}+n^{\text{tst}}}^1, \ldots, HOFO_{n^{\text{trn}}+n^{\text{tst}}}^{m_{n^{\text{trn}}+n^{\text{tst}}}})\},\end{aligned} \tag{A.10}$$

où $\{HOFO_i^1, \ldots, HOFO_i^{m_i}\}$ sont les matrices de covariance descripteur correspondant des blobs dans le cadre de *i*.

Étape 3: La troisième étape est l'application SVM une classe sur les descripteurs extraits de la formation des taches normaux pour obtenir les vecteurs de support.

$$\begin{aligned}&\{(HOFO_1^1 \ldots HOFO_1^{m_1}), \ldots, (HOFO_{n^{\text{trn}}}^1 \ldots HOFO_{n^{\text{trn}}}^{m_{n^{\text{trn}}}})\}\\ &\xrightarrow{SVM} \text{support vector } \{Sp_1, Sp_2, \ldots, Sp_o\},\end{aligned} \tag{A.11}$$

où $\{(HOFO_1^1 \ldots HOFO_1^{m_1}), \ldots, (HOFO_{n^{\text{trn}}}^1 \ldots HOFO_{n^{\text{trn}}}^{m_{n^{\text{trn}}}})\}$ sont les descripteurs de HOFO des blobs.

Étape 4: Sur la base des vecteurs de support obtenus à partir des blobs de formation, un échantillon de blob entrant $HOFO_l^{m'_l}$ est classé.

$$f(HOFO_l^{m'_l}) = \mathbf{sgn}(\sum_{i=1}^{o} \alpha_i \kappa(Sp_i, HOFO_l^{m'_l}) - \rho) \tag{A.12}$$

$$= \begin{cases} 1 & \text{if } f(HOFO_l^{m'_l}) \geq 0 \\ -1 & \text{if } f(HOFO_l^{m'_l}) < 0, \end{cases} \tag{A.13}$$

où $HOFO_l^{m'_l}$ est le descripteur de la HOFO du blob m'_l dans le cadre *l*. "1" correspond à la tache normale, "-1" correspond à la tache anormale.

Pour la détection des événements anormaux, la condition préalable d'un événement peut être défini comme normal ou anormal, c'est qu'il se produit pendant plusieurs cadres consécutifs. En d'autres termes, l'événement normal ou anormal n'est pas ponctuel. Sur cette base, la courte séquence d'événements anormaux qui se produit par intermittence à quelques images de la séquence vidéo normale pourrait être modifiée à l'état normal. De même, les événements cadres normaux qui sont détectés parmi la longue séquence d'images anormaux pourraient être modifiés pour anormal. Un seuil *N* du nombre de cadres d'image est prédéfinie, le post traitement des résultats de la détection est illustré sur la Fig.A.8. Si le nombre d'états anormaux (résultats négatifs prévus) dépasse le seuil *N* dans les états normaux (résultats positifs prévus), puis les étiquettes de prédiction normaux sont convertis en anormal.

A.3 Algorithmes de détection en ligne à base de SVM mono-classe

Avant de présenter nos contributions dans les aspects algorithmiques de détection en ligne, on introduit dans la suite le descripteur de covariance qui permet de fusionner plusieurs caractéristiques locales de l¡¯image d¡¯une manière efficace.

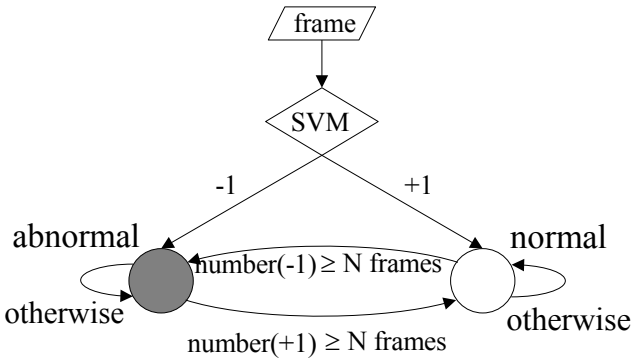


Figure A.8: Modèle de transition d'état. N est le seuil prédéterminé pour ajuster le résultat de détection.

La matrice de covariance est proposée par O. Tuzel [Tuzel 2006] pour décrire blob caractéristiques d'image de gris ou couleur. Il a été utilisé avec succès dans le problème de détection d'objet [Tuzel 2007, Tuzel 2008], le problème de la reconnaissance de visage [Pang 2008], et le problème de cheminement [Porikli 2006c]. Le descripteur de covariance est robuste contre le bruit, les distorsions d'éclairage, et la rotation [Porikli 2006a]. Nous proposons de construire la matrice de covariance en se basant sur le flux optique et l'intensité de mouvement pour coder des caractéristiques à la fois d'une blob et d'une image globale. Le descripteur de covariance est calculée en tant que:

$$F(x, y, \ell) = \phi_\ell(I, x, y), \tag{A.14}$$

où I est une image (qui peut être gris, rouge-vert-bleu (RVB), etc.), F est un $W \times H \times d$ fonction dimensions de l'image I, W est la largeur de l'image, H est la hauteur de l'image, d est le nombre de fonctions utilisées, ϕ_ℓ est une application concernant l'image avec la fonction de ℓ de l'image I. Pour une région donnée R rectangulaire, les points caractéristiques peuvent être représentés comme $d \times d$ matrice de covariance :

$$\boldsymbol{C}_R = \frac{1}{n-1} \sum_{k=1}^{n_p} (\boldsymbol{z}_k - \boldsymbol{\mu})(\boldsymbol{z}_k - \boldsymbol{\mu})^\top, \tag{A.15}$$

où $\boldsymbol{\mu}$ est la moyenne des points, $\boldsymbol{C}_R$ est la matrice de covariance de la fonction F, $\boldsymbol{z}_k$ est le vecteur d'éléments de pixel k, n_p pixels sont choisis. Les éléments diagonaux de la matrice de covariance représentent la variance de chaque caractéristique, les entrées de la matrice de repos indiquent la relation entre des caractéristiques différentes. Le $\boldsymbol{C}_R$ de covariance d'une région donnée R ne dispose pas d'information concernant l'ordre et le nombre de points.

Basé sur le flux optique et l'intensité, 13 vecteurs de caractéristiques différentes F indiquées dans le Table A.1 sont proposés pour construire le descripteur de covariance. I est l' intensité de l'image gris, le flux optique est obtenue à partir de l'image gris, u est le flux optique horizontale, v est le débit optique vertical; I_x, u_x, v_x et I_y, u_y v_y sont les dérivés premiers de l'intensité, le flux horizontal optique et flot optique vertical dans la direction

x et la direction y; I_{xx}, u_{xx}, v_{xx} et I_{yy}, u_{yy}, v_{yy} sont les dérivées secondes des fonctions correspondantes dans la direction x et la direction y; I_{xy}, u_{xy} et v_{xy} sont les dérivées secondes dans la direction y des dérivées premières dans la direction x des fonctions correspondantes. Fig.A.9 illustre la fonction de matrice de covariance des blobs, pour le blob de k dans i cadre B_i^k, fonction de la matrice de covariance est $\boldsymbol{C}_i^k$. Le flux optique montre l'information inter-cadre, il décrit les informations de mouvement. L'intensité montre l'information intra-cadre, il encode les informations de l'apparence. Si la cadre entière est prise comme une grosse blob, la matrice de covariance descripteur de i cadre est $\boldsymbol{C}_i$.

	Feature Vector F	
flux optique	$F_1(4\times4)$	$[y\ x\ u\ v\ ]$
	$F_2(6\times6)$	$[y\ x\ u\ v\ u_x\ u_y]$
	$F_3(6\times6)$	$[y\ x\ u\ v\ v_x\ v_y]$
	$F_4(8\times8)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y]$
	$F_5(12\times12)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}]$
	$F_6(14\times14)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ u_{xy}\ v_{xy}]$
flux optique et intensité	$F_7(5\times5)$	$[y\ x\ u\ v\ I]$
	$F_8(9\times9)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ I]$
	$F_9(13\times13)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ I]$
	$F_{10}(15\times15)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ u_{xy}\ v_{xy}\ I]$
	$F_{11}(11\times11)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ I\ I_x\ I_y]$
	$F_{12}(17\times17)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ I\ I_x\ I_y\ I_{xx}\ I_{yy}]$
	$F_{13}(20\times20)$	$[y\ x\ u\ v\ u_x\ u_y\ v_x\ v_y\ u_{xx}\ u_{yy}\ v_{xx}\ v_{yy}\ u_{xy}\ v_{xy}\ I\ I_x\ I_y\ I_{xx}\ I_{yy}\ I_{xy}]$

Table A.1: Caractéristiques F utilisée pour former les matrices de covariance.

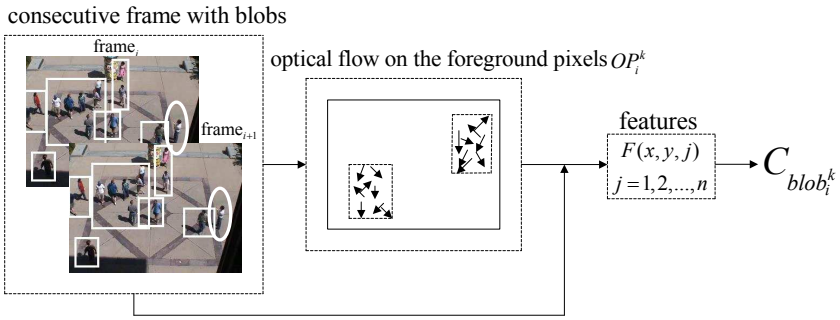


Figure A.9: Calcul du descripteur matrice de covariance (COV) de la blob.

La matrice de covariance est un élément d'un groupe de Lie G, où la mesure de la distance de deux éléments est définie par:

$$d(\boldsymbol{X}_1, \boldsymbol{X}_2) = \| \log(\boldsymbol{X}_1^{-1}\boldsymbol{X}_2)\|, \tag{A.16}$$

$$\text{with } \|\boldsymbol{A}\| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}, \tag{A.17}$$

où $\|\cdot\|$ est la norme de Frobenius, a_{ij} est un élément de la matrice $\boldsymbol{A}$, $\boldsymbol{X}_i$ et $\boldsymbol{X}_j$ sont les matrices dans un groupe de Lie G. Ainsi, le noyau gaussien dans un groupe de Lie G est:

$$\kappa(\boldsymbol{X}_i, \boldsymbol{X}_j) = \exp(-\frac{\| \log(\boldsymbol{X}_i^{-1}\boldsymbol{X}_j)\|}{2\sigma^2}), \quad (\boldsymbol{X}_i, \boldsymbol{X}_j) \in G \times G. \tag{A.18}$$

En utilisant la formule Baker Campbell Hausdorff [Hall 2003] séparé dans la théorie de groupe de Lie, le noyau est:

$$\kappa(\boldsymbol{X}_i, \boldsymbol{X}_j) = \exp(-\frac{\| \log(\boldsymbol{X}_i) - \log(\boldsymbol{X}_j)\|^2}{2\sigma^2}), \quad (\boldsymbol{X}_i, \boldsymbol{X}_j) \in G \times G, \tag{A.19}$$

$$\kappa(\boldsymbol{X}_i, \boldsymbol{X}_j) = \exp(-\frac{\|\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}_j\|^2}{2\sigma^2}), \tag{A.20}$$

où $\overline{\boldsymbol{x}}_i$ est le vecteur construit par des éléments de la triangulaire supérieure et les éléments diagonaux de la matrice de $\log(\boldsymbol{X})$.

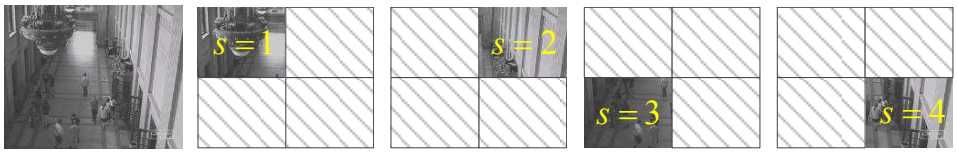
Pour construire un élément descripteur plus représentatif et discriminatoire, nous nous sommes séparés de chaque frame en m parties. La stratégie multi-noyau de notre descripteur de matrice de covariance est définie par [Noumir 2012a, Rakotomamonjy 2008, Chen 2013]:

$$\kappa(\boldsymbol{X}_i, \boldsymbol{X}_j) = \sum_{s=1}^{m} \mu_s \kappa_s(\overline{\boldsymbol{x}}_i, \overline{\boldsymbol{x}}_j). \tag{A.21}$$

Eq.(A.21) est un noyau constitué de m noyaux de base. Parce que chaque noyau de base remplit la Mercer condition, leur somme est aussi un noyau définie semi-positive en vertu de l'état de μ_s non-négatifs. Dans cette expression, le noyau gaussien est adopté avec:

$$\kappa_s(\overline{\boldsymbol{x}}_i, \overline{\boldsymbol{x}}_j) = \exp(-\frac{\|\overline{\boldsymbol{x}}_i - \overline{\boldsymbol{x}}_j\|^2_{[s]}}{2\sigma^2}). \tag{A.22}$$

Les noyaux $\kappa_s, s = 1, \cdots, m$ sont des gaussiennes. Chaque vecteur $\overline{\boldsymbol{x}}$ de l'échantillon se compose de m parties $[\overline{\boldsymbol{x}}_1, \overline{\boldsymbol{x}}_2, \ldots, \overline{\boldsymbol{x}}_m]$. Cette stratégie de noyau est similaire à la frame d'un filtrage en utilisant un masque. Par exemple, une frame est divisée en quatre parties, comme le montre dans la Fig.A.10. Si $s = 1$, la partie gauche vers le haut de l'image est sélectionnée. Nous présélectionner le poids μ_s according à la caractéristique de l'image pour régler l'importance de chaque sous-image. Dans la scène intérieure, dans les images normaux et les images anormaux, il n'y a personne dans la moitié supérieure de l'image. Ainsi, nous avons mis en $\mu_{1,2} = 0.1$, $\mu_{3,4} = 0.4$ à réduire l'importance de la sous-image où $s = 1$ et $s = 2$. Dans ce cas, le noyau résultant appartient à l'enveloppe convexe des quatre



(a) Image (b) $S = 1$ (c) $S = 2$ (d) $S = 3$ (e) $S = 4$

Figure A.10: Filtrer l'image par le masque pour sélectionner une sous-image. (a) Une image original de la scène intérieure. (b) $S = 1$, $\mu_1 = 0.1$, la partie gauche supérieure de l'image est sélectionné. (c) $S = 2$, $\mu_2 = 0.1$, la partie supérieure droite. (d) $S = 3$, $\mu_3 = 0.4$, la partie gauche inférieure. (e) $S = 4$, $\mu_4 = 0.4$, la partie inférieure droite.

noyaux considérés. En considérant cette combinaison, le noyau résultant exécute chaque κ_s du noyau individuellement.

Dans les problèmes de détection d'événements anormaux, les échantillons d'apprentissage peuvent durer une longue période de temps. L'algorithme SVM est généralement appliqué en batch, c'est à dire, où toutes les données de formation sont donnés a priori. Si des données de formation supplémentaires arrivent après, le SVM doit être recalculé. Dans le problème de la détection des événements anormaux pour la surveillance vidéo, la séquence normale pour la formation peut durer pendant une longue période. Il est impossible de former la grande série d'échantillons normaux. En outre, si une nouvelle donnée est ajoutée à un grand ensemble, il n'aura probablement qu'un effet minime sur la surface de la décision précédente. Compte tenu de ces deux aspects, la stratégie en ligne est adoptée dans notre travail pour s'adapter aux exigences de calcul et de mémoire.

A.3.1 Détection anormale en ligne via le soutien vecteur de description de données

La méthode de description de données de vecteurs de support (SVDD) calcule une forme de sphère décision frontière avec le volume minimal autour d'un ensemble d'objets. Le centre de la sphère $\boldsymbol{c}$ et rayon R sont à déterminer par l'intermédiaire du problème d'optimisation suivant:

$$\min_{R,\xi,c} \; R^2 + C\sum_{i=1}^{n}\xi_i, \tag{A.23}$$

$$\text{subject to: } \|\Phi(\boldsymbol{x}_i) - \boldsymbol{c}\|^2 \leq R^2 + \xi_i, \;\; \xi_i \geq 0, \forall i, \tag{A.24}$$

où n est le nombre d'échantillons de formation, ξ_i est une variable utilisée pour pénaliser les valeurs aberrantes. Le hyperparamètre C est le poids pour retenir variables d'écart, il règle le nombre de valeurs aberrantes acceptables. La fonction non-linéaire $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ cartographie un $\boldsymbol{x}_i$ de référence dans dans la fonction espace $\mathcal{H}$, il permet de résoudre un problème de classification non linéaire par la conception d'un classificateur linéaire dans l'espace des fonctions $\mathcal{H}$. κ est la fonction du noyau de calcul de produits scalaires dans

$\mathcal{H}$, $\kappa(\boldsymbol{x}, \boldsymbol{x}') = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{x}') \rangle$. En introduisant des multiplicateurs de Lagrange, le problème dual (A.24) est écrit par le problème d'optimisation quadratique suivante:

$$\max_{\alpha} \quad \sum_{i=1}^{n} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{A.25}$$

$$\text{subject to: } 0 \leq \alpha_i \leq C, \ \sum_{i=1}^{n} \alpha_i = 1, \ \boldsymbol{c} = \sum_{i=1}^{n} \alpha_i \Phi(\boldsymbol{x}_i). \tag{A.26}$$

La fonction de décision est:

$$f(\boldsymbol{x}) = \textbf{sgn}(R^2 - \sum_{i,j=1}^{n} \alpha_i \alpha_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) + 2 \sum_{i=1}^{n} \alpha_i \kappa(\boldsymbol{x}_i, \boldsymbol{x}) - \kappa(\boldsymbol{x}, \boldsymbol{x})). \tag{A.27}$$

Pour les grandes données de formation, la solution ne peut être obtenue facilement, une stratégie en ligne pour former les données est utilisée dans notre travail. Laissez $\boldsymbol{c}_{\mathfrak{D}}$ désigne un modèle rare du centre $\boldsymbol{c}_n = \frac{1}{n} \sum_{i=1}^{n} \Phi(\boldsymbol{x}_i)$ l'aide d'un petit sous-ensemble d'échantillons disponibles qui appelle dictionnaire:

$$\boldsymbol{c}_{\mathfrak{D}} = \sum_{i \in \mathfrak{D}} \alpha_i \Phi(\boldsymbol{x}_i), \tag{A.28}$$

où $\mathfrak{D} \subset \{1, 2, \ldots, n\}$, et laissez $N_{\mathfrak{D}}$ désigne le cardinal de ce sous-ensemble $\boldsymbol{x}_{\mathfrak{D}}$.

La distance d'un tracé de référence $\Phi(\boldsymbol{x})$ par rapport au centre $\boldsymbol{c}_{\mathfrak{D}}$ peut être calculée par:

$$\|\Phi(\boldsymbol{x}) - \boldsymbol{c}_{\mathfrak{D}}\| = \sum_{i,j \in \mathfrak{D}} \alpha_i \alpha_j \, \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) - 2 \sum_{i \in \mathfrak{D}} \alpha_i \, \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + \kappa(\boldsymbol{x}, \boldsymbol{x}). \tag{A.29}$$

Une modification de la formulation initiale de l'algorithme de classification une classe consiste à minimiser l'erreur d'approximation $\|\boldsymbol{c}_n - \boldsymbol{c}_{\mathfrak{D}}\|$ est [Noumir 2012c, Noumir 2012b]:

$$\boldsymbol{\alpha} = \arg \min_{\alpha_i, i \in \mathfrak{D}} \|\frac{1}{n} \sum_{i=1}^{n} \Phi(\boldsymbol{x}_i) - \sum_{i \in \mathfrak{D}} \alpha_i \Phi(\boldsymbol{x}_i)\|^2. \tag{A.30}$$

La solution finale est donnée par:

$$\boldsymbol{\alpha} = \boldsymbol{K}^{-1} \boldsymbol{\kappa}, \tag{A.31}$$

où $\boldsymbol{K}$ est la matrice de Gram avec (i, j)-ième entrée $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, et $\boldsymbol{\kappa}$ est le vecteur de colonne dont les entrées $\frac{1}{n} \sum_{i=1}^{n} \kappa(\boldsymbol{x}_k, \boldsymbol{x}_i)$, $k \in \mathfrak{D}$.

Dans le schéma en ligne, à chaque pas de temps, il y a un nouvel échantillon. Laissez $\boldsymbol{\alpha}_n$ désigne les coefficients, $\boldsymbol{K}_n$ représentent la matrice de Gram, et $\boldsymbol{\kappa}_n$ représentent le vecteur, au moment de l'étape n. Un critère est utilisé pour déterminer si le nouvel échantillon peut être inclus dans le dictionnaire. Un seuil μ_0 est prédéfini, pour la $\boldsymbol{x}_t$ de référence au temps t, le critère de base de la cohérence sparsification est [Honeine 2012, Richard 2009]:

$$\varepsilon_t = \max_{i \in \mathfrak{D}} |\kappa(\boldsymbol{x}_t, \boldsymbol{x}_{w_i})|, \tag{A.32}$$

Premier cas: $\varepsilon_t > \mu_0$

Dans ce cas, la nouvelle donnée $\Phi(\boldsymbol{x}_{n+1})$ est incluse dans le dictionnaire $\mathfrak{D}$:

$$\boldsymbol{\kappa}_{n+1} = \frac{1}{n+1}(n\boldsymbol{\kappa}_n + \boldsymbol{b}) \tag{A.33}$$

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{K}_{n+1}^{-1}\boldsymbol{\kappa}_{n+1} = \frac{n}{n+1}\boldsymbol{\alpha}_n + \frac{1}{n+1}\boldsymbol{K}_n^{-1}\boldsymbol{b}. \tag{A.34}$$

où $\boldsymbol{b}$ est le vecteur de colonne dont les entrées $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{n+1})$.

Deuxième cas: $\varepsilon_t \leq \mu_0$

Dans ce cas, la nouvelle donné $\Phi(\boldsymbol{x}_{n+1})$ est inclus dans le dictionnaire $\mathfrak{D}$. La matrice de Gram $\boldsymbol{K}$ change:

$$\boldsymbol{K}_{n+1} = \begin{bmatrix} \boldsymbol{K}_n & \boldsymbol{b} \\ \boldsymbol{b}^\top & \boldsymbol{\kappa}(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1}) \end{bmatrix}. \tag{A.35}$$

En utilisant la matrice d'identité de Woodbury:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U\left(C^{-1} + VA^{-1}U\right)^{-1}VA^{-1}, \tag{A.36}$$

$\boldsymbol{K}_{n+1}^{-1}$ peut être calculée de manière itérative:

$$\boldsymbol{K}_{n+1}^{-1} = \begin{bmatrix} \boldsymbol{K}_n^{-1} & \boldsymbol{0} \\ \boldsymbol{0}^\top & 0 \end{bmatrix} + \frac{1}{\kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1}) - \boldsymbol{b}^\top \boldsymbol{K}_n^{-1}\boldsymbol{b}} \times \begin{bmatrix} -\boldsymbol{K}_n^{-1}\boldsymbol{b} \\ 1 \end{bmatrix} \times \begin{bmatrix} -\boldsymbol{b}^\top \boldsymbol{K}_n^{-1} & 1 \end{bmatrix}. \tag{A.37}$$

Le vecteur $\boldsymbol{\kappa}_{n+1}$ est mis à jour à partir de $\boldsymbol{\kappa}_n$,

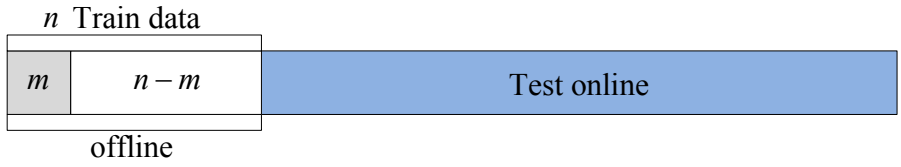
$$\boldsymbol{\kappa}_{n+1} = \frac{1}{n+1}\begin{bmatrix} n\boldsymbol{\kappa}_n + \vec{b} \\ \kappa_{n+1} \end{bmatrix}, \tag{A.38}$$

$$\text{avec } \kappa_{n+1} = \sum_{i=1}^{n+1} \kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_i). \tag{A.39}$$

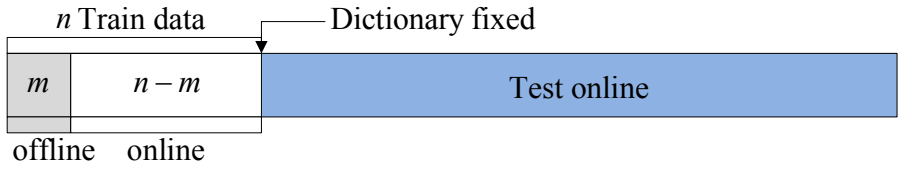
$$\begin{aligned} \boldsymbol{\alpha}_{n+1} =& \frac{1}{n+1}\begin{bmatrix} n\boldsymbol{\alpha}_n + \boldsymbol{K}_n^{-1}\boldsymbol{b} \\ 0 \end{bmatrix} \\ & - \frac{1}{(n+1)(\kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1}) - \boldsymbol{b}^\top \boldsymbol{K}_n^{-1}\boldsymbol{b})} \\ & \times \begin{bmatrix} \boldsymbol{K}_n^{-1}\boldsymbol{b} \\ 1 \end{bmatrix}\left(n\boldsymbol{b}^\top\boldsymbol{\alpha}_n + \boldsymbol{b}^\top\boldsymbol{K}_n^{-1}\boldsymbol{b} - \kappa_{n+1}\right). \end{aligned} \tag{A.40}$$

Dans un problème de détection d'événements anormaux, il est supposé qu'une série de frames de formation $\{I_1, \ldots, I_n\}$ (la classe positive) décrivant le comportement normal est obtenu. Les architectures générales de détection anormale sont introduites ci-dessous.

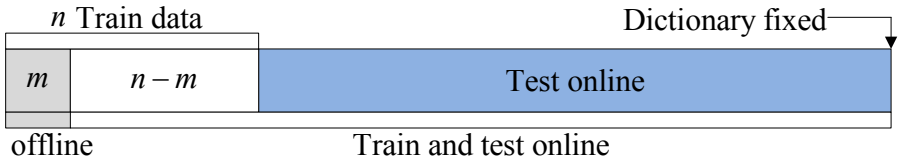
Nous proposons deux stratégies de détection anormaux, la différence entre ces deux stratégies est le temps lorsque le dictionnaire est fixe. Ces deux stratégies sont représentées sur la Fig.A.11(b) et (c). **Stratégie 1** est représentée sur la Fig.A.11(b). Les données d'apprentissage sont tirés un par un. Lorsque la période de formation est terminée, le dictionnaire et le classificateur sont fixés. Chaque donnée de test est classée selon le dictionnaire. Fig.A.11(c) illustrates **Stratégie 2**. La procédure de formation est aussi la même que la **Stratégie 1**. Mais dans la période d'essai, le dictionnaire est mis à jour si la donnée x_i satisfait à la condition de mise à jour du dictionnaire.



(a) Stratégie offline



(b) Stratégie 1



(c) Stratégie 2

Figure A.11: Hors ligne et deux stratégies de détection d'événements anormaux en ligne basés sur la description des données de vecteur de support en ligne (SVDD). (a) Stratégie hors ligne. Les données sont tirées de formation comme un hors-ligne de lot. (b) Stratégie 1. Le dictionnaire est fixé quand toutes les données d'entraînement sont apprises. (c) Stratégie 2. Le dictionnaire continue à être mis à jour pendant la période d'essai.

A.3.2 Détection anormale en ligne par des moindres carrés SVM mono-classe

Nous proposons une nouvelle méthode de classification en ligne par moindres carrés (LS-OC-SVM). Le LS-OC-SVM extrait un hyperplan comme une description optimale des objets de formation dans un sens des moindres carrés régularisés. La ligne LS-OC-SVM apprend tout d'abord à partir d'un ensemble d'apprentissage avec le nombre limite d'échantillons à fournir un modèle normal de base, puis met à jour le modèle à travers les données restantes. Dans le schéma en ligne, la complexité du modèle est commandée par le critère de cohérence. Et puis, la ligne LS-OC-SVM est adoptée pour traiter le problème de la détection d'événements anormaux.

A.3.2.1 SVM mono-classe moindres carrés

LS-OC-SVM extrait un hyperplan comme une description optimale des objets de formation dans un sens des moindres carrés régularisés. Il peut être écrit comme la fonction objective qui suit:

$$\begin{aligned} &\min_{\boldsymbol{w},\xi,\rho} \ \frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + \frac{1}{2}C\sum_{i=1}^{n}\xi_i^2 \\ &\text{sujet à: } \langle \boldsymbol{w}, \Phi(\boldsymbol{x}_i)\rangle = \rho - \xi_i. \end{aligned} \tag{A.41}$$

Le Lagrange associé est:

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}^{n}\alpha_i\left(\boldsymbol{w}^\top\Phi(\boldsymbol{x}_i) - \rho + \xi_i\right). \tag{A.42}$$

En dérivant par rapport aux variables primales :

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0 \qquad \Rightarrow \boldsymbol{w} = \sum_{i=1}^{n}\alpha_i\Phi(\boldsymbol{x}_i), \tag{A.43}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \qquad \Rightarrow C\xi_i = \alpha_i, \tag{A.44}$$

$$\frac{\partial L}{\partial \rho} = 0 \qquad \Rightarrow \sum_{i=1}^{n}\alpha_i = 1, \tag{A.45}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \qquad \Rightarrow \boldsymbol{w}^\top\Phi(\boldsymbol{x}_i) + \xi_i - \rho = 0. \tag{A.46}$$

On a:

$$\sum_{i,j=1}^{n}\alpha_i\Phi^\top(\boldsymbol{x}_i)\Phi(\boldsymbol{x}_j) + \frac{\alpha_i}{C} - \rho = 0. \tag{A.47}$$

$$\begin{bmatrix} \boldsymbol{K} + \frac{\boldsymbol{I}}{C} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ -\rho \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}, \tag{A.48}$$

L'hyperplan est alors décrit par:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n}\alpha_i\kappa(\boldsymbol{x_i}, \boldsymbol{x}) - \rho = 0. \tag{A.49}$$

La distance, $dis(\boldsymbol{x})$, d'une donnée, $\boldsymbol{x}$, par rapport à l'hyperplan est calculée par:

$$dis(\boldsymbol{x}) = \frac{|f(\boldsymbol{x})|}{\|\boldsymbol{\alpha}\|} = \frac{|(\sum_{i=1}^{n}\alpha_i\kappa(\boldsymbol{x}_i, \boldsymbol{x}) - \rho)|}{\|\boldsymbol{\alpha}\|}. \tag{A.50}$$

A.3.2.2 En ligne des moindres carrés SVM mono-classe

Dans un régime d'apprentissage en ligne, les données de formation arrivent en permanence. Nous devons donc accorder hyper paramètres dans la fonction objective et la classe de l'hypothèse d'une manière en ligne [Diehl 2003]. Laissez $\boldsymbol{\alpha}_n$, $\boldsymbol{K}_n$ et $\boldsymbol{I}_n$ désignent le coefficient, matrice de Gram et la matrice d'identité à l'étape de temps, n, respectivement. Les paramètres de LS-OC-SVM $[\boldsymbol{\alpha}_n \ -\rho_n]^\top$ à l'étape de temps, n, peuvent être calculés comme suit:

$$\begin{bmatrix} \boldsymbol{\alpha}_n \\ -\rho_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C} & \mathbf{1}_n \\ \mathbf{1}_n^\top & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0}_n \\ 1 \end{bmatrix}, \tag{A.51}$$

Afin de procéder, rappeler la matrice inverse identité pour les matrices A, B, C et D de dimensions adaptées: [Honeine 2012]:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -A^{-1}B \\ 1 \end{bmatrix} \times (D - CA^{-1}B)^{-1} \times [-CA^{-1} \quad 1]. \tag{A.52}$$

La matrice, $\boldsymbol{K}_n$, à chargement diagonale $\frac{\boldsymbol{I}_n}{C}$ peut être calculée de façon récursive par rapport au temps de l'étape n par:

$$\left[\boldsymbol{K}_{n+1} + \frac{\boldsymbol{I}_{n+1}}{C}\right]^{-1} \tag{A.53}$$

$$= \left[\begin{array}{c|c} \boldsymbol{K}_n + \frac{\boldsymbol{I}}{C} & \boldsymbol{\kappa}_{n+1} \\ \hline \boldsymbol{\kappa}_{n+1} & \kappa_{n+1} + \frac{1}{C} \end{array}\right]^{-1} \tag{A.54}$$

$$= \begin{bmatrix} \left(\boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C}\right)^{-1} & \mathbf{0}_n \\ \mathbf{0}_n^\top & 0 \end{bmatrix} + \frac{1}{\left(\kappa_{n+1} + \frac{1}{C}\right) - \boldsymbol{\kappa}_{n+1}\left(\boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C}\right)^{-1}\boldsymbol{\kappa}_{n+1}}$$

$$\begin{bmatrix} -\left(\boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C}\right)^{-1}\boldsymbol{\kappa}_{n+1} \\ 1 \end{bmatrix} \begin{bmatrix} -\boldsymbol{\kappa}_{n+1}^\top \left(\boldsymbol{K}_n + \frac{\boldsymbol{I}_n}{C}\right)^{-1} & 1 \end{bmatrix}, \tag{A.55}$$

où $\boldsymbol{\kappa}_{n+1}$ est le vecteur colonne avec i-ième entry $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_{n+1})$, $i \in \{1, 2, \ldots, n\}$, et $\kappa_{n+1} = \kappa(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1})$.

A.3.2.3 Sparse en ligne LS-OC-SVM

Nous approchons avec ces éléments de dictionnaire D:

$$\boldsymbol{w} = \sum_{j=1}^{D} \beta_j \Phi(\boldsymbol{x}_{w_j}). \tag{A.56}$$

L'hyperplan devient:

$$f(\boldsymbol{x}) = \sum_{j=1}^{D} \beta_j \kappa(\boldsymbol{x}, \boldsymbol{x}_{w_j}) - \rho = 0. \tag{A.57}$$

La distance, $dis_{\mathfrak{D}}(\boldsymbol{x})$, devient:

$$dis_{\mathfrak{D}}(\boldsymbol{x}) = \frac{|\sum_{j=1}^{D} \beta_i \kappa(\boldsymbol{x}, \boldsymbol{x}_{w_j}) - \rho|}{\|\boldsymbol{\beta}\|}, \tag{A.58}$$

La fonction de Lagrange est:

$$L = \frac{1}{2}\boldsymbol{\beta}^\top K_{\mathfrak{D}}\boldsymbol{\beta} - \rho + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}^{n}\alpha_i(\sum_{j=1}^{D}\beta_j\Phi^\top(\boldsymbol{x}_{w_j})\Phi(\boldsymbol{x}_i) + \xi_i - \rho). \tag{A.59}$$

En annulant les dérivées de la fonction de Lagrange (A.59) par rapport aux variables primaires,

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = 0 \qquad \Rightarrow K_D\boldsymbol{\beta} = K_D^\top(\boldsymbol{x})\boldsymbol{\alpha}, \tag{A.60}$$

$$\frac{\partial L}{\partial \xi_i} = 0 \qquad \Rightarrow C\xi_i = \alpha_i, \tag{A.61}$$

$$\frac{\partial L}{\partial \rho} = 0 \qquad \Rightarrow \sum_{i=1}^{n}\alpha_i = 1, \tag{A.62}$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \qquad \Rightarrow \sum_{j=1}^{D}\beta_j.\kappa(\boldsymbol{x}_{w_j}, \boldsymbol{x}_i) + \xi_i - \rho = 0 \tag{A.63}$$

On a:

$$\begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}^\top(\boldsymbol{x}) + \frac{\boldsymbol{I}}{C} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ -\rho \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}. \tag{A.64}$$

Premier cas: $\varepsilon_t > \mu_0$

Dans ce cas, au moment de l'étape $n+1$, les nouvelles données, $\boldsymbol{x}_{n+1}$, n'est pas inclus dans le dictionnaire. La matrice de Gram, $\boldsymbol{K}_{\mathfrak{D}}$, avec les entrées, $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i, j \in \{1, 2, \ldots, D\}$, est inchangée. Quand un nouvel échantillon, $\boldsymbol{x}$, arrive, nous devons calculer:

$$\left[\begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x}) \\ \boldsymbol{\kappa}^\top \end{bmatrix}\boldsymbol{K}_{\mathfrak{D}}^{-1}\begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})^\top & \boldsymbol{\kappa} \end{bmatrix} + \frac{\boldsymbol{I}}{C}\right]^{-1} = \begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}^\top(\boldsymbol{x}) + \frac{\boldsymbol{I}}{C} & \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{\kappa} \\ \boldsymbol{\kappa}^\top\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}^\top(\boldsymbol{x}) & \boldsymbol{\kappa}^\top\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{\kappa} + \frac{\boldsymbol{I}}{C} \end{bmatrix}^{-1}. \tag{A.65}$$

Deuxième cas: $\varepsilon_t \leq \mu_0$

Dans ce cas, les nouvelles données, $\boldsymbol{x}_{n+1}$, est ajouté dans le dictionnaire, $\boldsymbol{x}_{\mathfrak{D}}$. Ensuite, la matrice de Gram doit être changé par:

$$\overline{\boldsymbol{K}}_{\mathfrak{D}} = \begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}} & \boldsymbol{d} \\ \boldsymbol{d}^\top & d \end{bmatrix}, \tag{A.66}$$

Après quelques manipulations algébriques, nous avons:

$$\overline{\boldsymbol{K}}_{\mathfrak{D}}^{-1} = \begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}}^{-1} + \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^{\top} & c \end{bmatrix}, \tag{A.67}$$

où:

$$c = \frac{1}{d - \boldsymbol{d}^{\top}\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{d}}, \tag{A.68}$$

$$\boldsymbol{A} = c\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{d}\boldsymbol{d}^{\top}\boldsymbol{K}_{\mathfrak{D}}^{-1}, \tag{A.69}$$

$$\boldsymbol{b} = -c\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{d}. \tag{A.70}$$

Soit S la mise à jour $\left[\boldsymbol{K}_D(\boldsymbol{x})\boldsymbol{K}_D^{-1}\boldsymbol{K}_{\mathcal{D}}^{\top}(\boldsymbol{x}) + \frac{\boldsymbol{I}}{C}\right]^{-1}$ nous avons alors:

$$S = \left[\begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x}) & \boldsymbol{q} \end{bmatrix} \overline{\boldsymbol{K}}_{\mathfrak{D}}^{-1} \begin{bmatrix} \boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) \\ \boldsymbol{q}^{\top} \end{bmatrix} + \frac{\boldsymbol{I}}{C}\right]^{-1} \tag{A.71}$$

$$= [\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{K}_{\mathfrak{D}}^{-1}\boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})^{\top} + \frac{\boldsymbol{I}}{C} + \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{A}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) + \boldsymbol{q}\boldsymbol{b}^{\top}\boldsymbol{K}_{\mathfrak{D}}^{\top}(\boldsymbol{x}) + \boldsymbol{K}_{\mathfrak{D}}(\boldsymbol{x})\boldsymbol{b}\boldsymbol{q}^{\top} + c\boldsymbol{q}\boldsymbol{q}^{\top}]^{-1}. \tag{A.72}$$

Bibliography

[Adam 2008] Amit Adam, Ehud Rivlin, Ilan Shimshoni and David Reinitz. *Robust real-time unusual event detection using multiple fixed-location monitors*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pages 555–560, 2008. (Cited on pages 6, 32 and 58.)

[Aho 1972] Alfred V Aho and Jeffrey D Ullman. The theory of parsing, translation, and compiling. Prentice-Hall, Inc., 1972. (Cited on page 11.)

[Albanese 2008] Massimiliano Albanese, Rama Chellappa, Vincenzo Moscato, Antonio Picariello, VS Subrahmanian, Pavan Turaga and Octavian Udrea. *A constrained probabilistic petri net framework for human activity detection in video*. Multimedia, IEEE Transactions on, vol. 10, no. 6, pages 982–996, 2008. (Cited on page 12.)

[Antic 2011] Borislav Antic and Björn Ommer. *Video parsing for abnormality detection*. In Proceedings of IEEE International Conference on Computer Vision (ICCV), pages 2415–2422. IEEE, 2011. (Cited on page 11.)

[Aronszajn 1950] Nachman Aronszajn. *Theory of reproducing kernels*. Transactions of the American mathematical society, vol. 68, no. 3, pages 337–404, 1950. (Cited on page 13.)

[Ben-Hur 2002] Asa Ben-Hur, David Horn, Hava T Siegelmann and Vladimir Vapnik. *Support vector clustering*. The Journal of Machine Learning Research, vol. 2, pages 125–137, 2002. (Cited on page 8.)

[Benezeth 2009] Yannick Benezeth, P-M Jodoin, Venkatesh Saligrama and Christophe Rosenberger. *Abnormal events detection based on spatio-temporal co-occurences*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2458–2465. IEEE, 2009. (Cited on pages 6 and 10.)

[Benezeth 2011] Yannick Benezeth, Pierre-Marc Jodoin and Venkatesh Saligrama. *Abnormality detection using low-level co-occurring events*. Pattern Recognition Letters, vol. 32, no. 3, pages 423–431, 2011. (Cited on pages 6 and 10.)

[Bishop 2006] Christopher M Bishop and Nasser M Nasrabadi. Pattern recognition and machine learning, volume 1. springer New York, 2006. (Cited on page 7.)

[Blank 2005] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani and Ronen Basri. *Actions as space-time shapes*. In Proceedings of tenth IEEE International Conference on Computer Vision (ICCV), volume 2, pages 1395–1402, 2005. (Cited on pages 6 and 7.)

[Blei 2003] David M Blei, Andrew Y Ng and Michael I Jordan. *Latent dirichlet allocation*. Journal of machine Learning research, vol. 3, pages 993–1022, 2003. (Cited on page 9.)

[Bobick 2001] Aaron F. Bobick and James W. Davis. *The recognition of human movement using temporal templates*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 3, pages 257–267, 2001. (Cited on pages 6 and 7.)

[Boiman 2007] Oren Boiman and Michal Irani. *Detecting irregularities in images and in video*. International Journal of Computer Vision, vol. 74, no. 1, pages 17–31, 2007. (Cited on page 6.)

[Boser 1992] Bernhard E Boser, Isabelle M Guyon and Vladimir N Vapnik. *A training algorithm for optimal margin classifiers*. In Proceedings of ACM the fifth annual workshop on Computational learning theory (COLT), Pittsburgh, PA, USA, July, pages 144–152, 1992. (Cited on pages 8 and 13.)

[Bousquet 2004] Olivier Bousquet, Stéphane Boucheron and Gábor Lugosi. *Introduction to statistical learning theory*. In Advanced Lectures on Machine Learning, pages 169–207. Springer, 2004. (Cited on page 12.)

[Bradley 1997] Andrew P Bradley. *The use of the area under the ROC curve in the evaluation of machine learning algorithms*. Pattern recognition, vol. 30, no. 7, pages 1145–1159, 1997. (Cited on page 38.)

[Bradski 2002] Gary R Bradski and James W Davis. *Motion segmentation and pose recognition with motion history gradients*. Machine Vision and Applications, vol. 13, no. 3, pages 174–184, 2002. (Cited on page 6.)

[Bregler 1997] Christoph Bregler. *Learning and recognizing human dynamics in video sequences*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 568–574. IEEE, 1997. (Cited on pages 6 and 10.)

[Burges 1998] Christopher JC Burges. *A tutorial on support vector machines for pattern recognition*. Data mining and knowledge discovery, vol. 2, no. 2, pages 121–167, 1998. (Cited on pages 7 and 12.)

[Buxton 1995] Hilary Buxton and Shaogang Gong. *Visual surveillance in a dynamic and uncertain world*. Artificial Intelligence, vol. 78, no. 1, pages 431–459, 1995. (Cited on page 9.)

[Calavia 2012] Lorena Calavia, Carlos Baladrón, Javier M Aguiar, Belén Carro and Antonio Sánchez-Esguevillas. *A semantic autonomous video surveillance system for dense camera networks in smart cities*. Sensors, vol. 12, no. 8, pages 10407–10429, 2012. (Cited on pages 6 and 11.)

[Candamo 2010] Joshua Candamo, Matthew Shreve, Dmitry B Goldgof, Deborah B Sapper and Rangachar Kasturi. *Understanding transit scenes: a survey on human behavior-recognition algorithms*. Intelligent Transportation Systems, IEEE Transactions on, vol. 11, no. 1, pages 206–224, 2010. (Cited on pages 1 and 2.)

[Canu 2005] S. Canu, Y. Grandvalet, V. Guigue and A. Rakotomamonjy. *SVM and Kernel Methods Matlab Toolbox*. Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2005. (Cited on pages 8 and 55.)

[Casey 2011] Matthew C Casey, Duncan L Hickman, Athanasios Pavlou and James RE Sadler. *Small-scale anomaly detection in panoramic imaging using neural models of low-level vision*. In Proceedings of SPIE Defense, Security, and Sensing (DSS), pages 80420X–80420X. International Society for Optics and Photonics, 2011. (Cited on page 8.)

[Chanda 2004] Gaurav Chanda and Frank Dellaert. *Grammatical methods in computer vision: An overview*. 2004. (Cited on page 11.)

[Chen 2007] Yufeng Chen, Guoyuan Liang, Ka Keung Lee and Yangsheng Xu. *Abnormal behavior detection by multi-SVM-based Bayesian network*. In Proceedings of International Conference on Information Acquisition (ICIA), pages 298–303. IEEE, 2007. (Cited on pages 6 and 8.)

[Chen 2013] Jie Chen, Cédric Richard and Paul Honeine. *Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model*. IEEE Transactions on Signal Processing, 2013. (Cited on pages 57 and 118.)

[Cheng 1995] Yizong Cheng. *Mean shift, mode seeking, and clustering*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 17, no. 8, pages 790–799, 1995. (Cited on page 30.)

[Choi 2009] Young-Sik Choi. *Least squares one-class support vector machine*. Pattern Recognition Letters, vol. 30, no. 13, pages 1236–1240, 2009. (Cited on pages 84 and 86.)

[Cohn 2003] Anthony G Cohn, Derek R Magee, Aphrodite Galata, David C Hogg and Shyamanta M Hazarika. *Towards an architecture for cognitive vision using qualitative spatio-temporal representations and abduction*. In Spatial cognition III, pages 232–248. Springer, 2003. (Cited on page 6.)

[Collins 2000] Robert T Collins, Alan Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt*et al.* A system for video surveillance and monitoring, volume 2. Carnegie Mellon University, the Robotics Institute Pittsburg, 2000. (Cited on page 2.)

[Comaniciu 2002] Dorin Comaniciu and Peter Meer. *Mean shift: A robust approach toward feature space analysis*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pages 603–619, 2002. (Cited on page 30.)

[Cong 2011] Yang Cong, Junsong Yuan and Ji Liu. *Sparse reconstruction cost for abnormal event detection*. In Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, June, pages 3449–3456, 2011. (Cited on pages 44, 63, 68, 84, 100 and 103.)

[Cortez-Cargill 2009] Pedro Cortez-Cargill, Cristobal Undurraga-Rius, Domingo Mery and Alvaro Soto. *Performance evaluation of the covariance descriptor for target detection*. In International Conference of the Chilean Computer Society, Chile, 2009. (Cited on page 54.)

[Cristianini 2000] Nello Cristianini and John Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods. Cambridge university press:Chambridge,UK, 2000. (Cited on pages 7, 8, 12 and 13.)

[Dalal 2006a] Navneet Dalal. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2006. (Cited on page 32.)

[Dalal 2006b] Navneet Dalal, Bill Triggs and Cordelia Schmid. *Human detection using oriented histograms of flow and appearance*. In European Conference on Computer Vision (ECCV), pages 428–441. Springer, 2006. (Cited on page 32.)

[Davis 2001] James W Davis. *Hierarchical motion history images for recognizing human motion*. In Proceedings of IEEE Workshop on Detection and Recognition of Events in Video, pages 39–46, 2001. (Cited on page 6.)

[Diehl 2003] Christopher P Diehl and Gert Cauwenberghs. *SVM incremental learning, adaptation and optimization*. In Proceedings of International Joint Conference on Neural Networks (IJCNN),Portland, OR, US, July, volume 4, pages 2685–2690, 2003. (Cited on pages 13, 86 and 124.)

[Doersch 2012] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic and Alexei A Efros. *What makes Paris look like Paris?* ACM Transactions on Graphics, vol. 31, no. 4, page 101, 2012. (Cited on page 6.)

[Dollár 2005] Piotr Dollár, Vincent Rabaud, Garrison Cottrell and Serge Belongie. *Behavior recognition via sparse spatio-temporal features*. In Proceedings of 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, pages 65–72, 2005. (Cited on page 6.)

[Fusier 2007] Florent Fusier, Valéry Valentin, François Brémond, Monique Thonnat, Mark Borg, David Thirde and James Ferryman. *Video understanding for complex activity recognition*. Machine Vision and Applications, vol. 18, no. 3-4, pages 167–188, 2007. (Cited on page 12.)

[Ghahramani 1997] Zoubin Ghahramani and Michael I Jordan. *Factorial hidden Markov models*. Machine learning, vol. 29, no. 2-3, pages 245–273, 1997. (Cited on page 9.)

[Ghanem 2004] Nagia Ghanem, Daniel DeMenthon, David Doermann and Larry Davis. *Representation and recognition of events in surveillance video using petri nets*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)., pages 112–112, 2004. (Cited on page 12.)

[Ghanem 2007] Nagia M Ghanem. *Petri Net models for event recognition in surveillance videos*. PhD thesis, 2007. (Cited on page 12.)

[Gong 2003] Shaogang Gong and Tao Xiang. *Scene Events Recognition Without Tracking*. Acta Automatica Sinica, vol. 29, no. 3, pages 321–321, 2003. (Cited on page 6.)

[Gorelick 2007] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani and Ronen Basri. *Actions as space-time shapes*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pages 2247–2253, 2007. (Cited on page 7.)

[Gunn 1998] Steve R Gunn. *Support vector machines for classification and regression*. ISIS technical report, vol. 14, 1998. (Cited on page 12.)

[Haines 2011] Tom SF Haines and Tao Xiang. *Delta-dual hierarchical dirichlet processes: A pragmatic abnormal behaviour detector*. In Proceedings of IEEE International Conference on Computer Vision (ICCV), pages 2198–2205, 2011. (Cited on page 6.)

[Hall 2003] Brian Hall. Lie groups, lie algebras, and representations: an elementary introduction, volume 222. Springer: Berlin, Heidelberg, Germany, 2003. (Cited on pages 56 and 118.)

[Hanley 1982] James A Hanley and Barbara J McNeil. *The meaning and use of the area under a receiver operating characteristic (ROC) curve*. Radiology, vol. 743, pages 29–36, 1982. (Cited on pages 38 and 78.)

[Haque 2010] Mahfuzul Haque and Manzur Murshed. *Panic-driven event detection from surveillance video stream without track and motion features*. In Proceedings of IEEE International Conference on Multimedia and Expo (ICME), pages 173–178, 2010. (Cited on page 28.)

[Hoffmann 2007] Heiko Hoffmann. *Kernel PCA for novelty detection*. Pattern Recognition, vol. 40, no. 3, pages 863–874, 2007. (Cited on pages 18 and 93.)

[Honeine 2012] Paul Honeine. *Online kernel principal component analysis: a reduced-order model*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 9, pages 1814–1826, 2012. (Cited on pages 73, 86, 120 and 124.)

[Hongeng 2001] Somboon Hongeng and Ramakant Nevatia. *Multi-agent event recognition*. In Proceedings of IEEE International Conference on Computer Vision (ICCV), volume 2, pages 84–91, 2001. (Cited on pages 6 and 8.)

[Horn 1981] Berthold KP Horn and Brian G Schunck. *Determining optical flow*. Artificial intelligence, vol. 17, no. 1, pages 185–203, 1981. (Cited on pages 22, 30, 108 and 112.)

[Intille 1999] Stephen S Intille and Aaron F Bobick. *A framework for recognizing multi-agent action from visual evidence*. AAAI/IAAI, vol. 99, pages 518–525, 1999. (Cited on page 9.)

[Jensen 2007] Finn Verner Jensen and Thomas Dyhre Nielsen. Bayesian networks and decision graphs. Springer, 2007. (Cited on page 9.)

[Jiang 2006] Hao Jiang, Mark S Drew and Ze-Nian Li. *Successive convex matching for action detection*. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 1646–1653. IEEE, 2006. (Cited on page 7.)

[Jiang 2011] Fan Jiang, Junsong Yuan, Sotirios A Tsaftaris and Aggelos K Katsaggelos. *Anomalous video event detection using spatiotemporal context*. Computer Vision and Image Understanding, vol. 115, no. 3, pages 323–333, 2011. (Cited on pages 6 and 10.)

[Jiang 2012] Fan Jiang. Anomalous event detection from surveillance video. ProQuest / UMI, 2012. (Cited on pages 6 and 10.)

[Jiménez-Hernández 2010] Hugo Jiménez-Hernández, Jose-Joel González-Barbosa and Teresa Garcia-Ramírez. *Detecting abnormal vehicular dynamics at intersections based on an unsupervised learning approach and a stochastic model*. Sensors, vol. 10, no. 8, pages 7576–7601, 2010. (Cited on pages 6 and 9.)

[Joo 2006] Seong-Wook Joo and Rama Chellappa. *Attribute grammar-based event recognition and anomaly detection*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), pages 107–107. IEEE, 2006. (Cited on page 11.)

[Ke 2007] Yan Ke, Rahul Sukthankar and Martial Hebert. *Event detection in crowded videos*. In Proceedings of IEEE eleventh International Conference on Computer Vision (ICCV), pages 1–8, 2007. (Cited on page 7.)

[Kim 2009] Jaechul Kim and Kristen Grauman. *Observe locally, infer globally: a space-time MRF for detecting abnormal activities with incremental updates*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2921–2928, 2009. (Cited on pages 6 and 9.)

[Knuth 1968] Donald E Knuth. *Semantics of context-free languages*. Mathematical systems theory, vol. 2, no. 2, pages 127–145, 1968. (Cited on page 11.)

[Kosmopoulos 2010] Dimitrios Kosmopoulos and Sotirios P Chatzis. *Robust visual behavior recognition*. IEEE Signal Processing Magazine, vol. 27, no. 5, pages 34–45, 2010. (Cited on pages 6 and 9.)

[Kwak 2011] Sooyeong Kwak and Hyeran Byun. *Detection of dominant flow and abnormal events in surveillance video*. Optical Engineering, vol. 50, no. 2, pages 027202–027202, 2011. (Cited on pages 6 and 32.)

[Lafferty 2001] John Lafferty, Andrew McCallum and Fernando CN Pereira. *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*. 2001. (Cited on page 10.)

[Laptev 2007] Ivan Laptev and Patrick Pérez. *Retrieving actions in movies*. In Proceedings of IEEE International Conference on Computer Vision (ICCV), pages 1–8, 2007. (Cited on page 6.)

[Laptev 2008] Ivan Laptev, Marcin Marszalek, Cordelia Schmid and Benjamin Rozenfeld. *Learning realistic human actions from movies*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2008. (Cited on page 32.)

[Lavee 2009a] Gal Lavee, Ehud Rivlin and Michael Rudzsky. *Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video*. Rapport technique, Technion-Israel Inst. Technol., Haifa, Israel, CIS-2009-06, 2009. (Cited on pages 5, 7 and 19.)

[Lavee 2009b] Gal Lavee, Ehud Rivlin and Michael Rudzsky. *Understanding video events: a survey of methods for automatic interpretation of semantic occurrences in video*. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 39, no. 5, pages 489–504, 2009. (Cited on pages 5 and 6.)

[Lee 2012] Young-Sook Lee and Wan-Young Chung. *Visual sensor based abnormal event detection with moving shadow removal in home healthcare applications*. Sensors, vol. 12, no. 1, pages 573–584, 2012. (Cited on page 6.)

[Lv 2006] Fengjun Lv, Xuefeng Song, Bo Wu, Vivek Kumar Singh and Ramakant Nevatia. *Left-luggage detection using Bayesian inference*. In Proceedings 9th IEEE International Workshop on PETS, pages 83–90. Citeseer, 2006. (Cited on page 9.)

[Masoud 2003] Osama Masoud and Nikos Papanikolopoulos. *A method for human action recognition*. Image and Vision Computing, vol. 21, no. 8, pages 729–743, 2003. (Cited on page 7.)

[Medioni 2001] Gérard Medioni, Isaac Cohen, François Brémond, Somboon Hongeng and Ramakant Nevatia. *Event detection and analysis from video streams*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 23, no. 8, pages 873–889, 2001. (Cited on pages 6 and 9.)

[Mehran 2009] Ramin Mehran, Alexis Oyama and Mubarak Shah. *Abnormal crowd behavior detection using social force model*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, June, pages 935–942, 2009. (Cited on pages 44, 68, 84 and 103.)

[Metro a] Moscow Metro. *Offical websit for Moscow Metro, http://www.mosmetro.ru/*. (Cited on page 2.)

[Metro b] New York Metro. *Offical websit for Metropolitan Transportation Authority, http://new.mta.info/*. (Cited on page 2.)

[Metro c] Paris Metro. *Offical websit for Autonomous Operator of Parisian Transports, http://www.ratp.fr/*. (Cited on page 2.)

[Metz 1978] Charles E Metz. *Basic principles of ROC analysis*. In Proceeding of Seminars in nuclear medicine, volume 8, pages 283–298, 1978. (Cited on page 38.)

[Neubeck 2006] Alexander Neubeck and Luc Van Gool. *Efficient non-maximum suppression*. In Proceedings of the 18th IEEEE International Conference on Pattern Recognition (ICPR), volume 3, pages 850–855, 2006. (Cited on pages 31 and 112.)

[Ng 2001] Jeffrey Ng and Shaogang Gong. *Learning Pixel-Wise Signal Energy for Understanding Semantics*. In Proceedings of British Machine Vision Conference (BMVC), pages 71.1–71.10, 2001. doi:10.5244/C.15.71. (Cited on pages 6 and 7.)

[Ng 2003] Jeffrey Ng and Shaogang Gong. *Learning pixel-wise signal energy for understanding semantics*. Image and Vision Computing, vol. 21, no. 13, pages 1183–1189, 2003. (Cited on pages 6 and 7.)

[Niebles 2008] Juan Carlos Niebles, Hongcheng Wang and Li Fei-Fei. *Unsupervised learning of human action categories using spatial-temporal words*. International Journal of Computer Vision, vol. 79, no. 3, pages 299–318, 2008. (Cited on page 6.)

[Noumir 2012a] Zineb Noumir, Paul Honeine and Cedric Richard. *Kernels for time series of exponential decay/growth processes*. In Proceedings of IEEE International Workshop on Machine Learning for Signal Processing (MLSP), pages 1–6. IEEE, 2012. (Cited on pages 57 and 118.)

[Noumir 2012b] Zineb Noumir, Paul Honeine and Cédric Richard. *One-class machines based on the coherence criterion*. In Proceedings of IEEE Statistical Signal Processing Workshop (SSP), pages 600–603, 2012. (Cited on pages 73 and 120.)

[Noumir 2012c] Zineb Noumir, Paul Honeine and Cédric Richard. *Online one-class machines based on the coherence criterion*. In Proceedings of the 20th European Signal Processing Conference (EUSIPCO),Bucharest, Romania, August, pages 664–668, 2012. (Cited on pages 73, 87, 88 and 120.)

[Pang 2008] Yanwei Pang, Yuan Yuan and Xuelong Li. *Gabor-based region covariance matrices for face recognition*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 7, pages 989–993, 2008. (Cited on pages 54 and 116.)

[Pearl 1988] Judea Pearl. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, 1988. (Cited on page 9.)

[PETS 2009] PETS. *Performance Evaluation of Tracking and Surveillance (PETS) 2009 Benchmark Data. Multisensor sequences containing different crowd activities. http://www.cvg.rdg.ac.uk/PETS2009/a.html*. 2009. (Cited on pages 3, 40, 44 and 58.)

[Piciarelli 2005] Claudio Piciarelli, Gian Luca Foresti and Lauro Snidaro. *Trajectory clustering and its applications for video surveillance*. In Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 40–45, 2005. (Cited on pages 6 and 8.)

[Piciarelli 2006] Claudio Piciarelli and Gian Luca Foresti. *On-line trajectory clustering for anomalous events detection*. Pattern Recognition Letters, vol. 27, no. 15, pages 1835–1842, 2006. (Cited on pages 6 and 8.)

[Piciarelli 2007] Claudio Piciarelli and Gian Luca Foresti. *Anomalous trajectory detection using support vector machines*. In IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 153–158, 2007. (Cited on pages 6 and 8.)

[Piciarelli 2008a] C Piciarelli, C Micheloni, Gian Luca Foresti*et al.* *Kernel-based unsupervised trajectory clusters discovery*. In The Eighth International Workshop on Visual Surveillance, 2008. (Cited on pages 6 and 8.)

[Piciarelli 2008b] Claudio Piciarelli, Christian Micheloni and Gian Luca Foresti. *Trajectory-based anomalous event detection*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 18, no. 11, pages 1544–1554, 2008. (Cited on pages 6, 8 and 13.)

[Pittore 1999] Massimiliano Pittore, Curzio Basso and Alessandro Verri. *Representing and recognizing visual dynamic events with support vector machines*. In Image Analysis and Processing, 1999. Proceedings. International Conference on, pages 18–23. IEEE, 1999. (Cited on page 8.)

[Pontil 1998] Massimiliano Pontil and Alessandro Verri. *Properties of support vector machines*. Neural Computation, vol. 10, no. 4, pages 955–974, 1998. (Cited on page 13.)

[Popoola 2012] Oluwatoyin P Popoola and Kejun Wang. *Video-Based Abnormal Human Behavior Recognition—A Review*. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 42, no. 6, pages 865–878, 2012. (Cited on pages 1 and 9.)

[Porikli 2005] Fatih Porikli and Oncel Tuzel. *Bayesian background modeling for foreground detection*. In Proceedings of the third ACM international workshop on Video surveillance & sensor networks (VSSN), pages 55–58, 2005. (Cited on page 25.)

[Porikli 2006a] Fatih Porikli and Tekin Kocak. *Robust license plate detection using covariance descriptor in a neural network framework*. In Proceedings of IEEE International Conference on Video and Signal Based Surveillance (AVSS), pages 107–107, 2006. (Cited on pages 54 and 116.)

[Porikli 2006b] Fatih Porikli and Oncel Tuzel. *Fast construction of covariance matrices for arbitrary size image windows*. In Proceedings of IEEE International Conference on Image Processing, pages 1581–1584, 2006. (Cited on page 54.)

[Porikli 2006c] Fatih Porikli, Oncel Tuzel and Peter Meer. *Covariance tracking using model update based on lie algebra*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 728–735, 2006. (Cited on pages 54 and 116.)

[Rabiner 1989] Lawrence R Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*. Proceedings of the IEEE, vol. 77, no. 2, pages 257–286, 1989. (Cited on page 9.)

[Rakotomamonjy 2008] Alain Rakotomamonjy, Francis Bach, Stéphane Canu, Yves Grandvalet*et al.* *SimpleMKL*. Journal of Machine Learning Research, vol. 9, pages 2491–2521, 2008. (Cited on pages 57 and 118.)

[Ribeiro 2005] Pedro Canotilho Ribeiro and José Santos-Victor. *Human activity recognition from video: modeling, feature selection and classification architecture*. In Proceedings of International Workshop on Human Activity Recognition and Modelling, pages 61–78. Citeseer, 2005. (Cited on page 6.)

[Richard 2009] Cédric Richard, José Carlos M Bermudez and Paul Honeine. *Online prediction of time series data with kernels*. IEEE Transactions on Signal Processing, vol. 57, no. 3, pages 1058–1067, 2009. (Cited on pages 73, 88 and 120.)

[Ryoo 2006] Michael S Ryoo and Jake K Aggarwal. *Recognition of composite human activities through context-free grammar based representation*. In Proceddings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 1709–1718, 2006. (Cited on page 11.)

[Schölkopf 2000] Bernhard Schölkopf, Alex J Smola, Robert C Williamson and Peter L Bartlett. *New support vector algorithms*. Neural computation, vol. 12, no. 5, pages 1207–1245, 2000. (Cited on page 8.)

[Schölkopf 2001] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola and Robert C Williamson. *Estimating the support of a high-dimensional distribution*. Neural computation, vol. 13, no. 7, pages 1443–1471, 2001. (Cited on pages 15, 18, 55 and 72.)

[Schölkopf 2002] Bernhard Schölkopf and Alexander J. Smola. Learning with kernels: Support vector machines, regularization, optimization and beyond. MIT press: Cambridge, MA, USA, 2002. (Cited on pages 16 and 56.)

[Schuldt 2004] Christian Schuldt, Ivan Laptev and Barbara Caputo. *Recognizing human actions: a local SVM approach*. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR), volume 3, pages 32–36, 2004. (Cited on pages 6 and 8.)

[Shawe-Taylor 2004] John Shawe-Taylor and Nello Cristianini. Kernel methods for pattern analysis. Cambridge university press, 2004. (Cited on page 13.)

[Shechtman 2005] Eli Shechtman and Michal Irani. *Space-time behavior based correlation*. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 405–412, 2005. (Cited on pages 6 and 7.)

[Shet 2005] Vinay D Shet, David Harwood and Larry S Davis. *Vidmap: video monitoring of activity with prolog*. In IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 224–229, 2005. (Cited on page 12.)

[Shet 2006] Vinay D Shet, David Harwood and Larry S Davis. *Multivalued default logic for identity maintenance in visual surveillance*. In European Conference on Computer Vision (ECCV), pages 119–132. Springer, 2006. (Cited on page 12.)

[Shi 2010] Yinghuan Shi, Yang Gao and Ruili Wang. *Real-time abnormal event detection in complicated scenes*. In Proceedings of the 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, August, pages 3653–3656, 2010. (Cited on pages 44, 68, 84 and 103.)

[Shilton 2005] Alistair Shilton, Marimuthu Palaniswami, Daniel Ralph and Ah Chung Tsoi. *Incremental training of support vector machines*. IEEE Transactions on Neural Networks, vol. 16, no. 1, pages 114–131, 2005. (Cited on page 71.)

[Singh 2008] Meghna Singh, Anup Basu and Mrinal K Mandal. *Human activity recognition based on silhouette directionality*. Circuits and Systems for Video Technology, IEEE Transactions on, vol. 18, no. 9, pages 1280–1292, 2008. (Cited on page 6.)

[Singh 2012] Saurabh Singh, Abhinav Gupta and Alexei A Efros. *Unsupervised discovery of mid-level discriminative patches*. In European Conference of Computer Vision (ECCV), pages 73–86. Springer, 2012. (Cited on page 6.)

[Siskind 2000] Jeffrey Mark Siskind. *Visual event classification via force dynamics*. In AAAI/IAAI, pages 149–155, 2000. (Cited on page 6.)

[Sminchisescu 2006] Cristian Sminchisescu, Atul Kanaujia and Dimitris Metaxas. *Conditional models for contextual human motion recognition*. Computer Vision and Image Understanding, vol. 104, no. 2, pages 210–220, 2006. (Cited on page 6.)

[Starner 1995] Thad Starner and Alex Pentland. *Visual Recognition of American Sign Language using Hidden Markov Models*. Rapport technique, DTIC Document, 1995. (Cited on page 6.)

[Stolcke 1995] Andreas Stolcke. *An efficient probabilistic context-free parsing algorithm that computes prefix probabilities*. Computational linguistics, vol. 21, no. 2, pages 165–201, 1995. (Cited on page 11.)

[Sun 2010] Deqing Sun, Stefan Roth and Michael J Black. *Secrets of optical flow estimation and their principles*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2432–2439, 2010. (Cited on pages 30 and 112.)

[Sutton 2007] Charles Sutton and Andrew McCallum. *An introduction to conditional random fields for relational learning*. Introduction to statistical relational learning, vol. 93, pages 142–146, 2007. (Cited on page 10.)

[Suykens 1999] Johan AK Suykens and Joos Vandewalle. *Least squares support vector machine classifiers*. Neural processing letters, vol. 9, no. 3, pages 293–300, 1999. (Cited on page 84.)

[Suykens 2002] J.A.K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor and J. Vandewalle. Least squares support vector machines. World scientific: Singapore, 2002. (Cited on page 84.)

[Tax 1999] David MJ Tax and Robert PW Duin. *Data domain description using support vectors*. In Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), volume 99, pages 251–256, 1999. (Cited on page 72.)

[Tax 2001] David Tax. *One-class classification*. PhD thesis, Delft University of Technology, 2001. (Cited on pages 17, 18 and 72.)

[Tax 2004] David MJ Tax and Robert PW Duin. *Support vector data description*. Machine learning, vol. 54, no. 1, pages 45–66, 2004. (Cited on page 17.)

[Tropp 2004] Joel A Tropp. *Greed is good: Algorithmic results for sparse approximation*. IEEE Transactions on Information Theory, vol. 50, no. 10, pages 2231–2242, 2004. (Cited on pages 87 and 88.)

[Tuzel 2005] Oncel Tuzel, Fatih Porikli and Peter Meer. *A bayesian approach to background modeling*. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), pages 58–58, 2005. (Cited on pages 25 and 28.)

[Tuzel 2006] Oncel Tuzel, Fatih Porikli and Peter Meer. *Region covariance: A fast descriptor for detection and classification*. In European Conference on Computer Vision (ECCV), pages 589–600. Springer: Berlin Heidelberg, Germany, 2006. (Cited on pages 53, 63, 84, 100 and 116.)

[Tuzel 2007] Oncel Tuzel, Fatih Porikli and Peter Meer. *Human detection via classification on riemannian manifolds*. In Proceedings of IEEE onference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2007. (Cited on pages 54 and 116.)

[Tuzel 2008] Oncel Tuzel, Fatih Porikli and Peter Meer. *Pedestrian detection via classification on riemannian manifolds*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, no. 10, pages 1713–1727, 2008. (Cited on pages 54 and 116.)

[UMN 2006] UMN. *Unusual Crowd Activity Dataset of University of Minnesota, Department of Computer Science and Engineering, http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi*. 2006. (Cited on pages 3, 23, 40, 58, 78, 93 and 94.)

[Utasi 2008a] Ákos Utasi and László Czúni. *Anomaly Detection with Low-Level Processes in Videos*. In Proceedings of International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), pages 678–681, 2008. (Cited on pages 6 and 9.)

[Utasi 2008b] Akos Utasi and László Czúni. *HMM-based unusual motion detection without tracking*. In Proceedings of the 19th IEEE International Conference on Pattern Recognition (ICPR), pages 1–4, 2008. (Cited on pages 6 and 9.)

[Utasi 2010] Ákos Utasi and László Czúni. *Detection of unusual optical flow patterns by multilevel hidden Markov models*. Optical Engineering, vol. 49, no. 1, pages 017201–017201, 2010. (Cited on pages 6, 9 and 32.)

[Vapnik 1963] Vladimir Naumovich Vapnik and A. Lerner. *Pattern Recognition using Generalized Portrait Method*. Automation and remote control, vol. 24, pages 774–780, 1963. (Cited on pages 7 and 13.)

[Vapnik 1998] Vladimir N Vapnik. Statistical learning theory. Wiley: New York, NY, USA, 1998. (Cited on pages 12 and 16.)

[Vapnik 2000] Vladimir Vapnik. The nature of statistical learning theory. Springer, 2000. (Cited on pages 12 and 16.)

[Varadarajan 2009] Jagannadan Varadarajan and J-M Odobez. *Topic models for scene analysis and abnormality detection*. In Proceedings of the 12th International Conference on Computer Vision Workshops (ICCV Workshops), pages 1338–1345, 2009. (Cited on page 6.)

[Vassilakis 2002] Helen Vassilakis, A Jonathan Howell and Hilary Buxton. *Comparison of feedforward (tdrbf) and generative (tdrgbn) network for gesture based control*. In Gesture and Sign Language in Human-Computer Interaction, pages 317–321. Springer, 2002. (Cited on page 8.)

[Vu 2003] Van-Thinh Vu, Francois Bremond and Monique Thonnat. *Automatic video interpretation: A novel algorithm for temporal scenario recognition*. In IJCAI, volume 3, pages 1295–1300, 2003. (Cited on page 12.)

[Vu 2004] Van-Thinh Vu. *Temporal scenarios for automatic video interpretation*. PhD thesis, 2004. (Cited on page 12.)

[Vu 2006] V-T Vu, François Brémond, Gabriele Davini, Monique Thonnat, Quoc-Cuong Pham, Nicolas Allezard, Patrick Sayd, J-L Rouas, Sébastien Ambellouis and Amaury Flancquart. *Audio-video event recognition system for public transport security*. 2006. (Cited on page 2.)

[Wang 2006] Tao Wang, Jianguo Li, Qian Diao, Wei Hu, Yimin Zhang and Carole Dulong. *Semantic event detection using conditional random fields*. In Proceedings of IEEE conference on Computer Vision and Pattern Recognition Workshop (CVPRW), pages 109–109, 2006. (Cited on pages 6 and 10.)

[Wang 2007] Liang Wang and David Suter. *Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–8, 2007. (Cited on page 6.)

[Xiang 2002] Tao Xiang, Shaogang Gong and Dennis Parkinson. *Autonomous Visual Events Detection and Classification without Explicit Object-Centred Segmentation and Tracking.* In British Machine Vision Conference (BMVC), pages 1–10, 2002. (Cited on page 6.)

[Xiang 2005] Tao Xiang and Shaogang Gong. *Video behaviour profiling and abnormality detection without manual labelling*. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV), volume 2, pages 1238–1245, 2005. (Cited on page 6.)

[Xiang 2008a] Tao Xiang and Shaogang Gong. *Incremental and adaptive abnormal behaviour detection*. Computer Vision and Image Understanding, vol. 111, no. 1, pages 59–73, 2008. (Cited on page 6.)

[Xiang 2008b] Tao Xiang and Shaogang Gong. *Video behavior profiling for anomaly detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 5, pages 893–908, 2008. (Cited on page 6.)

[Yao 2010] Bangpeng Yao and Li Fei-Fei. *Modeling mutual context of object and human pose in human-object interaction activities*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 17–24, 2010. (Cited on pages 6 and 10.)

[Zelnik-Manor 2006] Lihi Zelnik-Manor and Michal Irani. *Statistical analysis of dynamic actions*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 9, pages 1530–1535, 2006. (Cited on pages 6 and 7.)

[Zhong 2004] Hua Zhong, Jianbo Shi and Mirkó Visontai. *Detecting unusual activity in video*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages II–819, 2004. (Cited on page 6.)

[Zhu 2011a] Xudong Zhu and Zhijing Liu. *Human behavior clustering for anomaly detection*. Frontiers of Computer Science in China, vol. 5, no. 3, pages 279–289, 2011. (Cited on page 10.)

[Zhu 2011b] Xudong Zhu, Zhijing Liu and Juehui Zhang. *Human Activity Clustering for Online Anomaly Detection*. Journal of Computers, vol. 6, no. 6, pages 1071–1079, 2011. (Cited on page 10.)

[Ziliani 2005] Francesco Ziliani, S Velastin, Fatih Porikli, Lucio Marcenaro, T Kelliher, Andrea Cavallaro and Philippe Bruneaut. *Performance evaluation of event detection solutions: the CREDS experience*. In Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 201–206. IEEE, 2005. (Cited on page 2.)

Tian WANG

Doctorat : Optimisation et Sûreté des Systèmes

Année 2014

Algorithmes d'apprentissage mono-classe pour la détection d'anomalies dans les flux vidéo

La vidéosurveillance représente l'un des domaines de recherche privilégiés en vision par ordinateur. Le défi scientifique dans ce domaine comprend la mise en œuvre de systèmes automatiques pour obtenir des informations détaillées sur le comportement des individus et des groupes. En particulier, la détection de mouvements anormaux de groupes d'individus nécessite une analyse fine des frames du flux vidéo. Dans le cadre de cette thèse, la détection de mouvements anormaux est basée sur la conception d'un descripteur d'image efficace ainsi que des méthodes de classification non linéaires. Nous proposons trois caractéristiques pour construire le descripteur de mouvement : (i) le flux optique global, (ii) les histogrammes de l'orientation du flux optique (HOFO) et (iii) le descripteur de covariance (COV) fusionnant le flux optique et d'autres caractéristiques spatiales de l'image. Sur la base de ces descripteurs, des algorithmes de machine learning (machines à vecteurs de support (SVM)) mono-classe sont utilisés pour détecter des événements anormaux. Deux stratégies en ligne de SVM mono-classe sont proposées : la première est basée sur le SVDD (online SVDD) et la deuxième est basée sur une version « moindres carrés » des algorithmes SVM (online LS-OC-SVM).

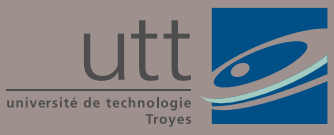
Mots clés : détection du signal - analyse multivariée - machines à vecteurs support - analyse de covariance.

Abnormal Detection in Video Streams via One-class Learning Methods

One of the major research areas in computer vision is visual surveillance. The scientific challenge in this area includes the implementation of automatic systems for obtaining detailed information about the behavior of individuals and groups. Particularly, detection of abnormal individual movements requires sophisticated image analysis. This thesis focuses on the problem of the abnormal events detection, including feature descriptor design characterizing the movement information and one-class kernel-based classification methods. In this thesis, three different image features have been proposed: (i) global optical flow features, (ii) histograms of optical flow orientations (HOFO) descriptor and (iii) covariance matrix (COV) descriptor. Based on these proposed descriptors, one-class support vector machines (SVM) are proposed in order to detect abnormal events. Two online strategies of one-class SVM are proposed: The first strategy is based on support vector description (online SVDD) and the second strategy is based on online least squares one-class support vector machines (online LS-OC-SVM).

Keywords: signal detection - multivariate analysis - support vector machines - analysis of covariance.

Thèse réalisée en partenariat entre :



Ecole Doctorale "Sciences et Technologies"