

PH.D THESIS

# Unusual Event Detection in Real-World Surveillance Applications

by

Jogile Kuklyte, B.Eng.

School of Electronic Engineering

Dublin City University

Supervisor:

Prof. Noel E. O'Connor

September 16, 2014



## Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: \_\_\_\_\_ ID No.: \_\_\_\_\_  
Jogile Kuklyte

Date: \_\_\_\_\_

# ACKNOWLEDGEMENTS

There are a number of people I would like to thank for assisting me during the project. Firstly, I would like to thank my supervisors Prof. Noel E. O'Connor for giving me the opportunity to pursue this interesting and challenging project and for invaluable guidance and advice that I have received. For supporting me through the ups and downs of the creative period. I would like to give a special thanks to Dr. Philip Kelly and Dr. Ciarán Ó Conaire for helping me on the first steps of my research. Dr. Kevin McGuinness for great advices and suggestions towards the end of the project. Dian Zhang, for listening and commenting on the proposed ideas. Dr. Dave Scott for cheering me up during the long writing periods. Dr. Kevin Collins for corrections and writing advice. Ramya Hebbalaguppe for always seeing the positive side. Margaret Malone, for a genuine care and support whenever I needed one. L1.29 lab-mates for a friendly atmosphere, which made me feel at home during long working hours. Everyone whom I met during my time in university who inspired, supported and challenged my ideas. I would like to thank IRCSET, BT and CLARITY for giving me a scholarship and resources to undertake this research project. This was the toughest as well as the most interesting period of my life, of which lessons and experience will be always with me. I would like to thank my twin, who understands me inside-out and unconditionally supports me. My mom and dad, who always believe in me and taught me to never give up. Finally, I would like to thank my husband for putting up with my weekends in the office, for interrupting my long hours of work, and

for giving me the motivation to finish this thesis. It is a blessing to have you all. This thesis would not be possible without you all.

## List of Publications

- Kuklyte, J., Kelly, P., Ó Conaire, C., O'Connor, N. E., & Xu, L. Q. (2009). Anti-social behavior detection in audio-visual surveillance systems. In: PRAI\*HBA - The Workshop on Pattern Recognition and Artificial Intelligence for Human Behaviour Analysis, 9-11 December 2009, Reggio Emilia, Italy.
- Kelly, P., Ó Conaire, C., Monaghan, D., Kuklyte, J., Connaghan, D., Prez-Moneo Agapito, J. D., & Daras, P. (2010). Performance analysis and visualisation in tennis using a low-cost camera network. In: ACM/IEEE International Conference on Distributed Smart Cameras, 25-29 October 2010, Firenze, Italy.
- Kuklyte, J., Kelly, P., & O'Connor, N. E. (2011, August). PhD forum: Investigating the performance of a multi-modal approach to unusual event detection. In: Distributed Smart Cameras (ICDSC), 22-25 Aug 2011, Ghent, Belgium.
- Hebbalaguppe, R., McGuinness, K., Kuklyte, J., Healy, G., O'Connor, N. E., & Smeaton, A. (2013). How Interaction Methods Affect Image Segmentation: User Experience in the Task. In: The 1st IEEE Workshop on User-Centred Computer Vision (UCCV), 16-18 Jan 2013, Tampa, Florida, U.S.A..
- Kuklyte, J., McGuinness, K., Hebbalaguppe, R., Direkoglu, C., Gualano, L., & O'Connor, N. E. (2013, July). Identification of moving objects in poor quality surveillance data. In: Image Analysis for Multimedia Interactive Services (WIAMIS), 3-5 July 2013, Paris, France.



# Contents

<b>List of Figures</b>	<b>i</b>
<b>List of Tables</b>	<b>v</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Overview . . . . .	2
1.2 Motivation . . . . .	3
1.2.1 Event Detection Framework . . . . .	12
1.3 Aims and Objectives . . . . .	18
1.4 Thesis Outline . . . . .	22
1.5 Summary . . . . .	22
<b>2 Literature Review</b>	<b>24</b>
2.1 Overview . . . . .	24
2.2 Visual Surveillance . . . . .	24
2.2.1 State-of-the-art . . . . .	25
2.2.2 Application Areas . . . . .	29
2.2.3 Discussion . . . . .	33
2.3 Event Representation . . . . .	34
2.4 Event Classification . . . . .	38
2.5 Anomaly Detection . . . . .	39
2.5.1 Definition . . . . .	40

2.5.2	Application Areas . . . . .	42
2.5.3	Discussion . . . . .	42
2.6	Result Evaluation . . . . .	43
2.7	Overall Conclusions . . . . .	46
2.8	Summary . . . . .	47
<b>3</b>	<b>A baseline approach to unsupervised event detection</b>	<b>48</b>
3.1	Overview . . . . .	48
3.2	Baseline System . . . . .	49
3.2.1	Temporal segments . . . . .	52
3.2.2	Feature vectors . . . . .	53
3.2.3	Training . . . . .	59
3.2.4	Classification . . . . .	64
3.3	Dataset . . . . .	65
3.4	Evaluation of Results . . . . .	69
3.5	Conclusions . . . . .	71
3.6	Summary . . . . .	72
<b>4</b>	<b>Visual descriptors for event representation</b>	<b>73</b>
4.1	Overview . . . . .	73
4.2	Definition of Visual Event . . . . .	74
4.2.1	Detection of Space-Time Local Regions . . . . .	78
4.2.2	Descriptors of Local Regions . . . . .	82
4.2.3	Postprocessing of Descriptors . . . . .	85
4.3	Comparison of Local Region Descriptors . . . . .	87
4.3.1	Visual Surveillance Dataset . . . . .	88
4.3.2	Results . . . . .	91
4.4	Conclusions . . . . .	96
4.5	Summary . . . . .	97

<b>5</b>	<b>Comparison of modeling and classification techniques</b>	<b>98</b>
5.1	Overview . . . . .	98
5.2	Motivation . . . . .	99
5.3	Dataset . . . . .	104
5.4	Supervised versus unsupervised training . . . . .	106
5.4.1	Experimental setup . . . . .	109
5.4.2	Evaluation . . . . .	111
5.5	Incremental vs batch learning . . . . .	116
5.5.1	Evaluation . . . . .	117
5.6	Dynamic threshold . . . . .	122
5.6.1	Evaluation . . . . .	123
5.7	Proposed Video Processing Flow . . . . .	128
5.8	Qualitative evaluation . . . . .	129
5.9	Conclusions . . . . .	133
5.10	Summary . . . . .	135
<b>6</b>	<b>Combining classifiers</b>	<b>137</b>
6.1	Overview . . . . .	137
6.2	Motivation . . . . .	138
6.3	Algorithm Stability . . . . .	139
6.3.1	Experimental setup . . . . .	141
6.3.2	Evaluation . . . . .	142
6.4	Incremental learning . . . . .	144
6.4.1	Experimental setup . . . . .	145
6.4.2	Evaluation . . . . .	147
6.5	Combining classifiers for surveillance applications . . . . .	149
6.5.1	Experimental setup . . . . .	151
6.5.2	Evaluation . . . . .	152

6.6	Validation of algorithm invariance . . . . .	153
6.7	Conclusions . . . . .	157
6.8	Summary . . . . .	158
<b>7</b>	<b>Conclusions and Future Work</b>	<b>160</b>
7.1	Overview . . . . .	160
7.2	Thesis summary . . . . .	160
7.3	Analysis and discussion of hypotheses . . . . .	165
7.4	Future Work . . . . .	169
7.5	Summary . . . . .	171
<b>A</b>	<b>Graphical User Interface</b>	<b>172</b>
	<b>Bibliography</b>	<b>179</b>

## Abstract

Given the near-ubiquity of CCTV, there is significant ongoing research effort to apply image and video analysis methods together with machine learning techniques towards autonomous analysis of such data sources. However, traditional approaches to scene understanding remain dependent on training based on human annotations that need to be provided for every camera sensor. In this thesis, we propose an unusual event detection and classification approach which is applicable to real-world visual monitoring applications. The goal is to infer the usual behaviours in the scene and to judge the normality of the scene on the basis on the model created. The first requirement for the system is that it should not demand annotated data to train the system. Annotation of the data is a laborious task, and it is not feasible in practice to annotate video data for each camera as an initial stage of event detection. Furthermore, even obtaining training examples for the *unusual* event class is challenging due to the rarity of such events in video data. Another requirement for the system is online generation of results. In surveillance applications, it is essential to generate real-time results to allow a swift response by a security operator to prevent harmful consequences of unusual and antisocial events. The online learning capabilities also mean that the model can be continuously updated to accommodate natural changes in the environment. The third requirement for the system is the ability to run the process indefinitely. The mentioned requirements are necessary for real-world surveillance applications and the approaches that conform to these requirements need to be investigated. This thesis investigates unusual event detection methods that conform with real-world requirements and investigates the issue through theoretical and experimental study of machine learning and computer vision algorithms.

# List of Figures

1.1	Event detection flow-diagram . . . . .	14
3.1	Event detection flow-diagram . . . . .	50
3.2	15 seconds of video feature example . . . . .	58
3.3	The capture environment . . . . .	68
3.4	Sample video frames from one day . . . . .	68
4.1	Event hierarchical taxonomy . . . . .	75
4.2	Space-Time Local Regions (Laptev and Lindeberg, 2006) . . . . .	79
4.3	Descriptors of interest points (Marín-Jiménez et al., 2013) . . . . .	82
4.4	Example frames of the “unusual” (top row) and “usual” (bottom row) classes . . . . .	90
4.5	Comparison of ROC curves of the unusual event detection results when different interest point descriptors are used . . . . .	92
4.6	Comparison of precision-recall curves of the unusual event detection results when different interest point descriptors are used . . . . .	94
4.7	Comparison of results using different descriptors to classify fight no fight event using SVM . . . . .	95
5.1	Selection of training and testing data for experiments . . . . .	110
5.2	Comparison of supervised (SVM) with unsupervised (GMM) learning approaches using different metrics: a) ROC curves b) Precision-Recall curves c) temporal unusual event probabilities . . . . .	112

5.3	Unusual event detection results thresholded using the optimal F1 score for each classification method independently: a) supervised learning (SVM) b) unsupervised learning (GMM) . . . . .	113
5.4	Recall of the fighting events . . . . .	115
5.5	Comparison of incremental (AGG) and batch (GMM) approaches using different metrics . . . . .	119
5.6	Unusual event detection results thresholded using the optimal F1 score for each classification method independently: a) incremental learning (AGG) b) batch learning (GMM) . . . . .	120
5.7	Recall of the fighting events . . . . .	121
5.8	Top row: distances' distributions with the threshold value marked with the read line; bottom row: binary results. The results are evaluated using three methods: (a),(b) <i>mean</i> ; (c),(d) <i>mean+sd</i> ; (e),(f) <i>unusual fraction</i> ; . . . . .	124
5.9	Recall of the unusual events (a) The automatic thresholding approaches applied to the online AGG results (b) F1 threshold selection for online unsupervised learning (online AGG); batch unsupervised learning (GMM) and supervised learning(SVM); . . . . .	126
5.10	Event detection results for qualitative evaluation . . . . .	129
5.11	TP - (4) - A fight with one man falling down on the floor (7.6s) . . . . .	130
5.12	TP - (11) Two fighters approaching each other (1.3s) . . . . .	130
5.13	TN A person browsing (20s) . . . . .	131
5.14	TN - A person walking out of the scene (3s) . . . . .	131
5.15	FN - (12) A person attacking another person (7.2s) . . . . .	131
5.16	FP - (1) coming towards the middle, turning towards the camera and picking up both hands (active gestures) (1.1s) . . . . .	131
5.17	FP - (2) facing the camera and lowering hands towards sides (after raising them) (1.4s) . . . . .	132

5.18	FP - (3) - Both fighters enter the scene from with a fast pace (3s)	132
5.19	FP - (5) A person sitting up after the fight (1s)	132
5.20	FP - (6) A person standing up after the fight (1s)	132
5.21	FP - (7) A person walking after standing for a while (0.8s)	132
5.22	FP - (8) A person walking in a fast pace (1.7s)	133
5.23	FP - (9) A person waving a paper towards the camera (1.3s)	133
5.24	FP - (10) Walking away while waving a white paper in his hand (1.2s)	133
6.1	Incremental bootstrap aggregation from 1 to 50; variance over 10 random experiments	142
6.2	Variance of the event detection results with a model combining an increasing number of aggregated bootstraps	143
6.3	Incrementing training data experimental results a) AUC-ROC statis- tics b) AUC-PR statistics	147
6.4	Combination of supervised and unsupervised classification meth- ods with different weighting schemes	152
6.5	Recall of the unusual events	153
6.6	Unusual event detection results thresholded based on the opti- mized $F - 1$ measure a) AGG b) <i>SVM-fusion</i>	154
6.7	Sample events superimposed on single frames: taken from Ilids dataset sterile zone video data	155
6.8	Incremental clustering (AGG) results on Ilids sterile zone dataset.	156
A.1	Graphical user interface of the live unusual event detection system	173
A.2	Menu explanation of the live unusual event detection system GUI	174
A.3	Unusual events in the live unusual event detection system GUI	175
A.4	Changing event annotation in the live unusual event detection system GUI	176



- A.5 Off-line search in the off-line unusual event detection system GUI . 177
- A.6 Off-line search in the off-line unusual event detection system GUI . 178

# List of Tables

1.1	Commercial visual surveillance analytics systems . . . . .	6
1.2	List of projects dedicated to visual surveillance . . . . .	9
3.1	Unusual event detection state-of-the-art research . . . . .	51
3.2	Freely available computer vision datasets . . . . .	66
3.3	List of scenarios and detection results . . . . .	71
4.1	A list of descriptors used in experiments . . . . .	88
4.2	Properties from the CAVIAR video clips' metadata . . . . .	89
4.3	Video clips representing unusual activity . . . . .	90
5.1	List of events used for training . . . . .	105
5.2	Video clips representing unusual activity; $N$ - length in seconds; $N_i$ - duration of fighting activity; $P_i$ - percentage of frames depicting fighting; . . . . .	105
5.3	Proportions of training data-set . . . . .	106
5.4	Unsupervised (SVM) classification using optimal F1-score thresh- old; 0 - usual class; 1 - unusual class . . . . .	114
5.5	Supervised (GMM) classification using optimal F1-score threshold; 0 - usual class; 1 - unusual class . . . . .	114
5.6	Online classification (AGG) results thresholded with optimal F1- score; 0 - usual class; 1 - unusual class . . . . .	120

5.7	Batch classification (GMM) results thresholded with optimal F1-score threshold; 0 - usual class; 1 - unusual class . . . . .	120
5.8	AGG classification results thresholded with <i>mean</i> threshold; 0 - usual class; 1 - unusual class . . . . .	122
5.9	AGG classification results thresholded with <i>mean+sd</i> threshold; 0 - usual class; 1 - unusual class . . . . .	123
5.10	AGG classification results thresholded with <i>mean</i> threshold; 0 - usual class; 1 - unusual class . . . . .	123
5.11	Comparison of the unusual event class precision, recall and f1-score values of threshold selection methods . . . . .	125
6.1	List of events used for training . . . . .	146

# Chapter 1

## Introduction

### 1.1 Overview

Exploration of the unusual event detection in surveillance applications is based on two main arguments. First, the number of security cameras is growing, and monitoring of the cameras is becoming increasingly difficult. Second, most of the solutions available to the public through commercialized products assume simple visual environments, and when more challenging environments are introduced, laborious and time intensive initialization procedures are required. One of the initialization procedures typically applied is calibration. Calibration is performed after a camera is mounted by mapping multiple points in a scene and recording by capturing a consistent object such as a pole. The pole becomes a reference object for a camera to help to determine the size of an average human being. The usual expectation is that the field of view is not going to change dramatically, including landscape, trees and other objects, and the camera will never be repositioned during routine maintenance. A great amount of published research has been dedicated to improving event detection and recognition techniques for visual monitoring applications, but there has been little effort to consolidate them for applications in real-world environments. The aim of this work is to fill the

gap between the approaches found in the literature and real-world surveillance applications and to propose and investigate solutions that work outside of the laboratory. This chapter begins with a short history of visual surveillance. It shows how surveillance systems worked when they were first introduced and how they evolved with the introduction of video analytics. The current commercial visual surveillance systems and the state-of-the-art techniques found in the research literature are introduced. A break-down of the event detection task into discrete steps is proposed together with the applicability of each step to real-world applications. This chapter concludes with the hypotheses and contributions of this work.

## **1.2 Motivation**

Video surveillance has undergone phenomenal growth since its introduction. For example, the introduction of CCTV cameras for public security in London, UK, started in the early sixties by placing two cameras in Trafalgar Square. By the early nineties, the security infrastructures around London grew to a network consisting of thousands of cameras. Currently there are more than more than four million cameras in the UK according to the M. McCahill and C. Norris report (McCahill and Norris, 2002). Similar examples of growth can be found in the rest of the world as evident by reports for Canada (Dawson et al., 2009), China (Kolekar, 2013), Australia (Wilson and Sutton, 2003). Such systems were introduced to assist the police and security personnel in preventing crime. The benefit of camera networks is clear: instead of having security or law enforcement personnel stationed at every corner, huge territories can be monitored by a few individuals from the control room. Even if an event is not identified at the time of its occurrence, recorded data can be used to provide evidence of the crime and to identify perpetrators and victims after the fact. The technology was effective in the

nineties. However, due to technological advances and decreasing cost, security cameras became affordable not only to governments and large corporations, but also to small businesses and households. As a result, the number of cameras has grown exponentially, e.g. in 2002 there were 4.2 million cameras in the UK — approximately one for every 14 people (McCahill and Norris, 2002). Therefore, the current issue is that many of the cameras are not being monitored by security personnel, and the recorded footage is reviewed only after an accident or crime has happened. Even when the cameras are monitored live, in most control rooms one operator has to monitor multiple camera views at the same time. The fact that attention has to be divided between multiple camera views makes it more likely that an interesting event will be missed. Studies done by Sandia National Laboratories for the U.S. Department of Energy (Goldgof et al., 2009) supported this intuition and showed that after only 20 minutes of watching and evaluating monitor screens, the attention of most individuals drops to well below acceptable levels.

Due to the nature of certain video surveillance scenarios, some steps of the event detection process can be automated to help prevent the security personnel from missing events of interest. Many commercial systems now include analytic software capable of some level of event detection that can trigger an alarm. The most basic approach that can be applied to direct the attention of the security personnel is motion detection. It can help to identify the periods when moving objects are in the camera view and allow the periods when nothing happens to be ignored. A more advanced approach is to implement an object tracking algorithm. In addition to the allocation of attention to the interesting periods, it gives information about the motion trajectory of an object in the scene. Lucas and Kanade (1981a) proposed a kernel-based tracking algorithm which has a number of variations, but is still widely used in object tracking applications. To improve the accuracy of the motion detection and tracking results, video analytics

software can allow the definition of the region-of-interest as well as the definition of the attributes of the moving object that would trigger an alarm. What is more, current video analytics technology allows retrieval of predefined events from large amounts of video. The events are usually defined by motion in restricted areas. Various alerts triggered by the pre-defined events can be disseminated using text messaging, on-screen alerts, email, geocoded maps, and video. Video storage can also be reduced significantly if the user chooses to record only during such events. A comparison of surveillance video analytics capabilities can be found in (Goldgof et al., 2009). Based on this analysis, a more comprehensive list of commercial analytic systems can be seen in Table 1.1. This list shows the video analytics companies targeting standard surveillance tasks that are usually performed by security personnel such as: detection of loitering, perimeter breach, detection of unattended object. In addition to that, some of the applications target more unusual scenarios such as fall detection, behaviour recognition, or identification of object posture changes. Based on the Table 1.1, it can be seen that object tracking is a fundamental part of all the video analytics systems applied in surveillance domain. Perimeter breach is also the task that is offered by all companies providing video analytics. Most of the video analytics algorithms have capability to detect of object that is left unattended or is removed from the scene, and some kind of crowd analysis. Less available is detection of people loitering and detection of events or activities, such as falling.

While current video analytics greatly improves passive surveillance systems, it suffers from high false positive rate. Typically, motion detection algorithms are sensitive to illumination changes, camera shake, motion in the background such as moving foliage, distant vehicles and usually cannot deal with continuous motion in the camera field of view. To adapt the current commercially available video analytics algorithms to specific scenarios requires highly specialized knowledge and can be labor intensive and expensive.

Ref.	Track	Breach	Crowd	Obj.	Loiter	Fall	Classif.	Event
(Agent Video Intelligence, 2014)	✓	✓	✓	✓	✓			
(Aimetis, 2014)	✓	✓	✓	✓	✓			
(Eptascape, 2014)	✓	✓	✓	✓	✓			
(Honeywell, 2014)	✓	✓	✓	✓	✓			
(IndigoVision, 2014)	✓	✓	✓	✓				
(IntelliView, 2014)	✓	✓						
(IntelliVision, 2014)	✓	✓	✓	✓		✓		
(Ipsotek, 2014)	✓	✓	✓	✓	✓			
(MarchNetworks, 2014)	✓	✓	✓	✓	✓	✓		
(Mango, 2014)	✓	✓	✓	✓			✓	
(ObjectVideo, 2014)	✓	✓	✓	✓	✓		✓	
(Sightlogix, 2014)	✓	✓						
(Verint, 2014)	✓	✓	✓	✓	✓			
(AgilityVideo, 2014)	✓	✓	✓	✓	✓			
(Nice, 2014)	✓	✓	✓	✓				
(SYNTAXIS, 2014)	✓	✓	✓	✓	✓			
(Dvtel, 2014)	✓	✓		✓	✓			
(Puretechsystems, 2014)	✓	✓	✓	✓	✓		✓	
(Acic, 2014)	✓	✓	✓	✓	✓	✓	✓	✓
(AllGoVision, 2014)	✓	✓	✓	✓	✓			
(BRSLABS, 2014)	✓	✓	✓	✓	✓	✓	✓	✓
(Cognimatics, 2014)	✓	✓	✓					
(Foxstream, 2014)	✓	✓	✓	✓				
(Iomniscient, 2014)	✓	✓	✓	✓	✓	✓	✓	✓
(VideoIQ, 2014)	✓	✓	✓	✓	✓	✓	✓	✓

Table 1.1: Commercial visual surveillance analytics systems, (Goldgof et al., 2009). **Track**: object tracking; **Breach**: detecting perimeter breach; **Crowd**: detection of crowding or people counting; **Obj.**: detection of object left unattended or removed from the captured scenes; **Loiter**: identify people loitering activity; **Fall**: detection a person falling on the floor; **Classif.**: Object classification; **Event**: Event detection.

How to improve current video analytics systems and avoid complex initialization techniques is still an open research question. There have been many projects conducted with the focus on visual surveillance and associated applications such as video/image retrieval, human identification, object recognition, etc. Some of the well-known projects are listed in Table 1.2.

Acronym	Full Name	Aims	Year	Reference
---------	-----------	------	------	-----------



Smarter Transportation	Smarter Transportation	Provide efficient video sequence data analysis either in real time or after an event occurs to enhance security at transportation facilities, including airports, ports, railways and roadways.	2014	IBM (2014)
PROTECT-RAIL	Integrated security of Rail Transport	To Develop implement and asses asset-oriented integrated solutions based on mature technology, and demonstrate a global architecture interoperability.	2010-2014	(Dambra, 2014)
THIS	Transport Hubs Intelligent Surveillance	To be able to provide real-time accurate behavioral analysis of people in transport hub, specific areas from day and night video sensors.	2009-2010	(Saldatos, 2009)
VIRAT	Video and Image Retrieval Analysis Tool	To develop and demonstrate a system which is able to recognize and report actions such as someone has entered a building, shooting, vehicle accelerating, a group is meeting, etc.	2008-2010	(DARPA, 2010)
VSAM	Video Surveillance And Monitoring	Real-time moving object detection and tracking from stationary and moving camera platforms, recognition of generic object classes.	2007-2010	(CMU, 2010)
CARETAKER	Content Analysis and Retrieval Technologies to Apply Knowledge Extraction to massive Recording	Aimed at studying, developing and assessing multimedia knowledge-based content analysis, knowledge extraction components, and metadata management sub-systems in the context of automated situation awareness, diagnosis and decision support.	2006-2008	(Ravera, 2008)
KNIGHT	Real Time Automated Surveillance System	Fully automated, multiple camera surveillance system that detects, categorizes and tracks moving objects in the scene.	2007	Shah et al. (2007)

MAVISS	Multi-modal Audio Visible and Infrared Surveillance System	Low cost surveillance system employing multi-modal information for monitoring small areas and detecting alarming events.	2005	Kumar et al. (2005)
VACE I/II	Video Analysis and Content Extraction	Automatic video content extraction, multi-modal fusion, event recognition and understanding all leading to automatic machine reasoning.	2000-2005	(CMU, 2005)
PRISMATIC	Pro-active Integrated Systems for Security Management	To enhance, develop and implement cost-effective technical tools for remote monitoring and automatic detection of security threatening events to public transport passengers, personnel and property.	2000-2003	(Lagrange, 2003)
HID	Human Identification at Distance	To develop automated biometric identification technologies to detect, recognize and identify humans at great distances.	2000-2004	(GVU, 2004)
AVS	Airborne Visual Surveillance	To develop and demonstrate real-time Precision Video Registration (PVR), Multiple Target Surveillance (MTS), and automated Activity Monitoring (AM) of sites.	1998-2002	(DARPA, 2002)
SAKBOT	Statistic and Knowledge-based Object Tracker	Visual traffic analysis system that consists of two main modules - object detection and object tracking.	2004	Cucchiara et al. (2001)
W4	Who, when, Where, What	Real time visual surveillance system that combines monocular gray-scale and infrared video imagery to monitor people activities in an outdoor environment.	2000	Haritaoglu et al. (2000)
ADVISOR	Annotated Digital Video for Intelligent Surveillance and Optimized Retrieval	Development of new algorithms for motion detection, tracking of people, crowd monitoring and behavior recognition.	1998-2002	(Naylor, 2002)
Pfinder	Person Finder	Real time system for tracking people and interpreting their behavior.	1996	Wren et al. (1997)

---

Table 1.2: List of projects dedicated to visual surveillance

The majority of the research efforts focused on these applications is in the computer vision community, because surveillance systems that are widely deployed comprise solely of video cameras. Systems that are targeting specific tasks related to visual surveillance are HID, VACE and THIS. Human identification from distance was the main focus of the work in HID project. The proposed method measures static body and stride parameters as a person walks, such as the distance between head and foot, head and pelvis, foot and pelvis, and left foot and right foot. These parameters allow the identification of a person via its gait. The human gait recognition results when a person is fully visible from the front and the side showed above 90% accuracies (Johnson and Bobick, 2001), but the performance of the algorithm was not reported on the more diverse surveillance data.

Defence and military organizations sponsored many pioneering visual surveillance projects. For example, the Defense Advanced Research Projects Agency (DARPA) Information Systems Office in 1998 funded the Airborne Visual Surveillance (AVS) project, followed by the three-year program to develop Video Surveillance and Monitoring (VSAM) technology in 1997, and the two-year project for development of the Video and Image Retrieval Analysis Tool (VIRAT) in 2008.

Project VACE addressed challenges of summarization and visualization techniques for very large video datasets. The output of the project was a test-bed incorporating data, interface and API standards for video information retrieval. The project THIS focused on creating an ontology for content, context, physical object and action in visual data.

Projects that target creation of a complete surveillance system include ADVISOR, PRISMATICA, CARETAKER and PROTECTRAIL. One of the early projects

focusing on intelligent visual surveillance was ADVISOR. It focused on implementing a complete surveillance system that integrates people tracking, behaviour recognition and video search and retrieval capabilities. The project culminated in a successful demonstration of the system at the TMB headquarters in Barcelona. The demonstration showed the potential for the ADVISOR system to be used to improve exploitation of data from CCTV cameras, but the anomalous event recognition through learning was not achieved (Naylor and Bastin, 2003). The PRISMATICA project focused on computer-vision solutions to detect situations of interest in busy conditions. Promising results were shown on Metro station data captured in London, Paris and Rome metros. Tasks such as train presence detection, detection of significant change, and loitering detection rates exceeded 80% (Velasin et al., 2005). The combination of visual and audio information was explored in CARETAKER project. The project focused on both online and offline security operations such as tracking, detection of overcrowding or fights, and information retrieval based on text or video queries. The system was developed for monitoring town centers, railway stations and other public spaces using video and audio devices. The PROTECTRAIL project is focusing on creating a security system for railways transport that combines multiple visual analysis tasks. The tasks of interest are tracking, staff, passengers, freight and luggage clearance control, protection of infrastructure, and monitoring of rolling goods.

The early systems, such as Pfinder focused on tracking a person's head, hands and body in real time. It has constraints of having one person in the camera view and a static camera view. The W4 project tracks multiple people using infrared imagery information and gray scale image data. This system is intended for an outdoor environment, especially during the night. SAKBOT uses color information to detect moving objects and to differentiate them from shadows and ghosts (the foreground area where the object was in the previous frame) and achieved 10 frames per second processing speed on a standard PC in 2004.

Project KNIGHT used the then state-of-the-art in computer vision techniques to identify, categorize and track moving objects in the scene and across multiple cameras. Some of the systems are multi-modal, for example, MAVISS employs visible, infrared and audio signals to identify events. A commonality between the methods is that they require *object detection*, followed by *tracking* and then by *supervised classification* procedures. Supervised classification requires definition of the *video events* that are to be detected as well as examples of those events to train the classifier. The approaches work well in a constrained setup where visual events-of-interest are predictable and can be defined in advance, but would break when applied to unconstrained real-world scenarios. The weaknesses of these techniques when real-world scenarios are considered need to be identified and alternative approaches need to be proposed to reflect constraints imposed by such surveillance applications.

While the approaches based on object detection, tracking and supervised learning techniques are proven to be highly successful and accurate in the majority of research experiments, their practical deployment is questionable (Turaga et al., 2008). This forms the motivation for the investigation of other approaches that conform to the requirements of *real-world* applications. Such applications pose requirements that are addressed in this thesis by designing real-world unusual event detection algorithms.

Firstly, the *descriptors* used to represent the events need to be invariant to the captured environment. Due to the variety of visual surveillance environments, assumptions such as visibility of full-length human body, or ability to robustly isolate and track the object in the camera view should be avoided. For example, if a person is only partially visible because he is walking behind a parked vehicle or is partially hidden by shrubs, adaptive object representation algorithms are required to extend the representation of objects, for example people, to include more invariant features such as texture, interest points and motion information to

eliminate the requirement of having a complete set of examples of the objects of interest.

Secondly, *initialization* processes of the algorithms need to be as simple as possible. Each algorithm has a number of parameters that need to be chosen depending on the type of the captured environment. An example could be maximum or minimum duration of the event, average size of a person in a particular camera view, sensitivity of the motion detection algorithm. Furthermore, most of the conventional algorithms rely on accurate training data to be able to detect objects or events of interest. The video analysis approaches for targeted surveillance applications should apply adaptive learning techniques to avoid predefined and labor intensive initialization processes.

Thirdly, surveillance applications have to be designed to run for long periods of time and to process large amounts of continuous data. The *computational cost* for each prediction using algorithms such as decision trees (Xue and Liu, 2013) or instance-based learning (Bishop, 2006) depends on the amount of data used in model creation. Over a period of time, the initial prediction rules become obsolete and new examples are required to be added to the model. Each additional example increases the complexity of the algorithm and eventually the algorithm becomes too slow for real-time processing. Thus, algorithms for which the complexity depends on the number of training examples should be avoided.

These three requirements are taken into account in this thesis when formulating the hypotheses and proposing solutions for real-world unusual event detection.

### **1.2.1 Event Detection Framework**

Video event detection in visual surveillance systems typically follows some generic high-level steps which are used as a guideline throughout this work.

Figure 1.1 shows the data flow diagram, which can be broadly partitioned into two processes: *Abstraction* and *Modeling*. The *Abstraction* process starts with acquiring raw video data in a form of a sequence of video frames which are presented to the machine as matrices of pixel intensities. These matrices are grouped into temporal segments, which are then transformed into  $D$ -dimensional feature vectors  $f$ . In the *Modeling* part, these vectors are used to train a model that defines different groups of events. The created model is further used to make a decision about the nature of the new data which is represented by the same type of vectors  $f$ . External information can be provided to aid the model creation process. This information can be acquired through manual labeling of the data. The labels, or annotations, to the vectors  $f$  can be obtained through the manual assessment of the data in an off-line manner. It can also be collected from the user in an online method via user feedback. Each step can be performed off-line with iterative optimization techniques, or on-line one data sample at a time. While it is common in the literature to optimize each step separately, in real-world scenarios it is desirable to process one data sample at a time. The following discussion considers each of the blocks in the flow-diagram and their applicability to real-world scenarios.

The abstraction part of the process is responsible for translating video sequence inputs into intermediate representations. In the *Video Frames* block, the system acquires data from a video sensor. Video frames are usually compressed using lossy compression algorithms to optimize network bandwidth and to save storage space and the sharpness of boundaries of the objects present in the captured scenes might be lost. To define visual events happening over time, consecutive frames need to be grouped into temporal segments. The *Temporal Segments* block represents algorithms that can be used to determine the boundaries of video segments and to define data units used in the following processing steps. If the procedure is performed online, delay is introduced to aggregate frames into segments. The more frames required to represent the segment, the greater the

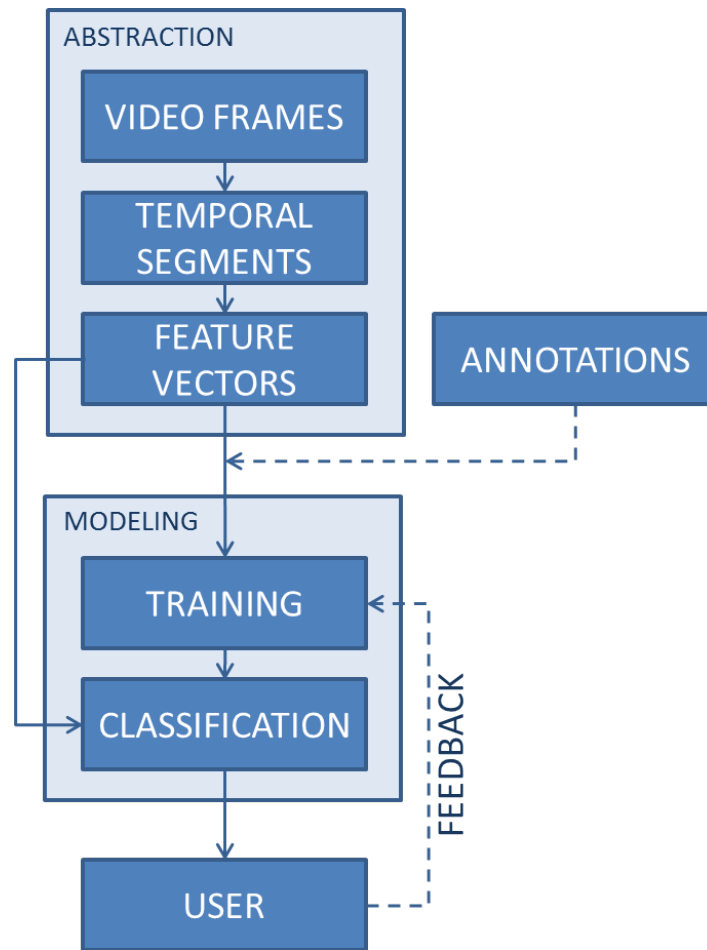


Figure 1.1: A typical data flow for generic event detection

delay. Traditional approaches for motion analysis mainly involve computation of optical flow or feature tracking. Although very effective for many tasks, both of these techniques have limitations. Optical flow approaches mostly capture first-order motion and often fail when the motion has sudden changes. Feature trackers often assume a constant appearance of image patches over time and may fail when the appearance changes, for example, in situations when two objects merge or split. Image structures in vide are not restricted to constant velocity or constant appearance over time. On the contrary, many interesting events in video are characterized by strong variations of the data both in spatial and in temporal dimensions.



The raw data from each frame in a temporal segment is transformed into a relevant event representation form in the *Feature Vectors* block. Event description includes feature extraction algorithms such as edge, motion or interest point detection, as well as techniques for their representation. Due to the nature of video events that evolve over time and space, it is important to include appearance and motion information in the representation. Extraction of appearance information in event description algorithms can broadly be classified into global and local methods. Global methods extract information from the entire video frame to describe the event. Local description methods use the information from the local regions extracted via object detection or other filtering methods to represent the event. To represent appearance information, points with a significant local variation of pixel intensities have been extensively investigated in the past, and such image points are frequently denoted as "interest points" and are attractive due to their high information content. In temporal representation of visual events, points with non-constant motion correspond to accelerating local image structures that might correspond to accelerating objects in the world. Hence, such points might contain important information about the forces that act in the environment and change its structure. Combination of spatial and temporal information forms a final representation of the visual events which is captured in a multi-dimensional vector, also called a feature vector. Space-time event representation should be able to capture event information broadly enough to allow for variations between events from the same class to be discarded, such as diverse walking patterns or various individuals. It should also capture specific information to be able to differentiate between distinct classes of events, for example, running, walking, loitering and fighting. Event representation methods for surveillance visual events are further discussed in Chapter 4.

The feature vectors are used to train a classifier, and to classify previously unseen data. The event modeling part of the overall process is devoted to describing

events of interest formally, and determining whether such an event has occurred. The training, or learning, of the classifier is described in the *Training* block. Based on the availability of *Annotations* for the training data, the training algorithms are grouped into supervised and unsupervised methods. In supervised training approaches, each training data instance is paired with a desired output value by an expert. Based on this information, supervised learning algorithm produces an inferred function which can be used for mapping new examples. On the other hand, unsupervised training, or learning, is trying to find a hidden structure in unlabelled data. Due to the vast amounts of data that needs to be processed in visual security applications, and the dynamic nature of its data, it is a demanding task to provide annotations for all the data. Therefore, unsupervised training might be preferable in these application. The drawback of the unsupervised training approaches might be uncertainty of the final results. Due to lack of corrective process, since the examples given to the algorithm are unlabeled, there is no error signal to evaluate a potential solution. Hybrid approaches called semi-supervised or weakly supervised could be alternative to strictly supervised or strictly unsupervised training approaches. All of the methods will be further investigated in the Chapters 5 and 6 of this thesis.

Depending on the way the training data is used, the training methods can further be classified into batch and online. In batch mode, all the training data is required in advance. This method uses iterative optimization techniques which use all the training data points at each step and delays the classification until after the training is finished. Classical batch learning techniques are off-line and rely on the fact that the learning and testing phases are completely separated in the system. These methods are trained on a specific dataset and then tested in a real-world environment without any further learning. However, in visual surveillance applications, the visual event classes and their properties are dynamic and time varying. In the online approach, the model is updated one sample at a time.

Online learning algorithms are concerned with problems of making decisions about the present based only on knowledge in the past. An online learning algorithm can be summarized by its four main features Liang et al. (2006):

1. The training observations are sequentially (one-by-one or chunk-by-chunk with varying or fixed chunk length) presented to the learning algorithm.
2. At any time, only the newly arrived single or chunk of observations (instead of the entire past data) are seen and learned.
3. A single or a chunk of training observations is discarded as soon as the learning procedure for that particular (single or chunk of) observation(s) is completed.
4. The learning algorithm has no prior knowledge as to how many training observations will be presented.

Because all input is not available for the algorithm, it is forced to make decisions that may later turn out not to be optimal. Nevertheless, online methods are particularly useful in situations that involve streaming data, because it allows classification to be started immediately, makes it possible to adapt to the changing environment and usually does not require storage of all the data from the past. Although it must be noted, that the classification results may be noisy at the beginning and might require some user input to allow for a more accurate representation of the environment at hand. This approach is scalable to large training datasets as it does not iterate over all data samples. Hybrid batch-online methods are also possible. These methods create an initial model using a batch approach, and continue to update it using the online method.

The *Classification* step is responsible for the comparison between the model created during the training step and incoming data represented by feature vectors.

As a result of the comparison, it provides a decision about the labels of events that those feature vectors represent.

Instead of the standard approach, where the annotations are manually collected in advance of the training, *User Feedback* could be applied to obtain annotations. This information, supplied by the user after inspecting selected results, provides an opportunity to utilize supervised training techniques in real-world scenarios.

The vast majority of works that have been published on event and action recognition are concerned with the recognition of a finite set of human actions in known and usually well controlled domains. This work focuses on unusual event detection which can be formulated as a classification task between two types of scenarios - usual and unusual. The usual scenarios are the ones that repeatedly happen in the captured environment, and the unusual events are the ones that do not repeat themselves. For example, walking across the corridor outside the research lab is an event that is constantly happening and is classified as usual. On the other hand, fighting outside the corridor is unusual event. As a comparison, people fighting in the CCTV footage from a prison territory would probably occur more often, therefore, there the unusual event detection would not detect this event as unusual.

### **1.3 Aims and Objectives**

The rapid growth in the number of surveillance cameras and expenses associated with human monitoring motivates the investigation of computerized optimizations for security systems. Such solutions aim at closing the gap between the availability of the security cameras and the efficient use of each deployed camera. A side benefit of real-world unusual event detection system would be to advance the applicability of the security cameras in small businesses or homes. It would

drive the security cameras from being a deterrent of crime and a tool for a post-incident analysis, to the instrument helping to intervene in undesirable incidents. The main goal of this research is to propose an *unusual event* detection framework that could be applied to *real-world surveillance scenarios*.

An *unusual event*, in the environment in which the research is taking place, is a video event which is not defined in advance, but is unexpected in the captured environment. It typically corresponds to some activity being performed by humans. An example of an event can be a person walking, running, kicking. An event also occurs when multiple people interact for example fight, walk, and eat. An event can depict not only humans, but other objects such as animals or vehicles. The specific definition of the unusual event is dependent on the environment being captured. In real deployments, many events of people interacting in a monitored scene are not of interest in a surveillance context e.g. people meeting, chatting, embracing, etc. because they are “acceptable” in some sense and/or occur regularly. Typically, it is the unusual or abnormal events that do not occur frequently and thus may not be “acceptable” that are of interest e.g. people fighting. The challenge for an automated system then becomes to identify these abnormal/unusual events from usual events. In the thesis, an unusual event carries the same meaning as an abnormal or interesting event, an anomaly, or novelty - these concepts are used interchangeably throughout this thesis meaning the same thing, unless a different meaning is explicitly specified.

A surveillance scenario in real-world applications could be multi-camera based, multi-modal or single sensor based. In the context of this thesis, a *real-world surveillance scenario* is defined to be a single camera based monitoring system, which is used by a user without a technical background. The camera captures a continuous video stream of data which ideally is processed on-the-fly giving real-time alerts of unusual events. The user can respond to the alerts by giving feedback about the correctness of the alerts. The system runs continuously for

an undefined period of time without interruption. The challenges of the unusual event detection in this environment are to find a definition for the usual and unusual types of scenarios; to find a suitable representation of the video data; to choose suitable training and classification methods; and to achieve event detection results with low false-positive rate and high true positive rate.

A huge amount of research efforts in computer vision and machine learning communities is being focused on improving the efficiency of visual monitoring systems. However, much of the work is toward algorithm development and is not overly concerned with practical deployment issues. The aim of this research is first to identify the constraints imposed by the real-world environment in which the research is taking place, and then to examine and propose a set of techniques that could be practically applied to these challenging natural conditions. The challenges of the targeted environment are addressed in the thesis when evaluating the proposed algorithms. The techniques considered in the thesis cover all the processes needed to make a prediction when provided with the raw visual sensor data. The prediction guides the decision whether the visual event is abnormal/unusual and whether it warrants human intervention of some description. Following the motivations and aims described in the previous sections, four main questions are addressed in the thesis:

1. Could unsupervised classification techniques be applied to unusual event detection and would it yield comparable results to the state-of-the-art supervised classification techniques?
2. Event representation is an essential part of the event classification task. Can space-time visual events be efficiently represented without relying on detection of the moving objects, accuracy of objects' shape, and their complete motion trajectory?

3. Could online training techniques be used as an alternative approach for training in applications where optimization techniques are not feasible?
4. Could a supervised classification method be integrated into the unsupervised system to benefit from the advantages of both techniques?

To answer the research questions, research and experiments tailored to each question are conducted, and the following contributions are identified:

1. Implementation of a baseline unsupervised event detection pipeline and its evaluation on continuous video data specifically created to simulate a real world surveillance scenario.
2. Evaluation of state-of-the-art visual event descriptors focusing on their applicability to real-world surveillance scenarios.
3. Evaluations of unsupervised unusual event detection approaches in comparison to a state-of-the-art supervised approach.
4. Identification of the trade-offs between online and batch training for unsupervised unusual event detection.
5. Stability analysis of the online unsupervised unusual event detection algorithm and experiments on improving the stability.
6. Investigation the effect of incremental learning on both the unsupervised and the supervised classification approaches.
7. Combination of supervised and unsupervised classification approaches to improve the overall performance of unusual event detection.

## **1.4 Thesis Outline**

In Chapter 1, the problem of unusual event detection in real-world surveillance applications is identified. The thesis is introduced by providing motivation, a brief overview of the research area and hypotheses. State-of-the-art computer vision and machine learning techniques applicable to the targeted application are described in Chapter 2. In Chapter 3, a baseline event detection system that conforms to real-world application requirements is implemented and tested. The areas that require further investigation are also identified. State-of-the-art video event description algorithms are examined in Chapter 4 and suitable approaches for visual events representation in surveillance-type environments are suggested. In Chapter 5, the evaluation of an unsupervised unusual event detection process is performed by comparing it to a state-of-the-art supervised event classification process. Comparison of batch and on-line training techniques is carried out, and a qualitative, as well as quantitative, evaluation of the results is performed. The final experiments are presented in Chapter 6 where the stability and the accuracy of the selected unusual event detection process are examined, and solutions are proposed to decrease the variability of the process and to improve the overall unusual event detection accuracy. Finally, in Chapter 7 the results are summarized, and suggestions for future experiments to be carried out are proposed. Additional methods that could substituted or added to the proposed methodology are described in a future work section.

## **1.5 Summary**

From the overview of the current surveillance systems and a brief analysis of the state-of-the-art research targeting surveillance applications, the unusual event detection problem was defined. It was identified that for a generic system to



be applicable to real-world environments three conditions have to be fulfilled: invariance to the captured environment, simplicity in the initialization procedures and constant algorithm complexity with an increasing amount of data. To help identify solutions conforming to the defined requirements, the event detection task was decomposed into two main parts: event abstraction and event modeling. These two parts are further broken down into smaller data processing steps, and implementation of each step is addressed in the rest of the document. The next chapter discusses state-of-the-art algorithms reported in the computer vision and the machine learning literature, and the remainder of the thesis reports the experimental results of evaluating the described steps for an unusual event detection system applicable to real-world surveillance scenarios.

# Chapter 2

## Literature Review

### 2.1 Overview

The main goal of this chapter is to present a review of the state-of-the-art research work in computer vision and machine learning that addresses surveillance applications. It starts with an overview of the current research and trends in visual surveillance surveillance topics. The discussion is then directed to the outlier detection approaches. Subsequently, the algorithms constituting event detection are discussed starting with techniques used for abstraction of visual data, such as representation and event segmentation, followed by event modeling techniques that include training the model and classification of the events of interest. Evaluation techniques are discussed in the final section. Conclusion summarizes the findings in the literature and explains the directions of the subsequent investigations reported in this thesis.

### 2.2 Visual Surveillance

The primary goal of visual monitoring systems is to help to ensure safety and security by detecting the occurrences of activities of interest within a captured

environment. In the past decade, visual surveillance has been an active topic in computer vision and machine learning communities. In this section, the advances of the past decade in the research are discussed. Technology advancements in distributed and heterogeneous surveillance systems, wide area and crowd monitoring, smart embedded cameras. Moreover, challenges in designing surveillance systems and public acceptance of the new technologies are discussed. A number of application areas are then discussed, and the standard video data analysis approaches are discussed for each of the area. At the end of the section, the advances, challenges, and future research trends in visual surveillance are summarized.

### **2.2.1 State-of-the-art**

A number of reviews have been conducted during the past decade, summarizing technological advances and state-of-the-art algorithms in visual surveillance systems. Early steps of intelligent CCTV based surveillance were targeting coverage of bigger areas with less cameras or easier monitoring. First, manual camera control such as pan, tilt and zoom was introduced to be able to manually identify and track events of objects of interest. Second, sequential switching was introduced to allow live coverage of multiple areas by a single security personnel. Then the recording of video data was introduced to provide an audit capability. Eventually, the availability of powerful computers has enabled increased automation with the used of image and video processing (Davies and Velastin, 2005).

A concept of "multimedia" surveillance systems is introduced by Cucchiara (2005), and defined as a system not only capable to furnish multimedia data, but also to collect, process in real-time, correlate and handle multimedia data coming from different sources. A pioneer project that defined cooperative multi-sensor architecture was VSAM Collins et al. (2001). Since then, a significant amount of progress was done on multi-camera object tracking, object re-identification,

data fusion in distributed surveillance systems as documented by Valera and Velastin (2005), and more recently by Wang et al. (2013) and Mosabbeh et al. (2013). Biometric data (Albeshier et al., 2014) as well as speech (Xu et al., 2014) and face recognition (Ming Du et al., 2014) are widely researched in human identification tasks in distributed surveillance systems.

Thanks to a confluence of simultaneous advances in disciplines as computer vision, image sensors, embedded computing, and sensor networks, distributed smart cameras are emerging. Shi and Lichman (2005) discuss challenges associated with smart cameras development and application areas. The idea of smart camera is to convert data knowledge by processing information locally, and transmitting only the higher abstraction results. Compared with PC-based systems, an embedded system is usually subject to many constraints of the design such as low power, limited resources, real-time processing and low cost. Important markets for smart cameras are industries such as robotics, pharmaceutical, manufacturing, food production. The tasks these smart cameras usually perform include bar-code reading, part inspection, flaw detection, dimension measurement, assembly verification, etc. Other emerging markets for smart cameras are intelligent transport systems, automobiles, human computer interfaces, healthcare, games, video conferencing, biometrics (Shi and Lichman, 2005). Development of low-complexity, low cost algorithms suitable for hardware implementation, and software and hardware co-design, in order to map algorithmic requirements to hardware resources are essential. Intelligent video surveillance systems require complicated video processing and is still an area of active research to implement as embedded components Rinner and Wolf (2008). In private places such as restrooms, surveillance systems can not transfer all the information out. Embedded smart cameras would be able to increased security without violating privacy in such places (Qi et al., 2012).

In situations where networks of hundreds of cameras are used to cover wide areas, algorithms for camera self calibration, finding corresponding objects in multiple sensors, and communication methods for data transmission are the main issues of research (Kim et al., 2010). Tracking algorithm for objects, such as people and vehicles, in visual systems where long-term occlusions and movements between camera views are present was proposed by Bowden and Kaewtrakulpong (2005). Each individual camera performed foreground segmentation, shadow removal, and tracking algorithms. Following this, a distributed tracking modules is used to connect spatially and temporally unconnected trajectories of the same objects. Moving objects were correctly re-identified 89% of the time, but the results showed degraded performance when the distances between paths are increased. Exchange of information between video sensors in the distributed system is addressed by Senst et al. (2011), where communication protocol, activity information and relative locations of cameras are communicated to the central node and are made available via user-interface to the end user.

When crowded environments need to be interpreted by visual data analysis algorithms, properties such as density and flow are used. The techniques for crowd analysis do not attempt to identify individuals in a crowd - the crowd is monitored as a generalized entity. Ideal gas theory provide a basis for predicting the behaviour of crowds (Davies and Velastin, 1995). Some analogies with fluid behaviour captured in computational fluid dynamics (CFD) (Nam and Hong, 2014) and with the behaviour of charged particles in an electric field (Davies and Velastin, 1995) may also be observed in visually captured crowds. Another important crowd feature is density, which defines number of people per unit area. Crowd density can be estimated by employing background removal techniques, texture analysis, or other pattern recognition techniques (Zhan et al., 2008). Crowd monitoring is of interest in urban places such as rail-stations, shopping malls and airports. Detecting of individuals who remain in once place while surrounding

crowds move could be an example of scenario of interest to be detected in crowded environments.

When designing a surveillance system, trade-offs have to be made. A review produced by Haering et al. (2008) focused on a discussion of the trade-offs such as sensor calibration versus fast sensor installation, choosing parameters of system components such as video storage, alerting services, camera controls, etc. Even though calibration of the sensors enables system to use the absolute size and speed of the detected objects, this labor intensive task can be replaced by a definition of size filters. Automatic sensor calibration is an active research topic in machine vision community. Evans and Ferryman (2010) use moving pedestrians to calibrate visual systems with multiple cameras and overlapping fields of view. They achieve calibration errors of 3 to 26 pixels. Choosing type of system, for example, distributed versus heterogeneous systems, generic versus specific system, is dependent on the application and different video analysis approaches would be required for each of the type. The need for improvement in computer vision algorithms to address issues such as occlusion handling, fusion of 2D and 3D tracking, anomaly detection, and behaviour prediction, were identified. A survey by Rababaah (2012) discusses strengths and weaknesses of possible components video data analytics such as motion segmentation, region of interest (ROI) characterization and tracking, event recognition and scenario profiling. The findings of the survey were that there is little work done in the area of large scale surveillance systems and that the majority of the work reported in intelligent visual surveillance area lack fusion oriented methodologies.

(Liu et al., 2013) composed a broad overview of the research in visual surveillance. A third-generation surveillance system (3GSS) that includes multi-sensor environments, wireless sensor networks, distributed intelligence and awareness, was discussed by Raty (2010). The review was focused on the technologies required to implement such a system such as architecture and middleware con-

siderations. The challenges in the current surveillance systems were identified such as application of wireless networks, energy efficiency and scalability.

### **2.2.2 Application Areas**

The research targeting intelligent visual surveillance systems covers different application areas such as security in *public transport* (Goldgof et al., 2009; Candamo et al., 2010), *motorways* (Kastrinaki et al., 2003; Sun et al., 2006; Buch et al., 2011), *airports* (Weber and Stone, 1994; Gong and Xiang, 2003; Foucher et al., 2011; Jargalsaikhan et al., 2013), *public places* (Zhan et al., 2008; Ho et al., 2012), and *homes* (Patrick and Bourbakis, 2009; De Silva et al., 2012; Brezovan and Badica, 2013). The main research areas that contribute to the progress of intelligent systems are computer vision, machine learning, and data management. Different visual event detection approaches are taken depending on the application area. Research in the mentioned application areas are further discussed in this section.

#### **Public Transport**

infrastructures, such as train and metro networks, have thousands of security cameras installed to monitor every-day operations. Accidents in the public transport networks affect and endanger thousands of people. As an example of the amount of commuters in these networks, the busiest metro in Europe, Moscow Metro, in 2007 had 9.55 million passengers per day (Goldgof et al., 2009). Large real-time transit visual surveillance systems have already been deployed and tested in large underground transit networks in order to improve the safety and security of the public transport zones. For example, the ADVISOR intelligent surveillance system architecture (Attwood and Watson, 2004) was tested live at *Barcelona's TMB* metro stations. A European Union funded project PRISMATICA (Velastin et al., 2005) tested automatic visual analysis approaches in *Liverpool St.*

station, London and Paris metro station *Gare de Lyon*. The CARETAKER project (Carincotte et al., 2006) integrated the automated situation awareness system in the *Metro of Rome*, Italy. The events of interest in public transport applications were identified by Ziliani et al. (2005). They include proximity breaching, dropping objects on tracks, launching objects across platforms, person trapped by the door of a moving train, walking on rails, falling on the tracks and crossing the rails. (Candamo et al., 2010) - understanding transit scenes. Incidents of interest in public transport systems include overcrowding, loitering, busking, begging, jumping over access barriers, drug dealing. Fears of terrorism lead to continuous monitoring for abandoned luggage or suspect packages.

### **Mototrways**

Intelligent transport systems increasingly apply computer vision and video analytics to monitor *roads and motorways*. Intelligent transport systems can be broadly grouped into three groups: highway traffic monitoring, urban traffic monitoring and on-vehicle vision systems. Traffic analysis of highways appears to be less challenging than other problems based on the detection and classification figures reported in the literature (Buch et al., 2011). Vehicle number plate recognition is one of the most reliable and effective techniques developed for traffic monitoring so far. An urban environment is more challenging than the highway traffic monitoring with respect to traffic density, lower camera angles, a high degree of occlusions (Buch et al., 2011). Moreover, in an urban environment, road users include pedestrians, bicycles and motorbikes which are usually absent from the highways. On-board automotive driver assistance is another transport monitoring application that aims at alerting the driver about driving environment and possible collisions. Most of the visual analysis techniques proposed for on-vehicle monitoring are based on the estimation procedures for recognizing the borders of the lanes and determining the vehicle paths. Two main tasks that



most of the traffic monitoring systems tackle, whether with static or moving cameras, are the estimation of the road geometry and vehicle and obstacle detection (Kastrinaki et al., 2003). On-road vehicle detection is also one of the most important components for any driver assistance systems (Sun et al., 2006). Speed monitoring is carried out by triggering cameras when a vehicle speed exceeds the speed limit. Automatic recognition of vehicle registration number can also be automated. Number plate recognition is also integrated in 'congestion charging' (automatically billing the owners of vehicles which are observed travelling within city centers during chargeable periods of time) and 'automatic road pricing' (toll collection without toll booths) technologies' (Davies and Velastin, 2005). A typical example of research transferred to commercial products is vehicle and traffic surveillance: systems for queue monitoring, accident and incident detection, tunnel monitoring have been developed (Cucchiara, 2005).

### **Airports**

Security is also of big concern in the air transport systems. Visual monitoring in *airports* targets different parts of the airport system operations. Two distinct groups of monitored environments are the outdoor and indoor areas. Weber and Stone (1994) addressed weather monitoring issues in the outdoor environment. They proposed an enhanced weather situation awareness for air traffic control teams where the speed and direction of storm movement are identified using Airport Surveillance Radar (ASR). Gong and Xiang (2003) also targeted outdoor airport environment and focused on aircraft cargo activities such as moving truck, loading, unloading. On the other hand, in indoor airport surveillance applications, the main focus is on recognition of human actions and interactions. Human activities in the airport lobby were investigated by Foucher et al. (2011) and Jargalsaikhan et al. (2013). The proposed algorithms were tested on video

footage from Gatwick airport, and the events of interest were: person running, person putting down an object, person point with his/her hand.

### **Public Places**

Most of the *public spaces* are perceived as crowded environments with a constant flow of people. The analysis of crowd phenomena is of interest in a number of public space applications. Crowd management in events such as sports matches, concerts and public demonstrations is applied to avoid crowd related disasters. To make the layout of shopping malls more convenient to costumers, for example, crowd behaviour is analyzed while designing public spaces. In intelligent environment applications, crowd analysis is applied to assist the crowd or individual in the crowd to navigate spaces such as museums and exhibitions. Zhan et al. (2008) conducted a survey on computer vision techniques for crowd analysis. They suggested that combining non-vision analysis techniques with computer vision techniques could aid in developing an intelligent system, capable of automatically understanding and modeling crowd behaviours. Non-crowded public places were the subject of the review by Ho et al. (2012). They undertook a pilot study where two modalities, video and ultrasonic sensor measurements, were applied to identify and track people in public spaces. Open challenges that produced errors in the study were identified. For example, tracking a person in a crowd, indoor and outdoor lighting, occlusion reasoning. Fusion of different modalities such as ultrasonic, thermal, infrared, audio, and pressure sensor reading, is identified as a potential approach to improving the accuracy of the results.

### **Smart Homes**

Video surveillance in *smart home* environments attempts to detect, recognize, and track persons, and to understand and recognize their actions (Brezovan and Badica, 2013). Pioneering work in this area is the Smart Rooms implemented

by the MIT Media Lab (Pentland, 1996). A number of different categories of the smart home applications exist that focus on assistance in childcare, healthcare and eldercare. Smart home applications aim to support the wellbeing of the residents of the home by providing feedback on the daily activities in the house, raising alarms when unexpected activities happen, or providing information in order to reduce energy consumption (De Silva et al., 2012). Providing an ambient intelligence that is required to make decisions in smart home applications is still a challenging task. Challenges arise due to highly unstructured human behaviour. (Mubashir et al., 2013) reports state-of-the-art methods of fall detection.

### **2.2.3 Discussion**

While the visual surveillance application areas are diverse, the research in all of them focuses on the task of event or activity understanding, description and identification. Data analysis approaches rely on the same computer vision and machine learning techniques. The most important techniques in these research areas, which are part of most visual surveillance systems are (Hu et al., 2004): modeling of environments, detection of motion, classification of moving objects, tracking, understanding and description of behaviours, human identification, fusion of data from multiple cameras. Future development of integrating different sensors (audio, thermal, etc.) was seen as a future trend a decade ago (Davies and Velastin, 2005; Cucchiara, 2005), and it is getting significant attention in current research (Liu et al., 2013).

The surveys are focused on slightly different parts of intelligent visual surveillance systems, but it can be noticed that all of the reviews discuss activity recognition methods based on object detection and tracking techniques, that are active topics in current computer vision research. Moreover, many reviews (Hu et al., 2004; Davies and Velastin, 2005; Haering et al., 2008; Raty, 2010) have a section

about unusual event or activity detection and identify it as an important visual surveillance application.

## 2.3 Event Representation

As previously discussed, event detection is composed of two parts: *abstraction*, where efficient video event descriptors that are pertinent to events of interest are extracted, and *modeling*, where these descriptors are applied to train the classifiers to model those events in order to separate them in the best possible way. In the surveillance application context, the term “event” usually refers to one of the following descriptions:

- human activity (a single person or a few people);
- crowd activity (without distinguishing between individuals);
- movement of vehicles or traffic;
- interactions between humans, objects, and their environment;
- other, such as facial expressions, gesture, environmental conditions.

Most of the research addressing surveillance applications concentrates on human activity. Many researchers focus their models on single person activities (Schuldt et al., 2004; Ke et al., 2005; Niebles et al., 2006; Danafar and Gheissari, 2007; Bregonzio et al., 2009b). Activities of a single person as well as interactions with a small number of other people are also widely researched (Lv et al., 2004; Kitani et al., 2005; Fernández-Caballero et al., 2012). Examples of such activities are walking, meeting, fighting, falling, etc. Activities specific to the environments, such as bank, or airport runway are also addressed by researchers (Georis et al., 2004; Xiang and Gong, 2006). Depending on the activities, different approaches

to modeling and identification techniques are applied. The approaches to event representation and detection also differ for indoor and outdoor environments.

Event representation is a feature extraction task that consists of extracting spatial and motion cues from the video that are discriminative with respect to particular activities within a scene. Description of an event or activity always starts from extracting low-level features. Low level information in a two-dimensional video frame consists of shape, color or texture depicted in that frame. If a sequence of video frames is available, differences between the consecutive frames provide motion information about the objects present in the captured scene. Using a combination of the static information in each frame and the differences between the frames that capture dynamic information, spatio-temporal descriptors are formed. Due to the temporal nature of video events, descriptors from consecutive frames have to be grouped into the meaningful event representations. For this task, temporal segmentation techniques are employed that identify boundaries of events in video data. The final video event representation is then used in the higher level event modeling and classification steps.

Four groups of low-level feature extraction techniques can be identified: background subtraction, optical flow, point trajectory, and filter responses. *Background subtraction* is a popular method for identifying the moving parts of the scene. The shape of the resulting object silhouette is often used to describe objects and their activities using global methods such as moments (Bobick and Davis, 2001). Although silhouettes provide strong cues for action recognition and are insensitive to color, texture and contrast change, they fail in detecting self occlusions and depend on robust background segmentation. *Optical flow* provides a concise description of both the regions of the image undergoing motion and the velocity of that motion. Optical flow often serves as a good approximation of the motion projected onto the image plane. Optical flow based representations do not depend on background subtraction, but they are sensitive to changes in color intensities of

the pixels due to variation of light, camera flicker, or camera motion. *Trajectories* of moving objects have been used as features in many applications to infer the activity of the object. The trajectory itself is not very useful as it is sensitive to translations, rotations and scale changes. Alternative representations such as trajectory velocities, trajectory speed, spatio-temporal curvature or relative motion have been proposed to acquire invariance to some of these variabilities. Extracting unambiguous point trajectories from video is complicated by several factors such as occlusions, noise, background clutter. Temporal *filtering* is an alternative approach to the region-of-interest detection in image sequences. These approaches usually represent actions using bag-of-features (BOF) which are histograms that count the occurrences of the vocabulary-features within a video segment. The practical advantage of this approach is that filter responses show consistency for similar observations, but can account for outliers. Filtering is useful in scenarios with low-resolution or poor quality videos where it is difficult to extract other features such as optical flow or silhouettes.

The methods of finding actions from video data can be classified into: non-parametric, volumetric and parametric. The *non-parametric* approach extracts a set of features from each video frame and compares them to a predefined template. Examples of non-parametric methods are dimensionality reduction, template matching, 3D object matching, and manifold learning. This approach requires background subtraction techniques to extract the shape of the moving object accurately and is limited to fixed short distance camera settings. The *volumetric* approach does not extract features on the frame by frame basis. Instead, it considers a video as a 3-D volume of pixel intensities and extends the standard image features to the 3-D space. Examples of volumetric methods are space-time filtering, constellation of parts, sub-volume matching, tensors. This approach is suitable for capturing the motion of the events that are difficult to define. The *parametric* approach imposes a model on the temporal dynamics of motion, from

which the parameters for a class of actions are estimated. Examples of parametric methods are hidden Markov models (HMM), linear dynamic systems (LDS). This approach is suited for complex actions such as dancing, juggling, and capturing the motion of a music conductor's hands.

The action segmentation task is responsible for separating out single action instances from streams of video data. In the literature, action recognition results are often demonstrated on pre-segmented video clips and each video clip represents a single action from start to finish. When real-world surveillance videos are analyzed, manual segmentation of video sequences is usually not feasible. Temporal action segmentation can be classified into three broad classes (Weinland et al., 2011): boundary detection, sliding windows and grammar concatenation. *Motion boundaries* are usually detected as a preprocessing step before event classification. Boundary detection methods provide a generic segmentation of video without dependence on the action classes, but are subject to errors in the recovery of motion fields and are affected by the presence of multiple, simultaneous movements. Video sequence can be divided into multiple, overlapping segments using a *sliding window*. Classification is performed on all the segments and peaks in the resulting classification scores are interpreted as action locations. The sliding window approach, when compared to motion boundaries, produces much more segments that need to be evaluated by the classifier, thus are usually more computationally intensive. However, sliding window methods based on fewer assumptions can be integrated with any action classifier. *Grammar concatenation* techniques require action representation that involves grammars, which give a model of transitions between states and actions. Concatenative grammars can be build by joining all models in common start and end node and by adding a loop-back transition between these two nodes. Typically these approaches are hand crafted to specific scenarios and do not generalize well to other scenes.

## 2.4 Event Classification

Event classification consists of the steps of learning statistical models from the action representations, and using those models to classify new observations. A major challenge for the algorithms is dealing with the large variability of events that belong to the same class. Objects participating in same class events can exhibit different size, speed, and style. Event classification approaches can be broadly grouped into four groups: logic based methods, graphical models, support vector machines and clustering approaches.

*Logic* based methods rely on formal logic rules to describe activities. Several researchers have proposed ontologies for specific domains of visual surveillance. For example, Chen et al. (2004) proposed an ontology for analyzing social interaction in nursing homes. Hakeem and Shah (2004) proposed an ontology for videos of meetings. Georis et al. (2004) proposed ontologies for activities in bank monitoring settings. Though empirical constructs are fast to design and work well, they are limited in their utility to specific deployments for which they have been designed. A *graphical* model is a probabilistic model for which a graph denotes the conditional dependence structure between random variables. Graphical models can be roughly divided into two families: Bayesian networks and Markovian networks. A Bayesian network (BN) is a graphical model that encodes complex conditional dependencies between a set of random variables that are encoded as local conditional probabilities. Dynamic belief network (DBN) is a generalization of the BN where temporal dependencies are incorporated between random variables. Usually the structure of the DBN is provided by the domain expert and to learn local conditional dependence relations requires very large amounts of training data or extensive hand-tuning by experts both of which limits the applicability of DBNs in large scale settings. A Markov network is represented by an undirected graph and is based on a set of random variables having Markov



properties. Hidden Markov model (HMM) is a widely used method in speech recognition and is increasingly used for visual event recognition. Zia et al. (2013) modeled visual activities by representing each activity by a distinct HMM and achieved 90 to 95% recognition rate for waking, running, skipping, sitting down and standing up activities. In comparison to DBNs, HMM encodes less complex conditional dependence relations. A *Support vector machine* (SVM) is a popular technique for solving problems in classification, regression and novelty detection (Bishop, 2006). An important property of support vector machines is that the determination of the model parameters corresponds to a convex optimization problem so any local solution is also a global optimum. The basic idea of a linear SVM is to find a suitable hyperplane that divides a given dataset into two parts with maximum margin. After that, the SVM is utilized to classify unlabeled datasets. However, in practice, many data are not linearly separable, and no hyperplane may exist that can split the data into two parts. A non-linear SVM can be achieved by using Kernels. An SVM is not only capable of learning in high-dimensional spaces but can also provide high performance with limited training data. *Clustering* analysis is the grouping of data instances in order to discover the structure in the data. The results of a cluster analysis may produce identifiable structure that can be used to generate hypothesis (Webb, 2002). Blank et al. (2005) applied spectral clustering algorithm with a Median Hausdorff distance to get a grouping of the dataset. Wang et al. (2006) used spectral clustering algorithm to find the classes of actions.

## **2.5 Anomaly Detection**

Detecting unusual activities in visual surveillance applications is of considerable practical interest. Algorithms able to single out abnormal events within streaming or archival videos can serve a range of applications - from monitoring surveillance

feeds, or suggesting frames of interest in scientific visual data that an expert ought to analyze, to summarizing interesting content on a day's worth of web-cam data. In any such case, automatically detecting anomalies should significantly improve the efficiency of video analysis, saving valuable human attention for only the most salient content (Kim and Grauman, 2009). Despite the problem's practical appeal, abnormality detection remains technically challenging, and intellectually hard to define. The foremost challenge is that *unusual* events naturally occur with unpredictable variations, making it hard to discriminate a truly abnormal event from noisy observations of normal observation. Furthermore, the visual context in a scene tends to change over time. This implies that a model of what is normal has to be incrementally updated as soon as new observations become available; a model requiring batch access to all data of interest at once would be useless in many real scenarios.

### **2.5.1 Definition**

The objective of unusual event detection is to detect, recognize and learn interesting events. A number of reviews are conducted for unusual event detection in different areas. In signal processing, outlier detection is a popular task for fault detection, radar target detection, detection of masses in mammograms, statistical process control and several other tasks. In the literature this task has been defined using terms such as *suspicious*, *irregular*, *uncommon*, *unusual*, *abnormal*, *novelty*, *anomaly* activity/event/behaviour. Markou and Singh (2003a,b) reviewed important issues related to novelty detection, such as robustness and trade-offs, parameter minimization, generalization and computational complexity. It was identified that assumptions on the nature of the data have to be made in advance of modeling with statistical approaches. Moreover, the amount and quality of the training data is found to be very important in the robust determination of the

model parameters. Hodge and Austin (2004) conducted a comparative review of techniques for outlier detection. The authors have broken down the outlier detection techniques into three fundamental groups: clustering, classification, and a novelty approach. It was concluded that algorithm developers should choose a modeling technique depending on the data type, the available ground-truth labeling, and how they wish to detect and handle the outliers. Most of the earlier work in unusual event detection has been conducted in studies of control systems. Network anomaly detection for managing cyber threats has received a lot of interest in the past decade due to the advances in networking technology that allowed internet to expand as a global medium in communications and commerce. The reviews in this area show that most of the solutions for computer intrusions are still based on the intrusion signatures (Lazarevic et al., 2005; Patcha and Park, 2007; Sabahi and Movaghar, 2008), but there is an increasing trend to employ techniques that create models of generic acceptable behaviours and identify unknown threats by evaluating the deviations from these models (Koch, 2011; Juvonen and Sipola, 2013; Berger et al., 2014).

A number of surveys for anomaly detection in a variety of domains have been conducted by Chandola et al. (2007a,b, 2008, 2012). Chandola et al. (2007a,b) classified outlier detection techniques based on input data, type of supervision and type of outlier. In a subsequent survey (Chandola et al., 2008), a comparative evaluation of a large number of anomaly detection techniques was presented. The conclusions from the experimental results were that, for anomaly detection, the nearest neighbour based techniques have a slightly better performance than clustering techniques. Finite state based techniques are the most consistent techniques, while probabilistic suffix trees and the sparse Markovian techniques perform poorly. It is also noted that the performance of a technique mostly depends on the nature of the data. In the latest survey (Chandola et al., 2012), the anomaly detection techniques were classified based on the problem formulation

that they are trying to solve: sequence based, contiguous subsequence based, and pattern based anomaly detection techniques. The importance of future research into anomaly detection in multivariate sequences and online anomaly detection was highlighted.

Surveys of anomaly detection in automated surveillance applications were conducted by Sodemann et al. (2012) and Popoola and Wang (2012). Popoola and Wang (2012) posed an abnormal behaviour task as a general task in visual surveillance applications and conducted a broad overview of the video event detection field. The anomaly detection task is posed as a pattern learning problem that deals with the classification of video object behaviour by finding good matches either with a priori known templates of behavior or learning and forming statistical models of behaviour types from the time varying feature data. A list of questions that motivate the research in this area are identified:

- Desired level of supervision;
- The types of features;
- Handling of noise and assurance of robustness;
- Compact representation;
- Appropriate similarity measures.

## **2.5.2 Application Areas**

### **2.5.3 Discussion**

The areas that require more input from researchers were identified. Firstly, benchmark evaluation dataset is required for evaluation. Secondly, research on methods that are applicable to real-time scenarios. Thirdly, the need for systems that can detect suspicious events with a minimum description of the scene context.

On the other hand, Sodemann et al. (2012) specifically focused on reviewing anomaly detection algorithms that are real-time. It was identified that further study is needed to address the applicability of methods to a wider range of surveillance targets in varying environments. The lack of a standardized benchmark evaluation approach was emphasized and a requirement for a common widely accessible repository of standardized and meaningful datasets was also highlighted. The following sections discuss event detection, representation, classification, and result evaluation techniques reported in the literature specifically targeting visual surveillance.

## **2.6 Result Evaluation**

Performance of classification algorithms is often measured in terms of accuracy. Evaluation of the accuracy is an important method of comparative evaluation of the algorithms. Many different approaches to evaluation could potentially be employed, from basic Precision and Recall, to more considered approaches that suit certain experimental use cases. A commonly used performance metric for event classification algorithms is the Receiver Operator Curve (ROC) where the true positive rate is plotted against the false positive rate while changing the classification threshold (Viola and Jones, 2001; Xiang and Gong, 2005; Junior et al., 2009; Mehran et al., 2009; Mahadevan et al., 2010). The measure is well suited for event classification techniques because it shows the trade-offs that can be made between accuracy of the results and the detection rate. While the ROC curves are well suited for comparison of different methods, the measure treats both classes equally. Therefore, it is most suitable for experiments with balanced testing data-sets where all the classes have an equal amount of instances. In event detection experiments, there is an imbalance between classes to be segregated. Significantly more data instances are available for usual events than for the events

of interest i.e. the ones to be detected. If ROC is applied to represent the results, the bias towards the usual class would show over-optimistic results.

Precision-recall curves (PRC) are used to get a more accurate evaluation of the event detection task (Davis and Goadrich, 2006; Willems et al., 2008; Umakanthan et al., 2012). The PRC representation of the results focuses only on the detected events and the ones that should be detected. In other words, it takes into account only events that are of interest in the event detection task. It plots the fraction of detected instances that are true events versus the fraction of overall true events detected while varying a threshold. Setting a threshold can be a subjective task depending on the required tradeoffs, therefore a single metric can be defined for comparison of methods before applying the threshold. Area under curve (AUC) is a common metric used to represent the overall performance of the classifier and to get a coarse comparison between different methods (Junior et al., 2009; Loy et al., 2010; Marín-Jiménez et al., 2013).

Once the classification accuracy trend is determined using the above mentioned methods the threshold can be chosen and other metrics can be further evaluated such as confusion matrix (Xiang and Gong, 2005; Danafar and Gheissari, 2007; Laptev et al., 2008).

To be able to construct the confusion matrix, binary results are required. Therefore, threshold for resulting probabilities needs to be chosen. How it is chosen depends on the application requirements, for example, it can be chosen by defining a required accuracy or by evaluation of some relevant measure. In the event detection application, *F-score* can be used to find an optimal threshold as it combines both precision and recall measures to get a single accuracy measure (Parker, 2010). The *F1* measure is the score obtained from the non-weighted precision and recall measures:

$$F1 = 2.0 \cdot \frac{recall \cdot precision}{recall + precision} \quad (2.1)$$

However, the  $F1$  metric requires the annotations of the test data to find the best performing threshold and this method is suitable for evaluation purposes. To evaluate algorithms proposed in this thesis, the ROC curve metric is used due to its popularity with visual event classification algorithms it provides for comparison between different classification approaches. In addition, the PRC is also evaluated for all experiments due to its independence to the amount of true negative detections. Different thresholding approaches and their effect on the accuracy measures are compared in the thesis.

To make the evaluation of the results meaningful, it is important to choose an appropriate dataset. Many different datasets have been used for human action and activity recognition (Chaquet et al., 2013). Some of the datasets are used in very specific action recognition tasks, such as abandoned object, daily living activities, detection of human falls, gait analysis, pose or gesture recognition. In order to compare different event detection and recognition systems, benchmark datasets have to be used. In the past decade, single-human and single-action datasets have been the most popular, such as Weizmann (Blank et al., 2005), KTH (Schuldt et al., 2004). As the robust methods for recognition of a single-action and single-human activities matured, datasets capturing multiple people interaction with uncontrolled illumination conditions and a non-static background became available. Examples of such datasets are: CAVIAR (INRIA, 2004), HOLLYWOOD (Marszalek et al., 2009), ETISEO (INRIA, 2011), i-LIDS (U.K. Home Office, 2011). Event detection datasets used in this project are specific to visual surveillance applications, where video data is assumed to be captured by a static camera, captured scenes depict multiple objects acting at the same time and the amount of data to process is unlimited. An in-house dataset is created to test the proposed algorithms with a dataset depicting the targeted application environment. In addition, benchmark datasets (CAVIAR and i-LIDS) are used to evaluate the

algorithms in order to be able to compare the results to other approaches in the literature. (Maciejewski et al., 2009) - evaluating visual analytics techniques

## 2.7 Overall Conclusions

Machine learning approaches for event classification are grouped into supervised and unsupervised. Most of the surveillance applications use supervised approaches, while unusual event detection applies either unsupervised approaches, or a mixture of supervised and unsupervised approaches. Most of the event detection approaches targeting surveillance applications in the literature apply supervised classification techniques where a model for a list of predefined activities is learnt from a number of representative examples. In addition, most of the approaches that target surveillance applications take video segments rather than continuous video stream. They also rely on the object detection and tracking techniques. Those techniques are not reliable in real world environments. On the other hand, unusual event detection approaches tend to focus on more real-world friendly methods. Unusual event detection approaches usually rely on the training data statistics, as they do not require detection and modeling of specific events. Two approaches of unusual event detection are the most popular. First, number of classes of the events are learnt, and anything that is different from those classes are marked as unusual. Therefore, the unusual class is not predefined, and its values are not fixed to certain scenarios. The second approach is based on learning of a multi-modal representation of the usual environment without learning specific or predefined activities.

Event representation approaches can be grouped into low-level and the high-level representations. Some event detection techniques approaches apply only low-level representation and classify events based on their distribution across temporal and spatial dimensions. The other approaches apply higher level repre-



sentations. Higher level representations require more precise knowledge about the event than the low-level representation.

Standard evaluation of event detection or recognition results is through the ROC curves. As the unusual event detection problem deals with the unbalanced data, the PR curves might give a better indication of the results.

The thesis focuses on the combination of the known approaches that conforms to the real-world environment. The real-world environment is defined by a continuous stream of data and the absence of the training data or specifications of events in the captured environment. In the literature, the relevant approaches have been used separately, whereas in this thesis we combine and test relevant approaches as a complete system.

## **2.8 Summary**

The chapter discussed the techniques proposed in the literature related to unusual event detection in visual surveillance. The overview of the state-of-the-art visual surveillance solutions showed that the main focus is on detecting predefined events, trained from a database of similar events. From a huge variety of event description techniques, the most applicable to the variety of real-world applications are the localized techniques. From the huge variety of training techniques, a subset for unusual event detection is identified. The literature reviewed in this chapter outlines the possible solutions for different steps in the unusual event detection framework proposed in the previous chapter (see the flow diagram in figure 1.1). The details of the approaches most relevant to this thesis are further investigated in the next chapter.

# Chapter 3

## A baseline approach to unsupervised event detection

### 3.1 Overview

In this chapter the event detection pipeline, described in chapter 1 (Figure 1.1), is implemented as a baseline unusual event detection system. It was identified that real-world systems require invariance to the captured environment, simplicity in initialization and stability of algorithmic complexity with increasing amounts of data. These requirements are taken into account when choosing the methods for the steps in the pipeline whereby unsupervised on-line learning is applied together with an adaptive thresholding approach for decision making. The unusual event detection pipeline is applied to continuous data captured over an extended period of time where video data used for training contains unknown activities. The sole assumption is that if there are unusual events in the dataset, then they are rare. The novelty of this approach is the limited amount of user input used to define the parameters of the algorithms. The use of multiple days of unconstrained continuous video data captured 24 hours a day is also novel. To evaluate the system a number of unusual behaviours were performed by vol-

unteers and captured in the dataset. Evaluation of the results obtained helps to guide the subsequent studies and the evaluation methodology for more advanced approaches in later chapters.

## 3.2 Baseline System

In real-world surveillance applications, it is not viable to foresee every possible event and the nature of events depends on the application. An unusual event defined in one environment might be a usual event in a different environment. For example, people walking in the university campus are part of the usual scene, while people walking on the motorway should trigger an alarm as it would be an indication of an incident. Nevertheless, if, for example, there are roadworks on the motorway, people walking would become a usual event over time.

Modeling of events under these conditions is a challenging task. In the literature, event detection approaches tend to address only some of the requirements for processing real-world surveillance data. The focus of this study is to combine the relevant approaches from the literature corresponding to each of the steps in the event detection diagram defined in Figure 3.1 so that all parts are in line with real-world requirements. A number of relevant papers are identified and compared in order to identify the most applicable approaches to the various parts comprising the overall system. The papers are listed in Table 3.1.

The baseline unusual event detection system allows a general event-based analysis of video information containing unknown event types. In particular, focus is on surveillance applications where the environment and the context may vary significantly and unpredictably with different camera setups. Thus, it is not optimized for the detection of a specific action. The proposed baseline unusual event detection system is trained using an on-line agglomerative clustering algorithm where the model of normality is constructed in a fully unsupervised manner.

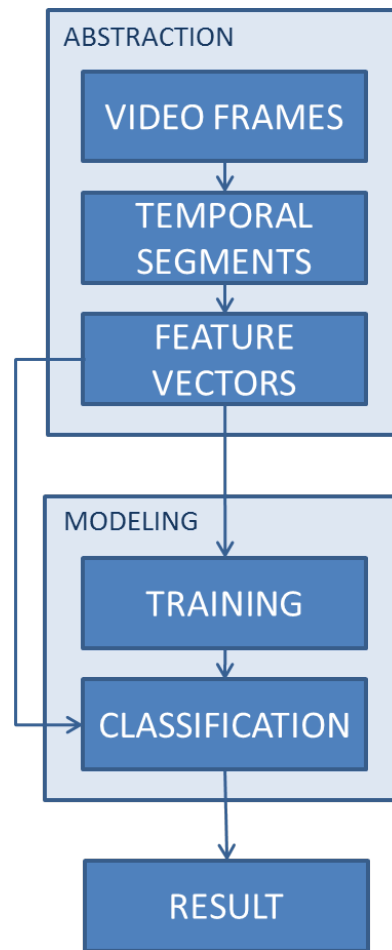


Figure 3.1: Event detection flow-diagram

It is also adaptive, which helps to avoid the model becoming outdated with the long term running of the algorithm. The model of usual activities is continuously updated with the incoming data instances that are classified as usual. The baseline unusual event detection system is described with respect to the data-flow diagram of video event detection in Figure 3.1. Following the diagram, incoming *video frames* are first grouped into *temporal segments*, then descriptors are extracted from each segment to represent the segments with *feature vectors* and a classifier is *trained* using the feature vectors extracted from the training videos. After the training is over, *classification* is carried out on new data that is also grouped into temporal segments and has descriptors extracted to form feature vectors. These feature vectors have the same representation as the training data feature vectors.

Reference	Type of data
Nam and Hong (2014)	Crowd activities
Huo et al. (2014)	Crowded escape events (UMN); Motion on the busy pedestrian walkway (UCSD)
Ouivirach et al. (2013)	Pedestrian and cyclists activities on the pathway near the entrance to a building
Mudjirahardjo et al. (2013)	Tracking pedestrians in uncluttered environment
Feng et al. (2012)	Surveillance video dataset; Subway
Matsugu et al. (2011)	Capture uncluttered scenes of people walking; Tennis match
Lecomte et al. (2011)	Audio recordings from subway mixed with audio recordings of 27 different categories
Mahadevan et al. (2010)	Motion on the busy pedestrian walkway (UCSD)
Loy et al. (2010)	Traffic motion in two road intersections captured with distant cameras
Kim and Grauman (2009)	Activities at the entrance and exit gates of the subway
Breitenstein et al. (2009)	Activities captured by the distant webcam overlooking the Time Square in New York; environmental changes captured by the camera overlooking a lake
Sillito and Fisher (2008)	People behaviour in the entrance lobby of the university
Basharat et al. (2008)	Pedestrian and vehicle motion trajectory based activities
Xiang and Gong (2008)	Activity in the corridor that has a restricted entry to the office
Andrade et al. (2006)	People behaviour in the entrance lobby of the university
Xiang and Gong (2005)	Activity in the corridor that has a restricted entry to the office
Boiman and Irani (2005)	Motion of the limbs of a single person captured in a single spot
Hamid et al. (2005)	Packet delivery activity in a loading dock area of a retail bookstore
Zhong et al. (2004)	Unusual motion patterns in a nursing home eating area, on the road and at the poker table

Table 3.1: Unusual event detection state-of-the-art research

### 3.2.1 Temporal segments

Depending on the nature of the video sequence to be processed, various segmentation approaches can be adopted. The most common assumption in the literature is that the events are pre-segmented based on semantic interpretation (Hamid et al., 2005; Sillito and Fisher, 2008; Basharat et al., 2008; Mahadevan et al., 2010). The segment boundaries in these approaches are usually found manually through laborious previews and examinations of the visual data or by recording event data only.

In some frameworks, a decision as to whether an unusual event happened is made in every frame. This approach does not require temporal segmentation of video data. Here, either only spatial anomalies are considered as in (Breitenstein et al., 2009) and (Sudo et al., 2008), or temporal information is included by representing video data using optical flow statistics as in (Kim and Grauman, 2009). If the data to be processed is known to have low volumes of activity, the non-activity gaps between two consecutive behaviours can be used to segment the videos as in (Xiang and Gong, 2005, 2008), and (Ouivirach et al., 2013). This approach saves a lot of processing power if there are many motionless scenes in the dataset. The drawback of implementing this approach in real-world applications is that the event has to be finished to allow the segment to be further processed, in other words, non-activity occurring after the event defines the end of the event. Furthermore, if the motionless periods are not present between the events, the consecutive events would be merged into one event and would cause incorrect interpretation by the event detector. Another approach is to divide video sequences into non-overlapping (Loy et al., 2010; Andrade et al., 2006) or overlapping windows (Zhong et al., 2004). The non-overlapping segments might fail to capture the important aspects of events that fall between the boundaries of the segments and the overlapping windowing technique solves this issue. In the

windowing approach, the choice of window size is important. Algorithms can be implemented in multiple scales to avoid choosing a window size.

Taking into account the continuous nature of real-world surveillance video data, the most appropriate approach to temporal video segmentation is to use windowing with an overlap. In the implementation of the baseline system, this approach is applied. A continuous video sequence  $V$  is segmented into  $N$  video segments  $V = v_1, v_2, \dots, v_N$  such that ideally each segment contains a single behaviour pattern. The  $n^{\text{th}}$  video segment  $v_n$  consists of  $T_n$  image frames represented as  $v_n = \{I_{n1}, I_{n2}, \dots, I_{nT_n}\}$  where  $I_{nt}$  is the  $t^{\text{th}}$  image frame of  $v_n$ . The overlap between the segments  $f_o$  can take values  $0 \leq f_o < T_n$ . The best representation of the data would be achieved by choosing  $f_o$  equal to  $T_n - 1$  which would create a sliding window with a step of a single frame, but smaller values can be adapted to minimize computational power required.

### 3.2.2 Feature vectors

Spatial and temporal information is required for a discriminative representation of the video events. Only spatial descriptors are used in the unusual event detection task in (Sudo et al., 2008) and (Breitenstein et al., 2009). To represent video events spatial information alone is usually not sufficient because events occupy a duration of time.

Bobick and Davis (2001) introduced an event representation based on temporal motion templates. A temporal template consists of two object motion representations. The first representation is captured in a motion energy image (MEI):

$$E_r(x, y, t) = \bigcup_{i=0}^{\tau-1} D(x, y, t - i) \quad (3.1)$$

The second representation is captured in a motion history image (MHI):

$$H_\tau(x, y, t) = \begin{cases} \tau & \text{if } D(x, y, t) = 1 \\ \max(0, H_\tau(x, y, t - 1) - 1) & \text{otherwise} \end{cases} \quad (3.2)$$

In both equations  $\tau$  defines the temporal extent of the movement and  $D(x, y, t)$  is the binary image sequence indicating regions of motion. The temporal motion template is summarized using image moments where statistical moments of the temporal templates are extracted to represent events. Xiang and Gong (2005, 2006, 2008) used these descriptors for visual behaviour profiling and abnormality detection in an entrance scenario captured with a directed camera located at a close proximity. This approach provides an accurate representation of clearly defined events captured in an uncluttered environment but it relies on the accurate detection of the shape of the moving object.

Motion trajectories have provided the basis for a large body of work on automated surveillance. The trajectories result from tracking the movements of objects over time. Sillito and Fisher (2008) represented trajectories by a vector of cubic spline control points. Basharat et al. (2008) represented object trajectory points  $O_j$  by five dimensional vectors  $O_j = (t, x, y, w, h)$ , where  $t$  is a time stamp of observation,  $(x, y)$  is object location at that time, and  $(w, h)$  is the size of an object. Ouivirach et al. (2013) track multiple objects in the scene using an appearance-based blob tracking algorithm. They implemented procedures for handling stale objects (objects that are inactive for a long time), merging and splitting of the moving objects. In addition to the time, location and object size, trajectory points are represented by the aspect ration of the object's bounding box and its speed. A temporally smoothed version of the object's speed is calculated using:

$$v_j = r \frac{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}}{\delta t} + (1 - r)v_{t-1} \quad (3.3)$$



where  $r$  is a constant,  $\delta t$  is the time difference between the frames at  $t$  and  $t - 1$ . In this approach, the object is tracked over time, and the normalized shape of the trajectory is used to define the events. This approach requires accurate tracking techniques which is still a challenging task in unconstrained environments such as surveillance applications.

To avoid explicit tracking of the objects in the camera view, the motion of all the pixels in the frame can be defined by their optical flow. Andrade et al. (2006) combined optical flow information with the foreground mask and only the flow vectors inside the foreground objects are further considered. Principle component analysis (PCA) is performed on the optical flow fields of each frame to reduce the dimensionality of the features. Optical flow observations are used by Kim and Grauman (2009) where each image pixel is represented by a 9-dimensional optical flow vector comprising of 8 orientations and a speed value. Each video frame is divided into  $u$  by  $v$  sub-regions and the optical-flow within each region is obtained by summing the flow from all the pixels with it. Each frame is represented by a  $9 \cdot u \cdot v$  dimensional activity descriptor. A similar approach is taken by Loy et al. (2010) where optical flow is extracted in each pair of consecutive video frames followed by the decomposition of each frame into  $D$  regions according to spatial-temporal distribution of motion patterns. Each pixel is represented by 4 motion orientations and the co-occurrence histogram of the four orientation is constructed for each region. Optical flow representations are widely used, but caution must be taken as optical flow vectors of each pixel incorrectly represent homogenous regions and the components along the edges are missing due to the aperture problem.

Motion information was computed via spatio-temporal filtering by Zhong et al. (2004). Motion information in each frame  $I_t(x, y, t)$  is found by the convolution of the image with a temporal Gaussian derivative filter  $G_t = t e^{-\left(\frac{t}{\sigma_t}\right)^2}$  and the spatial smoothing filter  $G_{x,y} = e^{-\left(\frac{x}{\sigma_x}\right)^2 - \left(\frac{y}{\sigma_y}\right)^2}$ . Video frames are represented by a

spatial histogram where the frame is divided into a spatial grid and the motion in each grid is accumulated into histogram bins. Another type of representation is used by Mahadevan et al. (2010), where events are represented by the mixtures of dynamic textures (MDT) combined with the discriminant saliency criteria for spatial abnormalities. A dynamic texture is a generative model for both the appearance and the dynamics of video sequences. This type of representation is valuable in crowded environments where objects are not separable from each other.

In the baseline implementation of the unusual event detection system, the focus is on meeting the conditions for a real-world unusual event detection system and interconnection of the various processing blocks. Descriptors chosen for events in the baseline unusual event detection system are spatio-temporal representations obtained by accumulating motion estimates over the frames and concatenating them temporarily. The motion in video segments is estimated by modeling pixels using a mixture of Gaussian (MOG) background modeling technique proposed by Stauffer and Grimson (1999). Each pixel in the frame is modelled using  $K$  weighted Gaussian distributions that represent the intensities of that pixel. The pixel values that have low probabilities of being generated by the created model are declared as foreground. The probability of observing the pixel value  $X_t$  at a time  $t$  is evaluated over all Gaussians for this pixel:

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} * \mathcal{N}(X_t, \mu_{i,t}, \sigma_{i,t}) \quad (3.4)$$

where  $K$  is the number of Gaussian distributions,  $\omega_{i,t}$  is the weight estimate,  $\mu_{i,t}$  is the mean, and  $\Sigma_{i,t}$  is the covariance matrix of the  $i^{th}$  Gaussian at time  $t$ . The parameters of the Gaussian distributions are updated at a run-time for each pixel

value  $X_t$  as follows:

$$\begin{aligned}
\mu_{k,t} &= (1 - \rho)\mu_{t-1} + \rho X_t \\
\sigma_{k,t}^2 &= (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \\
\omega_{k,t} &= (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t})
\end{aligned} \tag{3.5}$$

where  $\alpha$  is a learning rate;  $M_{k,t}$  is 1 for the model which matched, and 0 for the remaining models. A match is defined as a pixel value within 2.5 standard deviations of the distribution.

The run-time update allows the model to adapt to gradual changes of the environment. The foreground pixel values are thresholded over each frame:

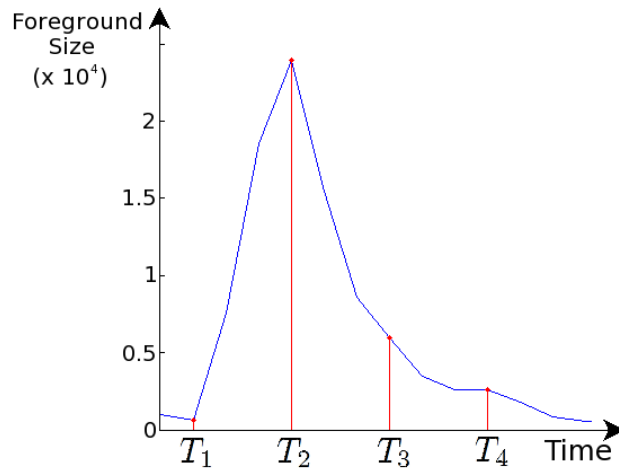
$$X_{m,t} = P(X_t) \leq m_{th} \tag{3.6}$$

where  $X_{m,t}$  is the binary motion mask with 0 representing no motion and 1 representing motion pixels;  $m_{th}$  represents the threshold applied to the probabilities of each pixel to belong to the background. Accumulated motion in each frame is used to represent the frame:

$$f_t = \sum X_{m,t} \neq 0 \tag{3.7}$$

where  $f_t$  is the number of foreground pixels in the frame at the time  $t$  and  $X_{m,t}$  is a binary motion mask. The final representation of each frame by accumulated motion estimation is achieved by combining the motion identified using MOG approach and accumulated over entire frame. Using this representation both spatial and temporal information is captured and has low computational complexity. This representation is suitable to differentiate between activities such as entering, exiting, running, loitering, or fighting. Figure 3.2 illustrates a sample feature vector from a typical indoor surveillance scenario. A single segment of 15 seconds

long captures a person crossing the corridor. The peak in the signal representation depicts the person appearing close to the camera, and the decreasing slope shows the person moving into the distance from the camera. A sequential combination of frames can differentiate between information such as direction of a person walking through the corridor, speed and amount of time spent in the scene. The



(a) Video Features



(b)  $T_1$

(c)  $T_2$

(d)  $T_3$

(e)  $T_4$

Figure 3.2: 15 seconds video segment of a person walking through the corridor. It starts with opening of the doors ( $T_1$ ). As a person walks away from the camera ( $T_2$  to  $T_3$ ), the distance from the camera is represented in the curve.

descriptor carries no information about spatial location of the object or number of objects acting in the scene and this limits this approach to the representation of scenes where a single activity is present at each moment in time. Therefore, a more complex environment would require a more detailed representation of the visual scenes.

### 3.2.3 Training

Based on the representation extracted from the raw video data, events are modeled using machine learning techniques. The techniques adopted in the literature for unusual event detection can be divided into semi-supervised and unsupervised. Supervised approaches are adopted for the detection of events that are known in advance. These approaches require well defined scenarios with a set of good representative examples to train the models. Due to these requirements, these approaches are not usually applied for unusual event detection.

A semi-supervised approach is used by Xiang and Gong (2005, 2008), where a model of the behaviour classes is created using Multi-Observation Hidden Markov Model (MOHMM) where a number of hidden states for each hidden variable is set to the number of event classes. A number of classes is predefined but the training samples belonging to each of the classes are found in an unsupervised way via clustering. Andrade et al. (2006) applied the same learning principle, where a spectral clustering algorithm is used to gather video segments into groups and a MOHMM is applied to train the models representing a normal event and a blocked-exit event. Sillito and Fisher (2008) applied a one-class learning algorithm based on Gaussian mixture model (GMM) where at the early stage of training the underlying distribution is modeled by placing a Gaussian kernel function on each training data item. When a new data item arrives, a new component is added to the model and a pair of components that are most similar are merged. Similarly, Sudo et al. (2008) created a model of the usual class using one-class support vector machine (SVM). The One-class SVM approach yields a discrimination axis that maximizes the distance of all samples from the origin except outliers. Another semi-supervised training approach is used in (Hamid et al., 2005), where a set of activities are considered as an undirected edge-weighted graph with  $K$  nodes,

each node representing an activity descriptor. Activities that do not belong to the trained model are defined as unusual.

Fully unsupervised approaches are different from semi-supervised approaches as they do not assume any knowledge about the training dataset except the assumption that only a small percentage of the data belong to the unusual class. A spatio-temporal saliency in the video is defined as an unusual event by Boiman and Irani (2005). They applied a graph based Bayesian inference algorithm. With every new data instance  $y$ , the joint likelihood  $P(x, y) = P(y|x)P(x)$  is estimated between the new element and the model.  $P(x)$  is estimated non-parametrically directly from the data. Kim and Grauman (2009) used a mixture of probabilistic principal component analyzers (MPPCA) to identify the typical patterns. A space-time Markov random field (MRF) is defined in terms of two functions: the node evidence and the pair-wise potential. The inference on the graph yields the maximum a posteriori (MAP) that specified which nodes are abnormal, as computed by maximizing:

$$E(x) = \lambda \sum_i n(x_i) + \sum_{i,j \in \text{neighbour}} \rho(x_i, x_j) \quad (3.8)$$

where  $n(\cdot)$  is the node evidence function, and  $\rho(\cdot, \cdot)$  is a pair-wise potential function. The value  $\lambda$  is a constant to weight the node evidence, and  $x_i$  denotes the label telling whether the node is normal or abnormal. Basharat et al. (2008) applied a GMM to model the motion trajectories extracted from the training dataset, and the data that have low probabilities to be part of the model are identified as atypical. Each component of the mixture is modelled as a Gaussian distribution of the form:

$$p(\gamma|\Theta_i^j) = \frac{1}{(2\pi)^{d/2} |\Sigma_i^j|^{1/2}} e^{-1/2(\gamma - \mu_i^j)^T \Sigma_i^{j-1} (\gamma - \mu_i^j)} \quad (3.9)$$

where  $d$  is the dimensionality of the model and  $\Theta_l^i = \mu_l^i, \Sigma_l^i$ . An unsupervised approach is applied by Zhong et al. (2004), where clustering based on the co-occurrence matrix is applied to model the entire dataset without separation between training and testing data. A bipartite graph co-clustering, similar to the normalized cuts algorithms used in image segmentation, is applied to the data. Isolated clusters are identified as unusual events. Another clustering approach is used for unusual event detection by Breitenstein et al. (2009), where agglomerative clustering algorithm of 10000 clusters is used to model the usual scenes and the unusual scenes are identified by measuring and thresholding their distance to the model.

The approaches to model events can also be classified based on the way the training data is used to train the model. There are two main groups of approaches, namely batch and online approaches. In more recent works, incremental updates into the learning process are integrated (Sillito and Fisher, 2008; Xiang and Gong, 2008; Sudo et al., 2008; Kim and Grauman, 2009; Breitenstein et al., 2009). A mixture of batch and online approaches is used by Loy et al. (2010) and Ouivirach et al. (2013). In the combined approach the initial classifier is trained using a small set of labelled samples from the known classes, and then further updated in an online manner with the unknown data. Xiang and Gong (2008) showed that when incremental learning is applied, the classification results are less than a percent lower. A CCTV camera was mounted on the ceiling of an office entry corridor, monitoring people entering and leaving an office area. The identification of six activities, such as entering from few different entrance locations and exiting by entering the lab or through the door at the end of the corridor, showed results as high as 80%. Sillito and Fisher (2008) reported accuracies as high as 90% on identifying outlier trajectories of people walking in the carpark. Loy et al. (2010) used a more complicated MIT traffic dataset defined in (Wang et al., 2009b) and reported accuracies of around 70%. It can be seen that the results do not suffer

when the online training is applied, but the results are highly dependent on the complexity of the dataset. Unusual events in crowded scenes were analyzed by Nam and Hong (2014) and Huo et al. (2014), and accuracies reported varied from 75% to 99% depending on the complexity of the scenarios. Because accuracy measures are highly dependent on the complexity of the dataset, many researchers choose to report only qualitative results (Mudjirahardjo et al., 2013; Matsugu et al., 2011; Breitenstein et al., 2009).

Considering the nature of the surveillance data where information about usual and unusual events is not easily obtained in advance, and huge amounts of data have to be processed, an unsupervised on-line training approach is preferred. Based on the accuracies of the results reported, in the baseline unusual event detection framework an online agglomerative clustering algorithm is applied to model usual scenarios. Deviations from the model are considered to be the unusual events. The algorithm is adapted from Breitenstein et al. (2009) where it was used to learn the model of usual scenes of Times Square in New York. This approach does not rely on an a priori knowledge and works well with non-stationary data (Guedalia et al., 1999).

The model is represented by  $K$  clusters. The parameter  $K$  is chosen base on the complexity of captured environment. The more diverse the captured scenes are, the more clusters are required for the model. Each cluster  $k$  is defined by the parameters  $\mu_k$ ,  $n_k$  and  $a_k$  corresponding to the cluster centroid, element count and age respectively. The clustering is implemented in the following steps:

1. Initialize first  $K$  clusters using the first  $K$  data points  $x_1, x_2, \dots, x_K$ . Parameters of each cluster  $k$  are initialized to the values:  $\mu_k = x_k$ ,  $n_k = 1$  and  $a_k = 1$ . The rest of the data  $x_l$ , where  $l = \{K + 1, \dots, N\}$ ,  $N$  is the number of instances, and  $K$  is the number of clusters in the model (note that first  $K$  elements are already used), is processed as follows:



2. Form a new cluster from the  $x_l$  data point and initialize its parameters:

$$\mu_{k+1} = x_l, n_{k+1} = 1 \text{ and } a_{k+1} = 1$$

3. Find the redundant clusters whose representation of the data is the most similar:

$$\{i, j\} = \arg \min_{i, j, i \neq j} \|\mu_j - \mu_i\| \quad (3.10)$$

4. The cluster with the smaller weight  $w$  (lets assume it is  $i$ ) is merged into the other cluster ( $j$ ) by updating its centroid  $\mu_i$  and count  $n_i$ :

$$\begin{aligned} \mu_j &= \mu_j \cdot (1 - \alpha) + \mu_i \cdot \alpha \\ n_j &= n_j + n_i \end{aligned} \quad (3.11)$$

where  $\alpha$  is the learning rate which controls the adaptation speed to the latest observations;  $w_i$  is the weight of the cluster and is defined by a ratio between the number of elements that matched the cluster  $n_i$  and the age of that cluster  $a_i$ :  $w_i = \frac{n_i}{a_i}$ .

5. The weaker cluster ( $j$ ) is removed from the model.

6. set  $l = l + 1$  and go to 2

In this clustering approach, cluster centers are only removed when a nearby cluster center exists, and the removed cluster center represents something that has not been observed for a long time ( $a_k \gg n_k$ ). The cluster center that is distant to every other center remains in the model. Removal of the old and rare clusters can be implemented by eliminating the clusters with weights less than a threshold  $th_w$ .

To make sure that the model is not affected by unusual instances, a threshold is introduced before step 2. If the minimum distance between the cluster centers  $\mu_k$  in the model and the data point  $x_l$  is higher than the threshold  $\arg \min_k \|\mu_k - x_l\| >$

$th_{a_r}$ , then the model is not updated with this data point and the data point is marked as an outlier. The threshold can be chosen empirically, but for a real-world application it needs to be set automatically.

### 3.2.4 Classification

To decide whether an unseen feature vector represents an unusual event is left to the classification step. In the unusual event detection system, if the model of unusual events is created using the clustering approach, the decision is made by thresholding the distance from that model to the unseen data (Zhong et al. (2004), Hamid et al. (2005), Sudo et al. (2008) and Breitenstein et al. (2009)). If a statistical model is used, the maximum log likelihood is used for classification (Boiman and Irani (2005), Xiang and Gong (2005), Andrade et al. (2006), Sillito and Fisher (2008) and Ouivirach et al. (2013)). The distance and the log-likelihood measures need to be thresholded to acquire binary result. Most of the unusual event detection approaches do not produce a binary results (Dee and Hogg (2004), Zhong et al. (2004), Sudo et al. (2008)). Some approaches calculate the threshold from the data (Andrade et al. (2006), Breitenstein et al. (2009), Ouivirach et al. (2013)). All the rest of the methods find the threshold through experimentation.

In real-world surveillance scenarios, it is preferable to avoid setting a static threshold. In the baseline unusual event detection system, the distance between the model created by the online agglomerative clustering algorithm and the new data instances is evaluated for decision making. The system is implemented with adaptive calculation of the threshold which was proposed by Breitenstein et al. (2009).

### 3.3 Dataset

Different datasets are available for testing algorithms for unusual event detection. Some unusual event detection algorithms are tested on outdoor video data such as traffic datasets (Zhong et al. (2004), Basharat et al. (2008), Breitenstein et al. (2009), Loy et al. (2010)), pedestrian dataset (Ouivirach et al., 2013), and web-cam footage overlooking a lake (Breitenstein et al., 2009). Indoor datasets, used for unusual event detection, range from subway staircase video footage (Kim and Grauman, 2009); video of the activities in a dining room (Zhong et al., 2004); entrance to a restricted area (Xiang and Gong, 2005, 2008); docking area of a retail bookstore (Hamid et al., 2005); footage of people playing cards (Zhong et al., 2004); a collection of well defined actions acted by a single or multiple persons (Boiman and Irani, 2005), Sudo et al. (2008). Most of the surveillance-like datasets are captured and annotated specifically for the experiments. Sharing of those datasets is often restricted by privacy protection laws due to the appearance of ordinary people in the footage.

Table 3.2 lists the datasets commonly used by researchers as benchmark datasets for testing event classification algorithms. TRECVID datasets contain videos from half a minute (BBC stock shots) to 48 minutes (recordings of lectures). The AVSS dataset contains fully annotated data of multiple scenarios approximately half an hour each. The scenarios are: abandoned baggage in the metro station, parked vehicle on the road from a far-view camera, audio-visual people detection and tracking dataset captured in the lecture rooms of Queen Mary University of London, and face recognition dataset. and PETS datasets are provided for participants of the yearly conferences for challenges in motion tracking in a single or multi-camera set-ups and detection of objects in challenges related to visual surveillance. Scenarios of interest in those datasets are: left-luggage, detection and tracking scenes on water, tracking of football players, smart meetings, indoor

Name	Details	Reference
TRECVID (2001-2007)	Broadcast news videos, airport surveillance videos	(TRECVID, 2007)
AVSS 2007	Abandoned baggage, parked vehicle, audio visual people, face datasets	(AVSS, 2007)
AMI 2007	Meeting environment, different roles in team, creating projects	(AMI, 2007)
OTCBVS 2007	Person detection in thermal imagery, facial dataset, weapon detection, thermal and color fusion	(OTCBVS, 2007)
PETS (2000-2006)	Outdoor people and vehicle, indoor people tracking, hand postures, smart meetings, facial expressions, gestures, football data, tracking scenes on water, detection of luggage events in public spaces	(PETS, 2006)
CANDELA (2003-2005)	Abandoned objects, people hanging out in livingroom; pedestrians crossing the street	(CANDELA, 2005)
CAVIAR (2003-2004)	People walking alone, meeting with others, window shopping, entering and exiting shops, fighting and passing out and last, but not least, leaving a package in a public place...	(INRIA, 2004)
UCSD 2010	Crowded pedestrian walkway with unusual event such as a cart, wheelchair, skateboarder or a bicycle passing the pedestrian flow	(UCSD, 2010)
I-LIDS 2011	Abandoned baggage, parked vehicles, doorway surveillance, sterile zone monitoring, multiple camera tracking	(U.K. Home Office, 2011)

Table 3.2: Freely available computer vision datasets

and outdoor people tracking, vehicle tracking. As most of the benchmark datasets, the data is separated into individual clips approximately half a minute long. The OTCBVS dataset contains far-view footage of pedestrians captured using standard and thermal cameras. The dataset contains clips captured at three different locations, and the fully annotated clips are approximately half a minute long. The AMI dataset contains video and audio data collected in a meeting environment. An approximate duration of the clips is 30 minutes, and the collection has 100 hours of meeting recordings in total. Project CANDELA has publicly shared datasets from indoors and outdoors for detection of people and cars. UCSD is

the benchmark dataset introduced by University of California at San Diego for outlier motion detection. I-LIDS dataset comprises of approximately 24 hours of sequences recorded in different conditions such as different time of the day, weather, background activity levels. This dataset matches the test requirements for the baseline unusual event detection system, but due to licensing restrictions it was not available during research. The experiments on this dataset are conducted towards the end of the project. None of the rest of the publicly available datasets are suitable for experimental purposes for the proposed baseline unusual event detection system. Even though various environments are captured in the available datasets to make them appropriate for comparing different systems, the datasets are the collections of individual video clips up to 30 minutes long. To test the proposed baseline unusual event detection system a continuous data stream is required of ideally one full day of continuous data for training and few hours of data for testing. A custom data capture infrastructure was built to evaluate the baseline unusual event detection system. The infrastructure is depicted in Figure 3.3. The infrastructure consists of a camera placed in the corner of the corridor so that the entire corridor could be captured. Wide angle field of view ( $140^\circ$ ) of the camera lenses provides slightly distorted view of the captured area but allows to cover the entire corridor. Example video frames can be seen in Figure 3.4 and the specification of the camera can be accessed from the manufacturers website<sup>1</sup>. Illumination changes caused by the weather captured through the windows and the movement outside the windows creates a challenging environment for any type of visual analysis but provides a suitable representation of real-world surveillance video data where such challenges are often present. The data was collected over a period of 30 days in August, 2009, captured 24 hours a day, at 15 frames per second frame-rate. The dataset is also challenging due to the amount of data that needs to be processed. The size of the frames was set to a standard frame size

---

<sup>1</sup>[http://www.axis.com/products/cam\\_212/index.htm](http://www.axis.com/products/cam_212/index.htm)

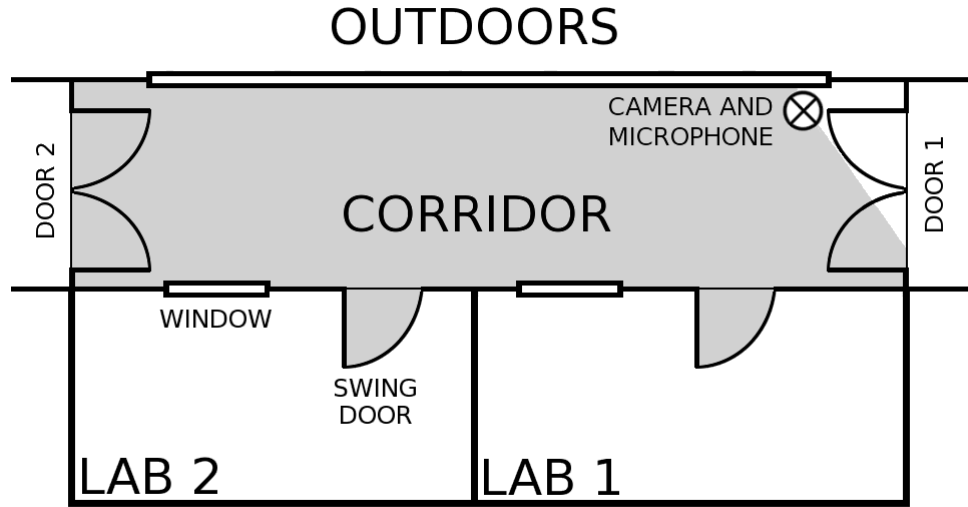


Figure 3.3: The capture environment. Camera location is indicated by  $\otimes$ ; the grayed area represents camera field of view; there are four ways to access the area - doors at both sides of the corridor and the two entrances to the laboratories.



Figure 3.4: Sample video frames from one day

of  $640 \times 480$ . To comply with privacy rules, signs indicating live data capturing were placed at the entrances to the monitored area.

To lower the computational overhead, the original data is sub-sampled to 1 frame per second. Five days of captured data were used to train the model (120

hours) and three days (72 hours) were used for testing. In total, 691.2 thousand of frames were used in experiments. The weekends were excluded from the experiments because the data capture location was almost empty during those days. Cross validation was not performed for the baseline experiments, but the detail examination of each step in the algorithms is performed in the further experiments. Testing data included unusual and antisocial events of interest that were performed and simulated by actors.

### 3.4 Evaluation of Results

The system parameters are set to the following values. The temporal segment size is set to be  $T_n = 15$ , which corresponds to 15 seconds of the video data (at the rate of 1fps). The segment size is chosen by visually evaluating the duration of visual events in the captured environment. It takes 15 seconds for a person to cross the corridor by walking. The events lasting longer than a single segment are captured by concatenating overlapping consequent segments. The overlap between consecutive segments is set to be  $\frac{2}{3}T_n$ . A motion threshold  $m_{th}$  is set to be 0.15 which is empirically chosen through experimentation. Motion threshold is the parameter that contributes to the sensitivity of the overall system and has to be chosen carefully. The number of clusters constituting the model representing usual scenarios is set to be 100. This number is also chosen through empirical tests. The effect of different parameters on final results is evaluated in later chapters of the thesis where more detailed evaluation of each processing step is provided. Event detection is performed on the feature vectors representing overlapping temporal data segments. The distance between the model and the test data instances is binarized with the threshold inferred from the data itself. To count the number of unusual events detected by the system the consequent

segments are grouped into events and the event is marked as a true positive if at least 50% of its frames fall into the segments marked as unusual.

To evaluate the proposed baseline unusual event detection system, its ability to detect the ground-truth events is assessed. The ground-truth data consists of 16 unusual or antisocial scenarios performed by 10 people. The scenarios covered events such as running, jumping onto the window-sill, fighting, etc. The full list of the scenarios and the detection results can be seen in Table 3.3. The ground-truth unusual scenarios occupied 26.65 minutes of the overall testing data.

Event detection accuracy is calculated based on the assumption that the ground-truth event is said to be detected if at least 50% of the event frames fall into the segments classified as unusual. Fourteen out of sixteen predefined unusual events were correctly identified as unusual by the algorithm, therefore 88% accuracy is reported. Table 3.3 shows the summary of the ground truth events and the results of the unusual event detection system where the symbol ✓ means that the event was correctly identified as unusual, and the symbol ✗ means that the event was missed by the system. Two events that were not detected are *running to the lab* and *a person standing on his head*. Further examination of these events showed that running was performed at a very slow pace and was not so different from walking which is a usual event in the captured environment. The second event was missed due to the event representation used. The descriptors used in the experiments could not differentiate between somebody standing on their feet and standing on their head because the information of the pose is not included. In addition to the ground-truth events that are detected as unusual, 300 additional unusual events were also detected, which is approximately 4 events per hour. This shows that even though most of the interesting events are detected, the false positive event detection rate is too high for real world environment. The investigation of the falsely detected events showed, that these events are identified mostly due to illumination changes caused by the weather conditions,



Scenario	Duration	Result
Two people fighting	15sec	✓
Someone putting a poster on the wall	4min 15sec	✓
Shouting and fighting	15sec	✓
Climbing on the window sill	10sec	✓
Running to the lab	30sec	✗
Tearing the poster from the wall	40sec	✓
Bullying/intimidation	1min 05sec	✓
Waving to the camera	15sec	✓
Attempt to enter a laboratory	30sec	✓
Football in the corridor	1min 05sec	✓
Bringing in a ladder, climbing on it	5min 30sec	✓
Arguing near the lab door	10min 50sec	✓
Running through the corridor	10sec	✓
Someone standing on his head	30sec	✗
Leaving something in the corridor	9sec	✓
Removing something from the corridor	30sec	✓

Table 3.3: List of scenarios and detection results

namely sunny days with clouds. The reason why the illumination changes were identified as unusual events is that very different weather conditions were present during the training period. To address this issue, training could be performed over days with more varying weather conditions. Furthermore, descriptors that include intensity information would help to eliminate most of the false positives.

### 3.5 Conclusions

A real-world surveillance environment poses requirements that are addressed in the proposed baseline unusual event detection system. The proposed baseline unusual event detection system is deliberately simple compared to other approaches. It is focused on specific aspects of the overall pipeline yet showed promising event detection results when tested on surveillance-like dataset created specifically for the evaluation of the system. Two ground-truth unusual events were missed, and the false event detection rate was approximately four events per hour. Clearly, the

event descriptors are over-simplistic and might fail in scenarios where the scenes are not restricted to having a single activity in the camera view. The following experiments in this thesis consider each part of the system to identify the most relevant methods.

### **3.6 Summary**

In this chapter, a complete unusual event detection pipeline was introduced and applied to visual surveillance data that meets the requirements of real-world settings. An unsupervised training technique with an on-line learning capability is applied to model the environment. An adaptive thresholding technique is applied to detect unusual events. The promising results showed that 88% of the ground-truth interesting events were detected using simple spatio-temporal visual features. A relatively high false alarm rate obtained is determined to be due to the overly simplistic descriptors. The focus of the next chapter is on finding suitable video event descriptors for surveillance applications.

# Chapter 4

## Visual descriptors for event representation

### 4.1 Overview

Visual event detection accuracy depends on the descriptors extracted from the sequences of video frames. In order to describe video events, continuous video data must first be segmented into temporal segments followed by the extraction of descriptors from each identified segment. The process of extracting descriptors can be further divided into two steps. First, *objects* or *interest points* have to be identified, followed by extraction of discriminative representation for each of them. In this chapter, the definition of events is first formulated, and a taxonomy of various event description approaches is discussed. Local spatio-temporal region properties are argued to be a suitable representation of visual events in unconstrained visual environments. As a result, five local-region description methods are compared, and the best performing one is identified as the most suitable for targeted applications. Experiments are also conducted to test the temporal segmentation techniques. Based on the overall experimental results,

the most suitable visual event description scheme to identify events in visual surveillance applications is proposed.

## 4.2 Definition of Visual Event

Understanding of visual events is a research topic that has received much interest in recent years. Visual events are those high-level semantic concepts that humans perceive when observing a video sequence. The major challenges in this research area are defining what a visual event is and translating raw video data into semantically meaningful descriptions of the defined events so that it could be easily classified using machine learning techniques. The meaning of the term “visual event” can be ambiguous and depends on the context. Three characteristics that define a general visual event in different application domains were defined by Lavee et al. (2009):

1. Visual events occupy a period of time;
2. Visual events are built of smaller building blocks;
3. Visual events are described using the salient aspects of the video sequence input;

Taking into account this abstraction, in a particular visual event each of these qualities can be explicitly instantiated. For example, in a gesture recognition task, motion of a hand can be treated as a single visual event. In visual surveillance applications, an event could be as specific as “opening the door”, or as abstract as “antisocial behaviour”.

Ahad (2011) proposed a hierarchical taxonomy of visual events where the events are grouped depending on the duration of time they occupy. Figure 4.1 visualizes a variant of this taxonomy where the three main groups of visual events are identified. The groups are visualized on the time axis (x-axis). The duration of

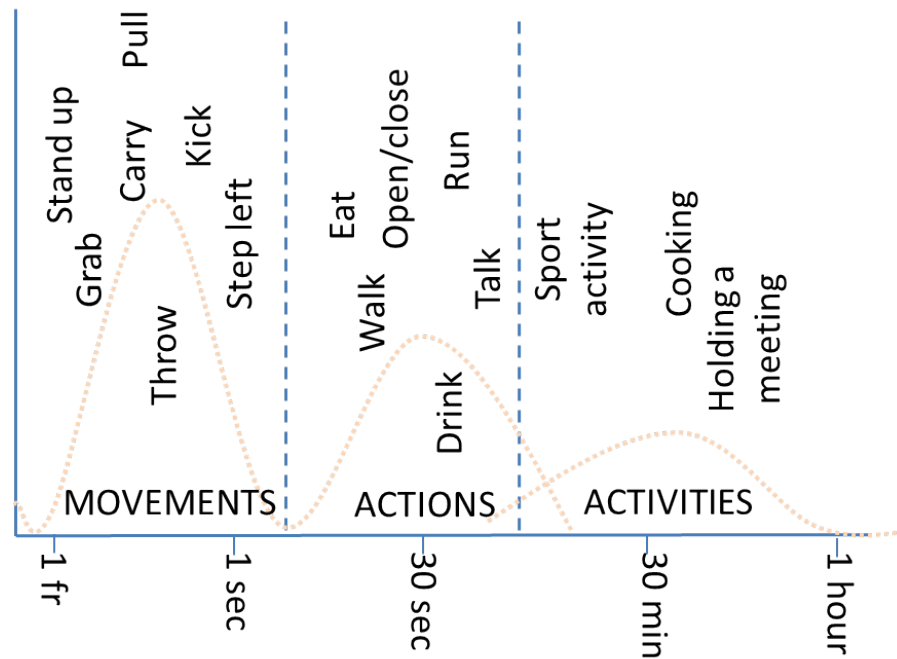


Figure 4.1: Event hierarchical taxonomy

the visual event starts from a single frame up to the duration of an hour. Based on this classification, the *movements* are visual events of duration lasting a multiple of a second. Examples of such visual events are *grab*, *kick*, *stand up*, *pull*. Visual events that require more than a second to identify them are classified as *actions*, e.g. *walk*, *talk*, *eat*, *write*. Finally, the *activities* are visual events that last minutes up to an hour, e.g. *holding a meeting*, *cooking*, *sport activity*. In the Figure 4.1, the dotted Gaussian curves represents a distribution of each class, where the movements are more concentrated around one second and the activities and actions have overlapping definitions. In this thesis, the term *event* describes a visual event that last from few seconds to few minutes and could be actions such as *walking*, *fighting*, *falling on the floor* or *drawing on the wall*.

Descriptors for events can be roughly classified into *pixel-based* and *object-based*. The *pixel-based* descriptors are defined by pixel-level primitives such as color, texture or gradient of the color intensity. The primitives are extracted

from each pixel or a pixel group and are also called interest points or interest regions. The *object based* events can be distinguished by the meaningful grouping of neighbouring pixels, sometimes called blobs, and are described by object-level primitives such as size, shape, trajectory.

Pixel-based descriptors do not attempt to group pixel regions into blobs or objects, but compute features based on the salient pixels or regions of the input video sequence. This abstraction approach usually does not allow for a straightforward semantic interpretation, but has the benefit of being general and can be used to describe any type of event. Two common implementations are available for event representation using pixel-based abstraction. One approach is grid-based, and the other is interest point or region detection based on their saliency. In the grid-based method, each frame of the segment is divided into rectangular or cuboid cells of equal size and the features are extracted from each cell and concatenated. Zhong et al. (2004) represented each frame by a histogram, where each bin represents a cell in the grid and the value of the bin represents the amount of motion in that cell. The grid based approaches are sensitive to position and time shifts. The grid also requires predefined temporal and spatial dimensions of the cuboid cells. Therefore, the duration of the event and the dimensions of the objects acting in the events cannot vary within action classes. One solution to make the method invariant to location and time shift is to extract descriptors at multiple spatial and temporal scales. However, with each additional scale, the dimensionality of the descriptor increases and consequently the complexity of the classification. On the other hand, local region based descriptors have been successfully used in image processing for object detection. Lowe (2004) introduced a scale invariant feature transform (SIFT) image description method that is robust to a range of affine distortion, change in 3D viewpoint, addition of noise and change in illumination. This approach and its variations applied to image classification and object recognition outperform many other descriptors (Mikolajczyk and Schmid, 2005).

Object-based descriptors are based on the intuition that a description of the objects participating in a video sequence is a good intermediate representation for event reasoning. Thus, a low level input is abstracted into a set of objects and their properties. These properties include information such as speed, position or trajectory. Object detection and visual tracking are active research areas in the computer vision community (Stauffer and Grimson, 1999; French, 2005; Han et al., 2008). The object-based descriptors work well when describing events in well defined visual scenes, where the motion of each object is spatially distinguishable from each other. Foresti et al. (2002) proposed an event detection and indexing system where human actions such as entering a restricted area, are represented by tracked blobs and related features. Sillito and Fisher (2008) represented events such as walking, falling down and fighting by motion trajectories of a moving object. In their approach, the trajectories are approximated using uniform cubic B-Spline curves in order to make the descriptions consistent in all the events. Basharat et al. (2008) defined pedestrian motion paths as events and described them by their motion trajectories and statistics such as location, size of an object and speed. The drawback of the object-based descriptors is that they assume a complete and at least sufficiently isolated moving object visible in the scene. This assumption does not hold in many real-world surveillance applications. In visual surveillance applications where diverse visual environments are captured, object detection and tracking are often not possible due to the complexity of the background, crowding, remote data capture or low quality of the video data. In these situations, the shapes of the objects or their trajectories can be extracted only with limited accuracy. Therefore, pixel-based descriptors are commonly the preferred approach to represent unconstrained visual data and are preferred in this work.

In the video domain, local spatio-temporal descriptors are proven to be a useful representation of video events in unconstrained visual environments such

as movie videos (Klaser et al., 2008; Kovashka and Grauman, 2010; Wang et al., 2012; Marín-Jiménez et al., 2013). The properties of local descriptors such as robustness to pose variations, occlusions and object variations suggest their suitability to represent visual events in surveillance applications. Local descriptors are extracted based on three main steps. Detection of local interest point or region, description of the local region and aggregation of the descriptions of the local regions to acquire a final representation of the event. The processing methods of each step are discussed in the following sections.

#### **4.2.1 Detection of Space-Time Local Regions**

Surveillance videos are usually of lower quality due to the limitations of data storage availability and network data transfer capacity. Thus, the objects acting in the events and their shapes can be identified with limited accuracy. In the unconstrained environment, where more than one object is moving in the captured scene and it is difficult to accurately extract objects, spatio-temporal local regions identified based on saliency in spacial and temporal dimensions have been proved to be a stable representation. Figure 4.2 shows an example of the local spatio-temporal points detected by evaluating cornerness and the change of the motion direction. The interest points are showed using superimposed circles centered around those points. The size of the circle represents a scale of the interest point. Their capability to represent events is proved in recent applications such as a single person action recognition (Laptev and Lindeberg, 2003; Klaser et al., 2008; Bregonzio et al., 2009b; Kovashka and Grauman, 2010), activities of multiple people (Chakraborty et al., 2012) and event classification in movies (Klaser et al., 2008; Wang et al., 2012; Marín-Jiménez et al., 2013). Laptev and Lindeberg (2003) introduced a method for detection of space-time local regions, and a detailed analysis of the techniques can be found in the doctoral dissertation by Laptev



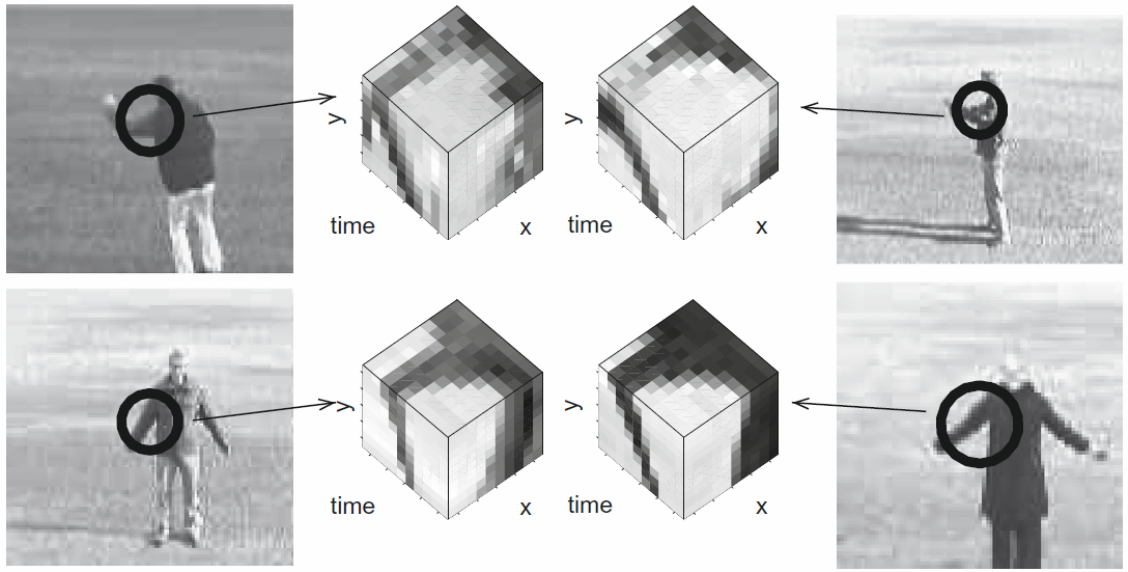


Figure 4.2: Space-Time Local Regions (Laptev and Lindeberg, 2006)

(2004). The proposed method is an extension of the interest point detection scheme from single images to video data based on the Harris corner detector (Harris and Stephens, 1988) and is called the *Harris3D* interest point detector. To find *Harris3D* interest points, space-time gradients  $L$  are acquired by the convolution of Gaussian kernel  $G_{x,\sigma^2,\tau^2}$ , where  $\sigma$  is the spatial scale and  $\tau$  is a temporal scale of the video frame. The space-time Gaussian kernel is defined as:

$$G_{x,\sigma^2,\tau^2} = \frac{1}{2\pi\sigma^4\tau^2} e^{-\frac{x^2-y^2}{2\sigma^2} - \frac{t^2}{2\tau^2}} \quad (4.1)$$

Interest points are set to the local maxima of the cornerness criterion  $H$  based on the spatio-temporal second moment matrix  $M$ . The corner function  $H$  is defined by detecting a spatial maxima of the Harris function  $H = \det(M) - k \cdot \text{trace}^2(M)$ . Schuld et al. (2004) successfully applied *Harris3D* interest point detection to identify single person activities. Kovashka and Grauman (2010) also used the *Harris3D* interest point detection algorithm and tested different description schemes for the interest points. Klaser et al. (2008) applied the *Harris3D* interest point detector to represent single person activities and movie scenes. This approach can discrimi-

nate amongst the behaviours that are characterized by the reversal of direction of motion that gives rise to the spatio-temporal corners (Dollar et al., 2005). Example behaviours could be walking, jogging, clapping or waving. In certain problem domains where the motion of an object does not create spatio-temporal corners, for example, facial expressions, or an object moving at a far distance from the camera, interest points detected using this method are sparse and do not provide sufficient information to discriminate between different actions.

An alternative interest point detector is proposed by Dollar et al. (2005) and is based on the notion that too many features are better than too few. The interest point detection method is called *Gabor3D* due to the utilization of the Gabor filters. In this method, 2D Gaussian smoothing functions  $G$  are applied to the spatial dimensions, and the two 1D Gabor filters  $H^{ev}(t; \tau, \omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2}$  and  $H^{od}(t; \tau, \omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$  are applied to the temporal dimension. The  $\omega$  and  $\tau$  parameters correspond to spatial and temporal scale of the detector. The response function has the form  $R = (I * G * H^{ev})^2 + (I * G * H^{od})^2$ . Dollar et al. (2005) tested *Gabor3D* interest point detection method on event classification in facial expression domain, mouse behaviour analysis and human activity identification. Bregonzio et al. (2009b) applied the same detector as part of a single person action recognition process with the resultant accuracies above 90%. The *Gabor3D* interest point detection method extracts sufficient amount of interest points, but the scales of the detectors have to be determined by the user and the features are not scale-invariant.

Willems et al. (2008) proposes an efficient way to extract scale-invariant interest points. In this work, a *Hessian3D* interest point detector is introduced which is a spatio-temporal extension of the Hessian blob detector proposed by Lindeberg (1998). The saliency of the gradients is given by the determinant of the 3D Hessian

matrix of second derivatives:

$$H_{x,\sigma^2,\tau^2} = \begin{pmatrix} L_{xx} & L_{xy} & L_{xt} \\ L_{yx} & L_{yy} & L_{yt} \\ L_{tx} & L_{ty} & L_{tt} \end{pmatrix} \quad (4.2)$$

where  $L_{xx}$  is the convolution of the Gaussian second order derivative with the frame  $I$  at point  $x$ . In comparison experiments conducted by Stöttinger et al. (2011), the *Hessian3D* interest point detector outperformed the *Gabor3D* and *Harris3D* detectors. It was shown to be superior in repeatability, variations of scale, rotation and addition of noise tests.

To improve video event identification capabilities, Wang et al. (2009a) proposed a densely sampled interest point detector (*Dense*). The proposed technique samples interest points in 5 dimensions  $(x, y, t, \sigma, \tau)$  where  $x$  and  $y$  are the spatial coordinates,  $t$  is temporal coordinate, and  $\sigma$  and  $\tau$  are the spatial and temporal scales, respectively. The spatial and temporal samplings are done with a 50% overlap providing an abundant amount of interest points. Wang et al. (2009a) showed from the experimental results that *Dense* interest point detector outperforms *Harris3D*, *Gabor3D* and *Hessian3D* when human actions are considered in realistic setups.

Inspired by the success of the *Dense* interest points detector and the efficiency of the video event representation using motion trajectories, Wang et al. (2011) introduced an interest point detection technique based on dense trajectories (*DenseTraj*). In this approach, each point  $P_t$  is tracked independently in a number of spatial scales by applying optical flow  $\omega = (u_t, v_t)$  and the median filter kernel  $M$ :

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + M * \omega|_{\bar{x}_t, \bar{y}_t} \quad (4.3)$$

where  $(\bar{x}_t, \bar{y}_t)$  is the rounded position of  $(x_t, y_t)$ . The trajectory length  $L$  is limited to avoid drifting from the initial locations. As soon as the trajectory reaches the length  $L$ , it is removed from the tracking process. A new track is initialized if no tracking points are found in a  $W \times W$  spatial neighbourhood. The descriptors are computed within a space-time volume around the trajectories, rather than the interest points as in the previously mentioned methods. The *DenseTraj* detector is robust in the presence of fast, irregular motions that are caused by noise, and in Wang et al. (2012) is shown to outperform the *Dense* methods.

## 4.2.2 Descriptors of Local Regions

To describe local regions around interest points extracted using the SIFT method, Lowe (2004) proposed to sample image gradients and orientations around the keypoint location and to create orientation histograms over  $M \times M$  sample regions. Example of this descriptor can be seen in Figure ?. The example shows the spatio-temporal interest point (STIP) descriptors where each interest point is represent by a vector, those vectors are then grouped to form a dictionary, and the classification step is showed as a comparison with the dictionary. Each region is represented by

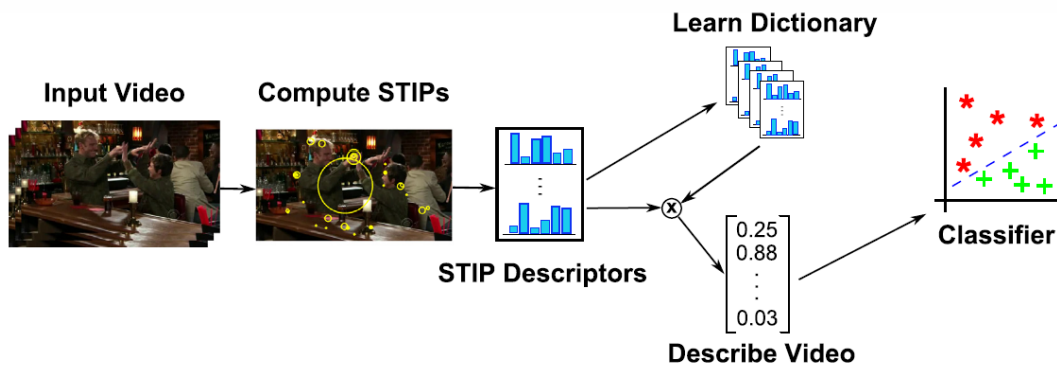


Figure 4.3: Descriptors of interest points (Marín-Jiménez et al., 2013)

a histogram of  $K$  gradient orientations, and the interest points are represented by a feature vector constructed by concatenating the histograms of the regions

$D = (d_1, \dots, d_{M^2K})^T$ , where  $d$  is a histogram bin of gradient orientations. This descriptor was powerful due to its invariance to shift (depending on the size of  $M$ ). The scale and rotation invariance are achieved by adjusting the level of Gaussian blur using the scale of the keypoint and adjusting the gradient orientations relative to the keypoint orientation.

While the SIFT descriptor captures shape information, a method introduced by Laptev and Lindeberg (2006) captures motion information. A Histogram of Optical Flow (HOF) descriptor extracts points' displacement by applying motion analysis technique introduced by Lucas and Kanade (1981b). To get a final descriptor, optical flow is computed from the second-moment matrices around the space-time interest points. The area around an interest point is divided into  $M \times M \times N$  cells, where  $M$  is a spatial and  $N$  is the temporal subdivision. Motion vectors from each cell are represented as histogram entries of the corresponding orientation. Representation of HOF for a region of interest is captured in a feature vector  $D = (d_1, \dots, d_{M^2NK})^T$ , where  $K$  is the number of motion directions. In later work, Laptev et al. (2008) conducted experiments to compare the HOF descriptor with the histogram of oriented gradient (HOG) descriptor that captures spatial information. The results showed that HOF descriptors produce better action classification results.

The shape of a motion trajectory is a frequently used descriptor for event recognition. Local motion patterns can also be represented by the shape of local trajectories, and Wang et al. (2012) proposed such a descriptor. Given a point  $P_t = (x_t, y_t)$  in frame  $I_t$ , its tracked position in the frame  $I_{t+1}$  is smoothed by applying a median filter and points of subsequent frames are concatenated to form trajectories  $P_t, P_{t+1}, P_{t+2}, \dots$ . The length of the trajectories is limited to  $L$  frames. The shape of the trajectory is defined by a sequence of displacement vectors  $(\delta P_t, \dots, \delta P_{t+L-1})$ , where  $\delta P_t = (P_{t+1} - P_t)$ . The final representation of the

trajectory shape is the normalized displacement vector:

$$D = \frac{(\delta P_t, \dots, \delta P_{t+L-1})}{\sum_{j=t}^{t+L-1} \|\delta P_j\|} \quad (4.4)$$

The local trajectory has dimensionality of  $2L$ , where  $L$  is the length of the trajectory, and each point in the trajectory is represented by horizontal and vertical displacements  $(x, y)$ .

Based on the performance of SIFT descriptors for image classification and object recognition and the importance of motion information to describe actions, Klaser et al. (2008) introduced a *HOG3D* descriptor which in addition to shape information captures some motion information. To compute HOG3D descriptor, a local cuboid  $r_s = \{x_r, y_r, t_r, w_r, h_r, l_r\}^T$ , represented by its location in a 3D space  $(x_r, y_r, t_r)$  and size  $(w_r, h_r, l_r)$ , is divided into a set of  $M \times M \times N$  cells, where  $M$  represents spatial subdivision and  $N$  represents the temporal subdivision. Consequently, each 3D cell is divided into  $S \times S \times S$  subblocks  $b_j$ . For each subblock  $b_j$ , the corresponding mean gradient  $\bar{g}_{b_j}$  is computed using integral videos to achieve computational efficiency. The integral video on gradient vectors is computed similarly to integral images reported by Viola and Jones (2001) where each pixel value  $(x, y)$  is replaced with the sum of pixels above and to the left of the location  $x, y$ . The mean gradient  $\bar{g}_{b_j}$  is subsequently quantized and summed over subblocks  $b_j$  to form a histogram  $h_c$ , and the histograms of all cells are concatenated to form a *HOG3D* descriptor  $D = (d_1, \dots, d_{M^2NK})^T$ . Wang et al. (2009a) applied this representation to describe activities in complex visual scenes and is reported to outperform both 2D HOG and HOF descriptors, that capture the gradient or motion information only.

Similarly to HOG3D which, in addition to spatial gradient information, captures motion information by introducing temporal gradient, the motion boundary histogram (MBH) descriptor captures gradient information by taking a derivative

of optical flow. This descriptor is originally introduced by Dalal et al. (2006) for human detection and later adapted for interest point description by Wang et al. (2012). By taking a gradient of optical flow, locally constant motion is removed leaving only the motion boundaries, which makes it robust to camera motion and can correctly describe the motion of the homogenous region. To represent this descriptor as a feature vector, a local spatio-temporal region is divided into  $M \times M \times N$  cells. The histograms of motion boundaries in each cell are concatenated into a single vector  $D = (d_1, d_2, \dots, d_{M^2NK})^T$ . MBH for  $x$  and  $y$  directions are calculated separately, thus two vectors  $D_x$  and  $D_y$  are extracted for a single region. This descriptor is designed to capture relative motion of the moving objects while resisting background motions. Wang et al. (2012) showed better action recognition results using MBH descriptor when compared to HOF and HOG3D descriptors on the Hollywood2 dataset (Laptev and Perez, 2007).

### 4.2.3 Postprocessing of Descriptors

A visual event, represented by a collection of the local regions requires further processing to transform this collection of local region descriptors into a fixed size vector. A fixed size vector is required to provide a meaningful comparison between events for event modeling and detection. The process of mapping the variable dimensional vectors onto the space of the fixed dimensionality is commonly known as vector quantization. The most widely used vector quantization approach in action recognition is bag-of-visual-words (BOVW) (Schuldt et al., 2004; Zhong et al., 2004; Klaser et al., 2008; Mahadevan et al., 2010). This method is adapted from text retrieval where a document is represented by counts of its words that match entries in the vocabulary. In the image or video representation, the vocabulary is called a *codebook* and is populated with *visual words* sometimes also called *prototypes* (Zhong et al., 2004). The BOVW representation is constructed

in two steps. First, a *codebook* is created, followed by the mapping of the local region descriptors to the *visual words* in the codebook. The final representation of the event is formed by assigning each descriptor to the closest visual word in the codebook and accumulating counts of each match in a histogram representation. Each histogram bin represents a visual word in the codebook, and the value of the bin shows the number of local regions that matched the visual word. For BOVW representation, the codebook is usually created using square-error partitioning methods, such as k-means (Duda et al., 2000). This algorithm proceeds by iterated assignments of points to their closest cluster centers and re-computation of the cluster centers. To handle large datasets, Mairal et al. (2009) proposed an online approach that processes one element (or a small subset) of the training data at a time.

Using this approach, each of the salient local regions found in a video segment representing a single event is compared to each of the visual words in the codebook using a distance measure, e.g. Euclidean. The shortest distance between the descriptor and visual word is regarded as an occurrence of the visual word in the event. All the occurrences of the visual words for a particular event are combined into a histogram of  $n_w$  bins. Each bin holds the count of occurrences of each word for the event.

A histogram with  $n_w$  bins is formed from the occurrences of visual words, where the bins hold a count of all visual words that are present in the considered event. The parameter  $n_w$  has to be predefined in advance, and in the event detection literature it is set to a value ranging from 1000 to 4000, depending on the computational power available. Csurka et al. (2004) showed that using 1000 words is a good trade-off between accuracy and speed. Umakanthan et al. (2012) showed that only 3 - 4 % improvement in action recognition performance can be achieved when  $n_w$  is more than 1000.



### 4.3 Comparison of Local Region Descriptors

Comparison of event descriptors for action recognition has been carried out in a number of papers (Wang et al., 2009a, 2012; Umakanthan et al., 2012). The descriptors are compared for classification of activities performed by a single person, e.g. KTH dataset, IXMAS, UIUC dataset; sports activities such as UCF sports, Olympic sports; general activities such as YouTube videos, UCF50 or HMDB51 datasets; or movies such as in the Hollywood2 dataset. Investigation of visual descriptors' performance on surveillance videos, to our knowledge is not reported in the literature. The dataset closest to the complexity of the surveillance applications is the movie dataset (Laptev and Perez, 2007), where data is captured from various camera angles and various distances from the actions of interest. This dataset is used by Wang et al. (2011) to compare interest point detection methods. The results showed *DenseTraj* to be superior when compared with the latest single interest point detection methods from the literature.

Thus, in the experiments reported here, *DenseTraj* is used to find salient local regions due to its superiority among other state-of-the-art methods. The trajectories are extracted on six spatio-temporal grids, and those detected on each grid are treated independently. In order to find suitable descriptors for visual events in surveillance data, a number of local-region descriptors are evaluated. The descriptors and their anagrams are listed in Table 4.1. The HOG descriptors are represented by combining gradient information with orientations of the gradients; HOF descriptors are represented by a combining motion information with the orientation of motion vectors; MBH descriptors are represented by combination of motion information and have the non-boundary motion filtered out; TRACK descriptors are represented by the normalized shape of the motion trajectory of the interest points; ALL descriptors are a combination of HOG, HOF, MBH and TRACK descriptors by concatenating their normalized representations; SIFT

Anagram	Full name	Details
HOG	Histogram of Oriented Gradients in 3D space	Captures gradient information; Klaser et al. (2008)
HOF	Histogram of Optical Flow	Captures motion information; Laptev and Lindeberg (2006)
MBH	Motion Boundary Histogram	Captures boundary motion information; Dalal et al. (2006)
TRACK	Trajectory Shape	Captures the shape of motion of the interest points; Wang et al. (2012)
ALL	Combination of all above	Captures motion and gradient information
SIFT	Scale Invariant Feature Transform	Captures gradient spatial information; Lowe (2004)

Table 4.1: A list of descriptors used in experiments

descriptors are represented by orientations of local gradients. All the descriptors, except SIFT, are extracted from the regions around the dense trajectories (*DenseTraj*). The SIFT descriptor uses a spatial interest point detection scheme as proposed originally by Lowe (2004). The interest points of this descriptor are spatial only. All the descriptors are quantized using bag-of-visual-words (BOVW), where the number of words is set to 1000, which has been shown to be sufficient (Csurka et al., 2004). A Support Vector Machine (SVM) with a radial basis function kernel (RBF) is used to model and classify between two classes - “usual” and “unusual”. The average of 6-fold cross-validation is reported as a final result.

### 4.3.1 Visual Surveillance Dataset

To find a good event representation for unusual event detection, the experiments are performed on a benchmark dataset CAVIAR<sup>1</sup>. The dataset contains a number of video clips with hand labeled object trajectories and metadata describing the scenes at each frame. In the literature, this dataset has been used to test human

<sup>1</sup>Data from EC Funded CAVIAR project/IST 2001 37540, found at URL: <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>

activity recognition techniques. Fernández-Caballero et al. (2012) used this dataset and the provided object trajectories to create a state machine which models object behaviour in the scene. Based on the available ground-truth motion tracks of the subjects, Lv et al. (2004) tested different approaches to detect specific human activities. Kitani et al. (2005) used the dataset to test human activity recognition when activities are modeled using Hierarchical Bayesian Networks. It is used in the work reported here in order to ensure the results are reproducible.

The dataset is composed of video clips that resemble an indoor surveillance environment. Each clip contains acted scenarios such as walking, browsing, people meeting and fighting. The data is captured using a fixed camera sensor and contains multiple instances of each scenario. In addition to the people that are part of the scenarios, the clips contain people that belong to the background, which makes the scenes more realistic. All the short clips are provided with manually collected annotations describing various properties of the scenarios and the locations of objects active in each frame. For the experiments, metadata of the

Appearances	Movements	Contexts	Roles	Situations
appear	walking	immobile	walker	moving
visible	inactive	walking	browser	browsing
disappear	active	browsing	fighters	inactive
occluded	movement	fighting	fighter	joining
	running	drop down	walkers	fighting
		meeting	meeters	split up
		leaving	leaving object	leaving victim
				interacting
				leaving object

Table 4.2: Properties from the CAVIAR video clips' metadata

clips specifies five properties of the scenes: *appearance*, *movement*, *context*, *role* and *situation*. *Appearance*, *movement* and *role* describe the objects visible in the scene. The *context* and *situation* captures the global aspect of the scene. Table 4.2 lists five properties together with the values that those properties can take.

The metadata provided with the clips is adapted to represent “usual” and “unusual” event classes. Based on the general description of unusual events in visual surveillance scenarios, the “unusual” class is represented by the fighting events. The dataset contains four video clips that have fighting scenes. The names of the four clips are listed in Table 4.3, together with their length in seconds and the duration of the scene depicting the fighting activity. As can be seen in the table,

Title	$N$	$N_i$	$P_i$
Fight_Chase	17 sec	5.32 sec	31.3%
Fight_OneManDown	38 sec	7.8 sec	20.5%
Fight_RunAway1	22 sec	8.24 sec	37.5%
Fight_RunAway2	22 sec	7.84 sec	35.6%

Table 4.3: Video clips representing unusual activity;  $N$  - length in seconds;  $N_i$  - duration of fighting activity;  $P_i$  - percentage of frames depicting fighting;

the fighting activity occupies approximately 30% of each video clip. It would be inaccurate to take full video clips to represent the “unusual” class. Therefore, instead of using full video clips, only the frames that belong to the actual fighting events are set to belong to the “unusual” class. After a thorough examination of the data and its annotations, the frames that are labeled to have objects or groups in *fighter* or *fighters* role are chosen. The “usual” class is represented by all the rest of the frames that have people walking, browsing or standing. Examples of the frames depicting both classes are shown in Figure 4.4.



Figure 4.4: Example frames of the “unusual” (top row) and “usual” (bottom row) classes

Six video clips from the dataset are used for the experiments. Four video clips that captured fighting scenes (listed in Table 4.3), and two video clips (originally named Browse1 and Browse2) that capture the usual scenarios such as walking and browsing. The dataset has unbalanced data classes, where 17% of the data belongs to “unusual” class, and all the rest of the data belongs to “usual” class. When the data is divided into the training and testing sets, the constraint is added to keep the original ratios between the classes in both training and testing datasets. Half of the data is used for training, and the rest is used for testing. For cross-validation of the results, the video clips are shuffled six times and the average result is reported.

### 4.3.2 Results

In order to evaluate event descriptors, statistics of Receiver Operating Curves (ROC) and precision-recall curves are evaluated. ROC curve shows the trade-offs that can be made between the true positive rate (TPR) and the false positive rate (FPR):

$$TPR = \frac{TP}{TP + FN} \quad (4.5)$$

$$FPR = \frac{FP}{FP + TN} \quad (4.6)$$

where  $TP$  is the number of frames correctly identified as part of unusual events,  $FP$  is the number of usual frames incorrectly identified as part of unusual events, and  $FN$  is the number of unusual event frames identified as usual. The area under ROC (AUC-ROC) represents the overall performance of the algorithm and provides a single measure that can be used to compare different methods. Precision-recall curve shows the trade-offs between the fraction of detected frames that belong to the unusual events (precision) and the fraction of unusual event

frames detected as unusual (recall):

$$precision = \frac{TP}{TP + FP} \quad (4.7)$$

$$recall = \frac{TP}{TP + FN} \quad (4.8)$$

The area under precision-recall (AUC-precision-recall) is also used to compare the accuracies of the algorithms via a single measure. The classification results depicted via ROC statistics can be see in Figure 4.5. The precision-recall statistics can be seen in Figure 4.6. To simulate the sequential nature of the visual surveillance data, event detection and evaluation are conducted on a frame-by-frame basis. The per-frame classification results, when different descriptors are used

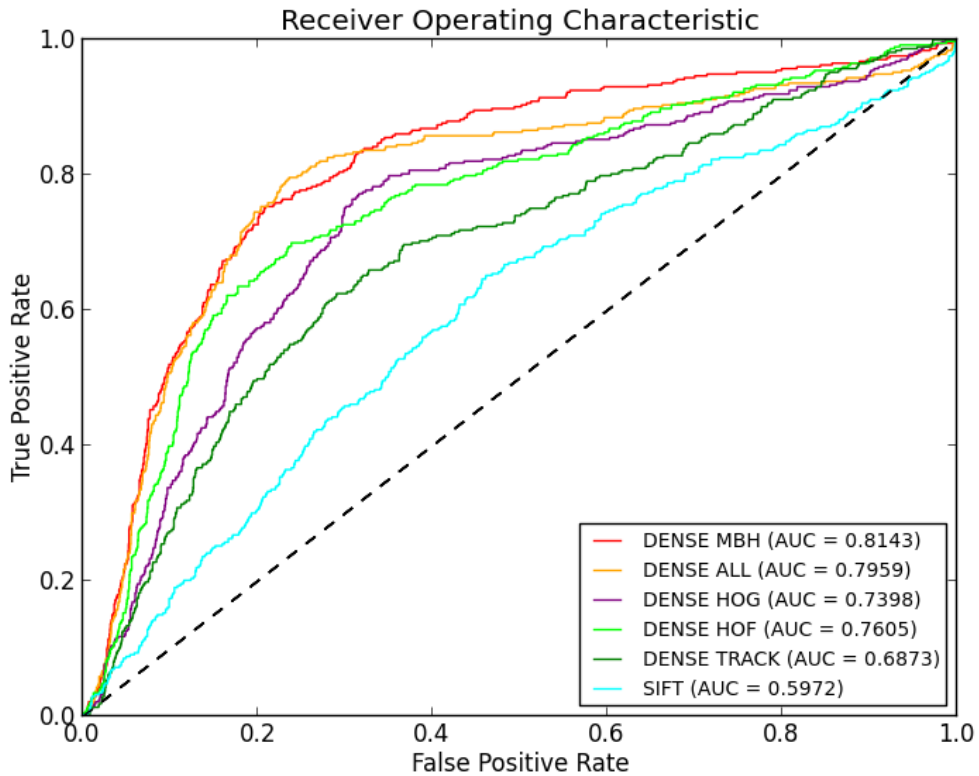


Figure 4.5: Comparison of ROC curves of the unusual event detection results when different interest point descriptors are used

(Table 4.1 lists the descriptors) can be seen in Figure 4.5. As can be seen from

the graph, the *SIFT* interest point detector and descriptors show the lowest accuracy of 59.7% AUC. It is different from the rest of the tested approaches as it examines interest points in each frame independently without taking into account any temporal information between the frames. A very similar descriptor to *SIFT*, but with different interest point detection method is *DENSE HOG*. It shows a 13.2% improvement in accuracy when compared to the *SIFT* approach. When the interest point detection method based on dense trajectories (*DENSE*) is used with different descriptors, the lowest accuracy is 9% higher than that achieved using the *SIFT*. It can be concluded that densely sampled trajectory regions of interest represent events better than the regions of interest extracted around singleton interest points. The overall false positive rates are high for all descriptors because of the per-frame result representation. A single true event can have as many true positives as number of frames it spans. The false positives are treated the same, and a high false-positive rate is due to significantly more negative events in the dataset that can become false-positives.

The lowest accuracy using *DENSE* interest point detection method is achieved when the shape of the point trajectory (*TRACK*) is used. The lower event detection accuracy is attributed to the fact that the trajectory shape information does not carry any information about the gradients around the trajectories. The resulting accuracy is improved by 7.3% when *HOF* descriptors are used. This descriptor is based on motion information, but captures some spatial information by extracting the optical flow orientations. The best performance is observed when *DENSE* interest points are represented using *MBH* descriptors, yielding accuracy of 81.4%. This descriptor captures similar information to the *HOF* descriptors. The improvement is that it represents only the boundary motion by taking a derivative of optical flow and filters out irrelevant motion.

To test if aggregation of different descriptors improves the overall accuracy of the event detection system, the four descriptors representing *DENSE* inter-

est points - *TRACK*, *HOF*, *HOG* and *MBH* - are combined using an early fusion concatenation approach (represented by the *DENSE ALL* acronym). Vector quantization is applied to the concatenated feature vector. The combined descriptor showed a 2% accuracy reduction when compared to the best performing single descriptor. The result implies that additional descriptors do not necessarily provide better accuracy and less accurate descriptors degrade the final accuracy of the detection results.

In addition to the ROC curves, precision-recall curves are also evaluated and depicted in figure 4.6. A precision-recall curves show the resulting ratios

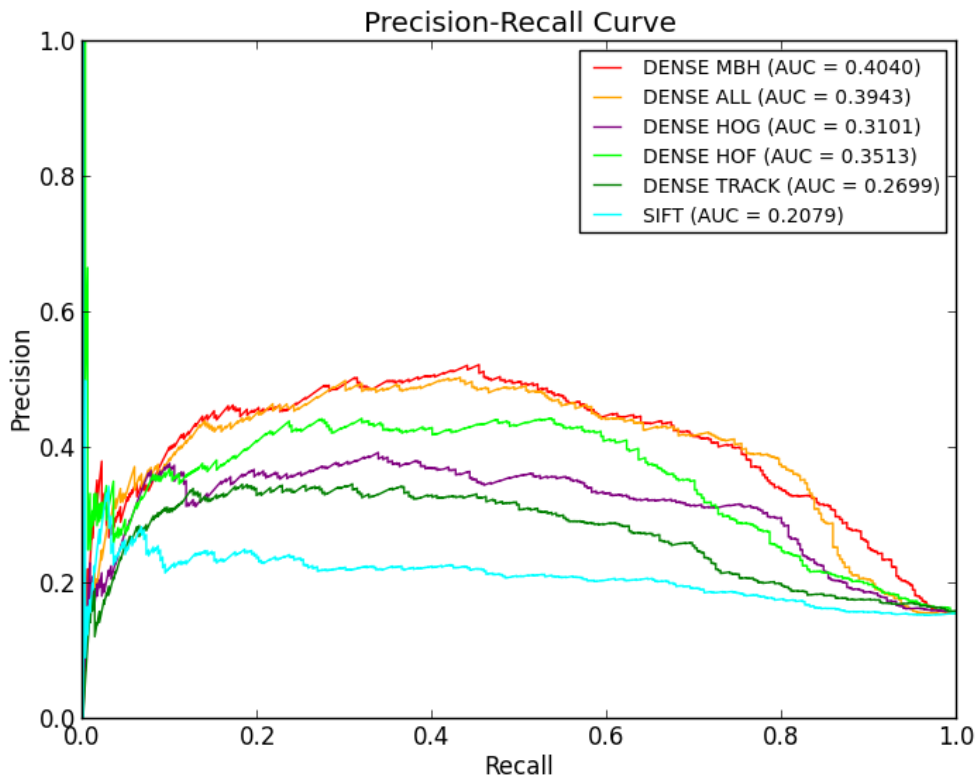


Figure 4.6: Comparison of precision-recall curves of the unusual event detection results when different interest point descriptors are used

of the “*unusual*” frames being correctly classified as “*unusual*” (recall), and the fraction of frames classified as “*unusual*” that belong to the true “*unusual*” events. The graph shows the same trends between different descriptors as in the ROC



statistics (Figure 4.6). The *SIFT* descriptor gives the lowest AUC-PR (area under the precision-recall curve), and the *MBH* descriptor gives the highest accuracy. The accuracy using this metric is much lower for all the methods because all the frames correctly classified as “usual” are not included in the statistics.

In the following experiments, the best performing local region descriptor is used in segment evaluation. *Temporal segmentation* experiments are carried out by fixing the step of the segment to a single frame, and increasing the number of frames in the segment from 1 to 100. In figure 4.7 the ROC curves are plotted when varying the size of the segment. It can be observed that the improvement

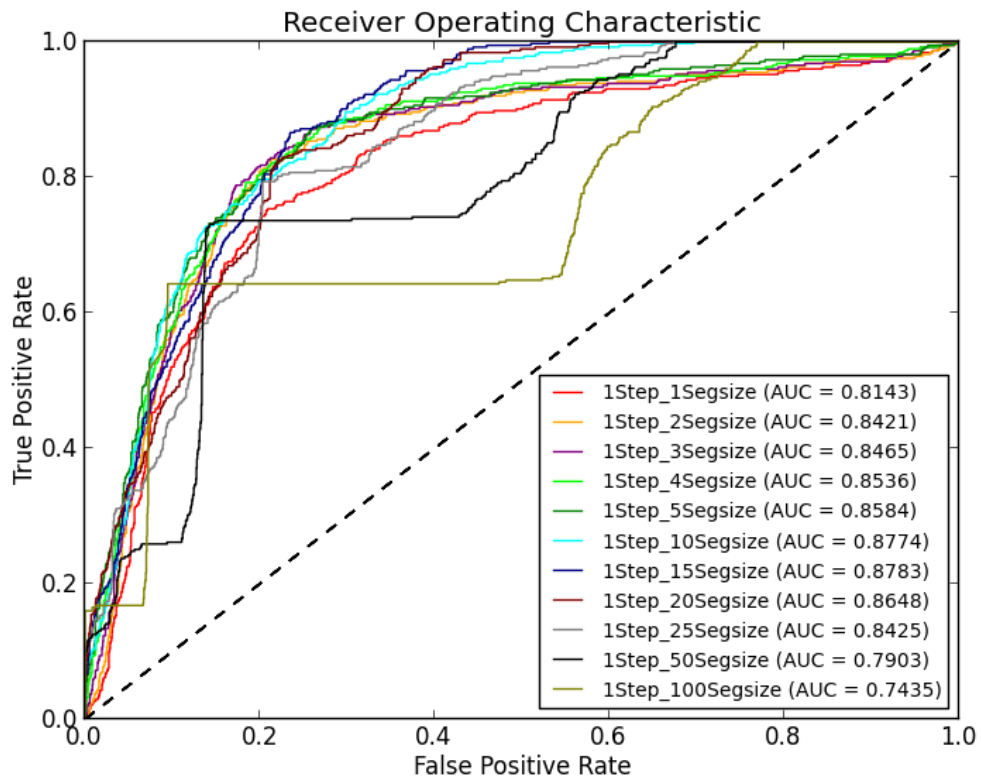


Figure 4.7: Comparison of results using different descriptors to classify fight no fight event using SVM

of up to 6.4% can be achieved in event detection accuracy (represented by the AUC-ROC). The highest accuracy is observed when 15 frames are combined into a single segment, and then starts to drop with the increase of the segment size.

This observation shows that a sufficient amount of information is observed in 15 frames of the action, and a higher number of the frames can smooth the event features and reduces their discriminative properties.

## 4.4 Conclusions

Choosing suitable descriptors is an essential part of all event detection or recognition approaches. The quality of the results is highly dependent on the descriptors' capability to differentiate between events. There is no general approach to determine the best descriptor for all tasks and the performance is highly dependant on application. Visual surveillance applications usually have lower quality video data due to limited storage resources and limited bandwidth. Methods that rely on sharp edges of object shape boundaries may not provide sufficient information to discriminate unusual events. To represent events in an unconstrained environment, local space-time interest points are investigated. In the literature, local regions of interest have been successfully used to represent events in movie datasets. Even though movie videos are usually of high quality, the complexity of the scenes and the amount of clutter and motion present in the videos suggest that the local space-time interest point based representation of the events is suitable for surveillance video data.

A number of descriptors representing regions of interest were proposed in the literature. The aim of the experiments conducted in this chapter is to find suitable descriptors for events in an unusual event detection application. The results show that the descriptors based on motion and gradient information provide superior performance to the descriptors that use only motion or only gradient information. The experimental results show that the motion boundary histogram (MBH) descriptors of the dense trajectory (DENSE) interest points perform better than other descriptors evaluated. The analysis of the results also concluded that

it is useful to aggregate information from consecutive frames to represent them using segments. It was found that by aggregating consecutive frame into a single segment produces a better classification accuracy (15 frames in these experiments).

The experimental results and the review of the literature show that the events in visual surveillance data can be better represented using the following components:

1. Dense trajectory interest point detector;
2. Motion Boundary Histogram descriptor for each local region;
3. Bag of visual words vector quantization;
4. Event representation using overlapping segments (15 frames length and 1 frame shift in these experiments).

## **4.5 Summary**

This chapter outlined the experiments performed on visual event descriptors computed on local spatio-temporal regions. First, the literature has been reviewed, following by the experiments on various local region description methods to find a suitable representation for events in surveillance applications. The best performance was achieved with MBH descriptor that is calculated from the local regions around the dense spatiotemporal trajectories. In addition to the evaluation of the descriptors, experiments were also carried out to evaluate the influence of different segment sizes on the ability to classify the results. The results show that MBH representation aggregated to segments of 15 frames achieved better performance than the other tested segmentations. The suggested event representation is applied in the following chapter to investigate the classification techniques.

# Chapter 5

## Comparison of modeling and classification techniques

### 5.1 Overview

To make event detection algorithms applicable to real-world applications, the algorithms must conform to the constraints that are usually not considered when developing algorithms in a laboratory setting. Three main features of real-world scenarios that introduce these constraints are the unavailability of annotations for training data, huge volumes of continuous video data, and processing data over long periods of time. The first feature is addressed using unsupervised algorithms to learn the model of the events. The second feature is addressed when incremental learning techniques are used. The third feature is addressed by substituting static parameters with dynamic ones. To evaluate methods that take these constraints into account, a comparison of classification capability is first made between an unsupervised method and a state-of-the-art supervised classification method. Then, two types of unsupervised classification approaches are investigated. One is trained using batch processing, and the other method is trained using an incremental approach. In addition to the comparison of

classification approaches, investigation of automatic thresholding techniques, a key aspect of the proposed unsupervised approach, is also carried out.

## 5.2 Motivation

Researchers have designed a variety of models and methods to recognize human activity and interactions, vehicle travel activity or environmental changes. Machine learning techniques used for modeling and identifying these various events can be divided into *supervised* and *unsupervised* approaches, based on how the algorithms are trained.

Supervised approaches are designed to model and identify activities that are known in advance, and require a sufficient number of examples of each activity to train the model. Foresti et al. (2002) trained a neural tree (NT) algorithm with the motion trajectories that were manually drawn on the camera-view for later recognition. The NT model representation is hierarchical, where each node in the modeled decision tree is either associated with one of the neurons, or is a branch terminator associated with one region of the input space. The classification accuracy is shown to decrease with an increasing number of people in the scene due to the difficulties of tracking them. To classify between activities in ballet, tennis and football games, Efros et al. (2003) modeled representations of predefined activities using a k-nearest neighbour (k-NN) classifier. k-NN does not create a model of each activity. It assigns a class label to a new instance based on the labels of the nearest instances in the training set. Danafar and Gheissari (2007) modeled and identified single person activities such as *walking*, *running*, *waving*, etc., using a Support Vector Machine (SVM) classification algorithm. SVM is a two-class classifier which finds the hyperplane that separates training samples in the feature space with the largest margin possible. Multiple binary classifiers have to be trained to apply this method in order to the multi-class problem. Bregonzio

et al. (2009b) performed recognition of single person activities using both k-NN and SVM classifiers. Both approaches showed similar accuracies, which indicates that the data is well separated and does not require complex modeling. Dollar et al. (2005) applied 1-NN with  $\chi^2$  distance and SVM classification approaches to classify single person activities such as *wave*, *run*, *walk*, etc., as well as more diverse recognition tasks such as facial expression recognition and mouse behaviour recognition. The resulting accuracy varied based on the ambiguity of the event. For example, facial expression such as *fear* was identified with an accuracy of 64%, while a *surprised* face expression was identified with 100% accuracy. Wang et al. (2012) applied SVM classification with an RBF- $\chi^2$  kernel to classify human activities in uncomplicated single person activities, and the actions in more complex environments such as YouTube sports videos and movies. The accuracies of the results ranged from 94.2% for activity identification in uncomplicated scenes, to 46.6% for recognition of events in highly cluttered and diverse scenes such as the ones in the movies. Fernández-Caballero et al. (2012) proposed to use finite state machines to model human activity based on the known objects in the scenes and the entrance and exit locations. This approach requires knowledge about the layout of the captured territory which is not available in most surveillance setups. Gao and Sun (2013) used hierarchical Dirichlet process hidden Markov model (HDP-HMM) to learn predefined activities such as *leaving*, *passing*, *wandering*, etc., represented by subjects' motion trajectories. The sequence of motion labels are modeled as Markov chains, and a Dirichlet process is applied to avoid the rapid-switching problem between states. Accuracies above 94% are reported for activities such as *leaving*, *browsing*, *walking*, etc.

It can be concluded, that if the events are visually distinguishable, such as a single person in an empty scene performing *walking*, *waving* or *boxing* activities (Efros et al., 2003; Danafar and Gheissari, 2007; Bregonzio et al., 2009b), the accuracies of the results are close to 95% for most of the supervised classification

approaches. On the other hand, when less defined activities are to be identified, the accuracies drop to 65%, for example, when *mouse drinking* is an event of interest (Dollar et al., 2005), or to 58% when actions in movies such as *answering the phone, hugging or kissing* are of interest (Wang et al., 2012).

Unsupervised techniques are applied when labels of the training data are not available. Ben-David and Lindenbaum (1997) introduces a concept of *learning without a teacher* which is an alternative name for unsupervised learning. Rather than attempting to infer an approximation to the unknown distribution, they propose to settle for the task of learning its high-probability-density areas. Blank et al. (2005) applied spectral clustering to distinguish between activities represented by space-time silhouettes. Events such as *walking, running, waving, etc.*, are clustered into nine groups without providing labels of the actions. Unsupervised learning algorithms are frequently applied for *unusual* event detection. In these scenarios, the data are mostly dominated by *usual* events, while the *unusual* events rarely appear. This constraint allows application of one-class modeling algorithms to model the *usual* activity class as long as the modeling techniques are able to create a multi-modal representation of the data. Andrade et al. (2006) applied a spectral clustering algorithm to identify a number of distinct motion classes that represent different crowd activities. After the grouping, a model is trained via iterative process using separate multi-observation hidden Markov model (MOHMM) for each class. Deviations from the model are defined as *abnormal* activities. A one-class learning approach is also applied by Basharat et al. (2008) to learn the *normal* activities represented by motion trajectories. In this work, a Gaussian mixture model (GMM) is used to create a *usual* model of pedestrian and vehicle paths and the instances that have low probabilities to be generated by the learnt model are declared as *abnormal*. Learning a single class model with a One-class SVM is applied by Sudo et al. (2008) to identify *unusual* video events, and by Lecomte et al. (2011) to identify *unusual* audio events.

Based on how the data is presented to the training algorithm, the machine learning techniques for modeling events can be further classified into *batch* and *online* approaches.

If the learning is performed in a *batch* mode, event detection takes place only after the model is built. Andrade et al. (2006) applied a multi-observation hidden Markov model (MOHMM) to iteratively learn a predefined number of normal sequences that are represented by motion trajectories. The classes of normal trajectories are identified using an EM-based Gaussian mixture model (GMM). The trajectories with low probabilities of being part of the learned models are interpreted as representative of atypical events.

To learn the model gradually with the opportunity to perform event detection after each learning step, an *online* or *incremental* learning version of the algorithm is introduced. Online modeling is frequently used to maintain object appearance during tracking. Han et al. (2008) applied a GMM algorithm to model object appearance, and implemented an online learning procedure to update the appearance of the object while tracking it. In unusual event detection applications, online modeling is also frequently used. Breitenstein et al. (2009) applied online agglomerative clustering algorithm, initially introduced by Guedalia et al. (1999), for unusual scene detection in time-square web-cam images. The model is updated with each new data instance, and the closest clusters in the model are merged forming a hierarchical representation of the sequential data. A number of events such as *rain*, *smoke* or *building tents* in the square are successfully detected as "*unusual*" using this approach. Feng et al. (2012) applied online agglomerative clustering (AGG), similarly to (Breitenstein et al., 2009), but introduced a two layer clustering procedure, where cluster centers resulting from the clustering procedure are further grouped introducing a second level clustering. The experimental results show above 90% detection rate of predefined *unusual* events.



Combination of online and batch processing is applied by Kim and Grauman (2009), where model training is performed using MPPCA, which utilizes the Expectation Maximization (EM) algorithm to learn parameters. The frequency and co-occurrence histograms of Mixture of Probabilistic Principal Component Analyzers (MPPCA) describe typical local activities and their interactions, and are used to establish the Markov Random Field (MRF) model. After the training is done, abnormality levels are inferred while incrementally updating the MRF model.

The last step in event detection is decision making. As the output from the classifier is not a binary value, a decision has to be made what probabilities or distances are used to declare an event *unusual*. Feng et al. (2012) applied two thresholds for anomaly identification: one is a prespecified threshold for a distance from the model, and the second threshold is the intra-cluster distance for the matching cluster. An anomaly is declared if the distance from the new element exceeds the maximum intra-cluster distance. Andrade et al. (2006) defined events *abnormal* if their probabilities are smaller than the minimum likelihood value present in the usual class training set. Similarly, Breitenstein et al. (2009) set the threshold to be the highest value for the shortest distances to the model.

The first constraint of real-world scenarios is that labels for the training data are not readily available to train the classifiers, in particular for unusual events that are very rare. To our knowledge there is no research done on activity recognition when training and testing are performed with videos from different domains targeting surveillance applications. In real-world applications, it is challenging to get annotations for each camera separately. Therefore, an unsupervised learning approach is more suitable than the supervised learning approach. The second constraint is that huge volumes of continuous video data have to be processed on the fly. Batch mode approaches learn the model during the training phase, and the model stays static afterwards. If the batch model is to be retrained to integrate

new data, its computational complexity would gradually increase restricting its usage to finite duration video streams. In real-world scenarios, an online learning approach is the preferred choice to the batch learning. The third constraint is introduced to the real-world scenarios when the parameters defined at the start may become obsolete after running the algorithm for the extended period of time. One of the parameters of event detection systems is a classification threshold. To take into account this constraint, a classification threshold from the data at each processing step could be extracted to allow the value to adapt over time.

The goal of this chapter is to propose an unusual event detection approach that conforms to the three real-world application requirements defined here. The experiments are performed by gradually updating algorithms to integrate all three criteria. This experimental approach facilitates an investigation of the trade-offs of conforming to each criteria independently.

### 5.3 Dataset

Definition of unusual events is challenging due to a wide variety of events that can be characterized as unusual and due to their dependence on the context of the monitored environment. In Chapter 4, the adaptation of the CAVIAR dataset<sup>1</sup> annotations was introduced to accommodate unusual event detection experiments. To evaluate the event classification methods, the same dataset with adapted annotations is used. The difference in these experiments is that the usual event class is represented by six types of events captured in 20 video clips listed in Table 6.1. Table 5.2 lists the unusual events used in the experiments. There are significantly more instances belonging to the usual class than instances of the unusual class events. The imbalance is deliberate so that the dataset would

---

<sup>1</sup>Data from EC Funded CAVIAR project/IST 2001 37540, found at URL: <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>

Clip ID	Title of The Video Clip	Duration
1	Browse WhileWaiting1	31.1s
2	Browse WhileWaiting2	1min 15.2s
3	Browse1	41.8s
4	Browse2	35s
5	Browse3	54.6s
6	Browse4	45s
7	LeftBag AtChair	21.5s
8	LeftBag	57s
9	LeftBag PickedUp	53.6s
10	LeftBox	34s
11	Meet Crowd	19.1s
12	Meet Split 3rdGuy	36.4s
13	Meet WalkSplit	24.4s
14	Meet WalkTogether1	27.7s
15	Meet WalkTogether2	32.5s
16	Rest FallOnFloor	39.7s
17	Rest InChair	39.7s
18	Rest SlumpOnFloor	35.9s
19	Walk2	41.6s
20	Walk3	54.6s

Table 5.1: List of events used for training

Title of The Video Clip	$N$	$N_i$	$P_i$
Fight_Chase	17 sec	5.32 sec	31.3%
Fight_OneManDown	38 sec	7.8 sec	20.5%
Fight_RunAway1	22 sec	8.24 sec	37.5%
Fight_RunAway2	22 sec	7.84 sec	35.6%

Table 5.2: Video clips representing unusual activity;  $N$  - length in seconds;  $N_i$  - duration of fighting activity;  $P_i$  - percentage of frames depicting fighting;

resemble real-world scenarios where unusual events rarely happen. Table 5.3 shows the summary of the training data used to train the classifiers. Usual events together make up of data, or 4292 segments (11 minutes 30 seconds), and unusual events make up 2% of the dataset or 88 segments (14 seconds). Approximately 2% of the training data belongs to the unusual class, and 98% belongs to the usual

	Usual(All the events)	Unusual(Fighting)
Duration	11min 30s	14s
#Segments	17250	350
Percentage	98%	2%

Table 5.3: Proportions of training data-set

class - similar proportions as would be expected in real-world scenarios. This value can be identified by an experienced security personnel.

The same training data is used to train supervised and unsupervised classifiers to ensure a fair comparison between the methods. The difference between the supervised and unsupervised training is that the supervised training algorithm is provided with the data together with corresponding class labels. Training data for the unsupervised method contained both usual and unusual events without giving labels to them. The training data deliberately contains outliers to represent real surveillance training data.

## 5.4 Supervised versus unsupervised training

One of the constraints of real-world scenarios is that labels for the training data are not readily available to train the classifiers, especially for unusual events that are very rare. Unsupervised learning techniques take advantage of the surveillance data property that most of the captured data belongs to the usual class. Using unlabeled data and the assumption that the data is dominated by the usual scenes, unsupervised classification techniques learn the model of usual classes from as much training data as possible and treat the examples that are significantly different from the created model as unusual. Two state-of-the-art algorithms, supervised and unsupervised, are investigated in order to compare their ability to identify unusual events. Support Vector Machine (SVM) is a supervised classification algorithm that is widely used in the event classification

literature (Laptev et al., 2008; Bregonzio et al., 2009a; Wang et al., 2011). If we consider a training dataset  $T$  consisting of  $N$  pairs  $(x_i, y_i)$ , where  $x_i \in R^d$  is the input data ( $i = 1, \dots, N$ ) and  $y_i \in 1, \dots, K$  is the output class label. In unusual event detection task  $K = 2$ . The SVM classifier is defined by

$$k = \underset{j=1 \dots K}{\operatorname{argmax}} f_j(x_i) \quad x_i \in C_k; \quad (5.1)$$

each decision function  $f_i$  is expressed as

$$f_j(x_i) = w_j^T \theta(x_i) + b_j \quad (5.2)$$

where function  $\theta(x_i)$  is a kernel function which maps the original data  $x_i$  to a higher-dimensional space in order to separate classes that are not linearly separable. The margin between classes  $i$  and  $j$  is defined by the relationship  $2/\|w_i - w_j\|$ . The minimization function is defined as follows:

$$\min_{w, b_i} \frac{1}{2} \sum_{i=1}^K \sum_{j=i+1}^K \|w_i - w_j\|^2 + \frac{1}{2} \sum_{i=1}^K \|w_i\|^2 + C \sum_{i=1}^K \sum_{j=i+1}^K \sum_{x_i \in C_{ij}} \xi_l^{ij} \quad (5.3)$$

Here  $\frac{1}{2} \sum_{i=1}^K \|w_i\|^2$  is a regularization term, and  $\sum_{i=1}^K \sum_{j=i+1}^K \sum_{x_i \in C_{ij}} \xi_l^{ij}$  is a loss function used to find the decision rule with the minimal number of errors in the inseparable case.

SVM is an effective algorithm for classification of the data represented by high dimensional vectors. Once the training is finished, the process is memory efficient. It is well suited for unusual event detection task as it can learn from only a few examples, and examples of the unusual events are generally sparse. The drawback is that it requires at least one example of an unusual event to be able to make a prediction. Experimental results when the supervised classification

method is applied to detect unusual events are expected to provide a benchmark accuracy for the unsupervised method.

Gaussian mixture models (GMM) are widely used in data mining, pattern recognition, machine learning and statistical analysis. A GMM is applied to model events consisting of multiple modalities and can be used as a one-class classifier to model the usual class (Porikli and Haga, 2004; Valera and Velastin, 2005; Sillito and Fisher, 2008; Basharat et al., 2008). In many applications, the parameters of GMM are determined by maximum likelihood, typically using the expectation maximization (EM) algorithm (Bishop, 2006). A GMM is expressed in the form:

$$p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \quad (5.4)$$

Each Gaussian density  $\mathcal{N}(x|\mu_k, \Sigma_k)$  is called a *component* of the mixture and has its own mean  $\mu_k$  and covariance  $\Sigma_k$  parameters. The parameters  $\pi_k$  are called *mixing coefficients*. One way to estimate  $\pi$ ,  $\mu$  and  $\Sigma$  parameters is to use maximum likelihood. The log-likelihood function is given by

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k) \right\} \quad (5.5)$$

where  $X = x_1, \dots, x_N$  is a collection of the data samples. EM for GMM proceeds as follows: First, initial values for means, covariances and mixing coefficients are chosen. Then two steps are alternated called expectation step (E) and maximization step (M):

- In step (E), the current values of the parameters are used to evaluate the posterior probabilities:

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)} \quad (5.6)$$

- In step (M), the calculated probabilities are used in this step to re-estimate the means, covariances, and mixing coefficients using the following functions:

$$\begin{aligned}\mu_k &= \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n \\ \Sigma_k &= \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T \\ \pi_k &= \frac{N_k}{N}\end{aligned}\tag{5.7}$$

where

$$N_k = \sum_{n=1}^N \gamma(z_{nk})\tag{5.8}$$

Finally, the log-likelihood is evaluated using Equation 5.5. The next step is to check for convergence of either parameters or the log likelihood. If the convergence criterion is not satisfied, return to step (E).

### 5.4.1 Experimental setup

In the comparison experiments between the supervised and unsupervised classification approaches, the SVM is chosen to represent the supervised methods and the GMM is chosen to represent the unsupervised methods. A Radial Basis Function (RBF) is chosen as a kernel for the SVM method due to its ability to generalize data well and its ability to deal with noise. Parameters  $C$  and  $\gamma$  are set to 1 and  $1/d$  respectively, where  $d$  is the number of features. The GMM classifier is set to have 100 mixtures. As long as the number of mixtures is reasonably big, the inherent clusters of the sample space could be covered with some redundancy without significantly affecting the model. Both the SVM and GMM methods are implemented in batch mode.

Representation of the usual and unusual events is designed based on the results of Chapter 4. First, the video data is segmented into temporal segments of a predefined size. The segments are extracted using the windowing technique with the window 0.6 seconds long and 0.04 second shift. Evaluation of the results

is conducted on a per frame basis rather than per segment to avoid ambiguous representation of the overlapping segments. Each segment is represented by the occurrences of the spatio-temporal regions in the codebook consisting of 1000 instances. Spatio-temporal regions of the segments are represented by the motion boundary histogram (MBH), and the MBH is extracted from multiple scale regions around the trajectories of local points (for more details about the descriptors see Chapter 4).

Segregation of data into training and testing datasets is depicted in Figure 5.1. Both supervised and unsupervised models are trained using 80% of usual data (10 000 segments). The rest of the data is used for testing (2 500 segments). 10-fold cross-validation is performed by randomly selecting training and testing

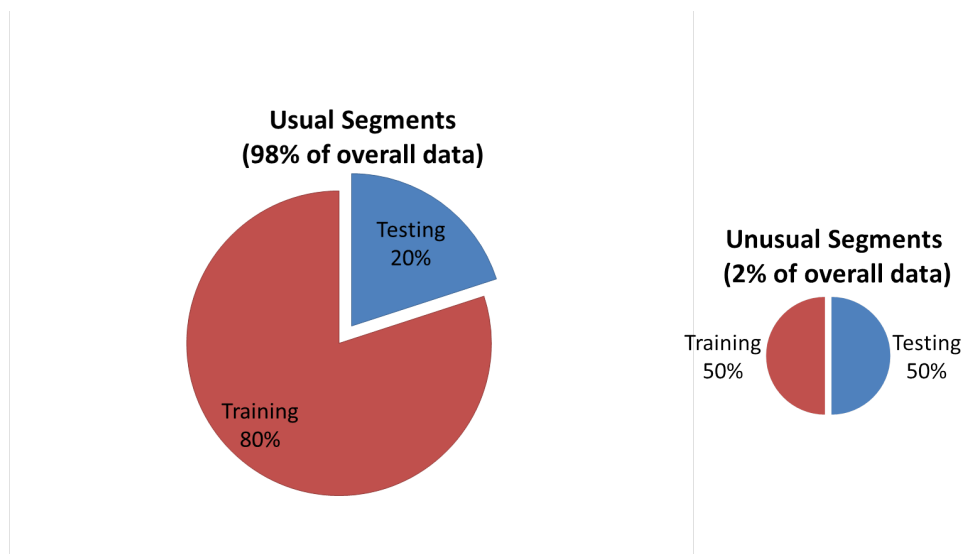


Figure 5.1: Selection of training and testing data for experiments

data from the dataset. The average of the cross-validation results is evaluated and discussed. Results from the state-of-the-art supervised method are expected to provide the benchmark accuracy for the unsupervised method.



## 5.4.2 Evaluation

To compare the classification results of supervised and unsupervised methods, a number of different evaluation metrics are used (Figures from 5.2a to 5.2c). Different metrics help to uncover different aspects of the results. The ROC curves in Figure 5.2a show similar trends of trade-off between true/false detections rates for both SVM and GMM classification approaches, with the supervised learning approach (SVM) performing better by 8% in terms of the AUC-ROC measure. Higher accuracies are expected from the supervised method due to the annotations provided together with the training data. The unsupervised (GMM) method is not provided with any knowledge about unusual events to be detected.

The precision-recall curves in Figure 5.2b show the trade-offs between the precision and recall of instances belonging to the unusual class. It can be observed that the precision of the unsupervised classification results (GMM) reaches a maximum of 0.4. The temporal probability plot reveals why the precision is low. In Figure 5.2c it can be seen that the unsupervised approach assigned high probabilities of being unusual for the frames captured immediately before the fighting event (the area near frame 1000). Visual investigation of the data revealed that the high probability frames represent the activity right before the fight starts where two persons are approaching each other at an increased pace (the snapshots of the scenes can be seen in Figure 5.18 in the section on qualitative evaluation). This is a good example of how the annotation decision affects the results. If the beginning of the fighting event had been marked at the point when the fighters noticed each other, the unsupervised method would have correctly detected the pre-fighting actions as highly probable of being unusual.

Binary classification results can be seen in Figure 5.3. To acquire the binary results, the threshold for the results for each method individually is identified by

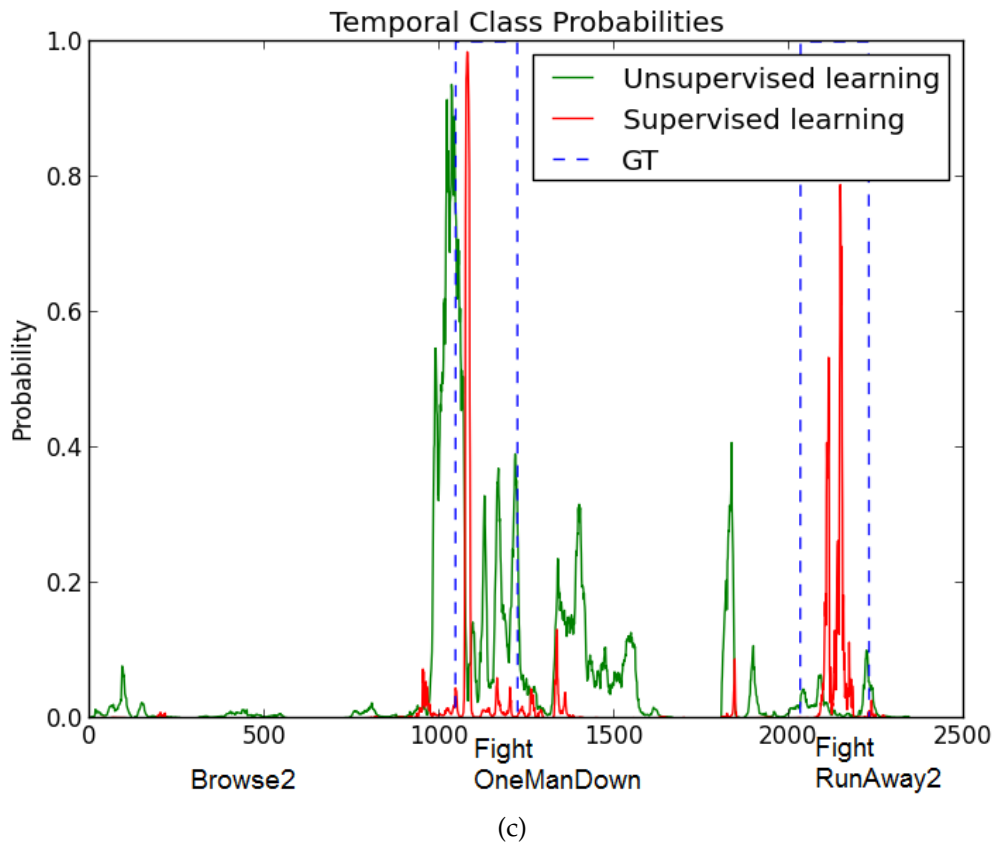
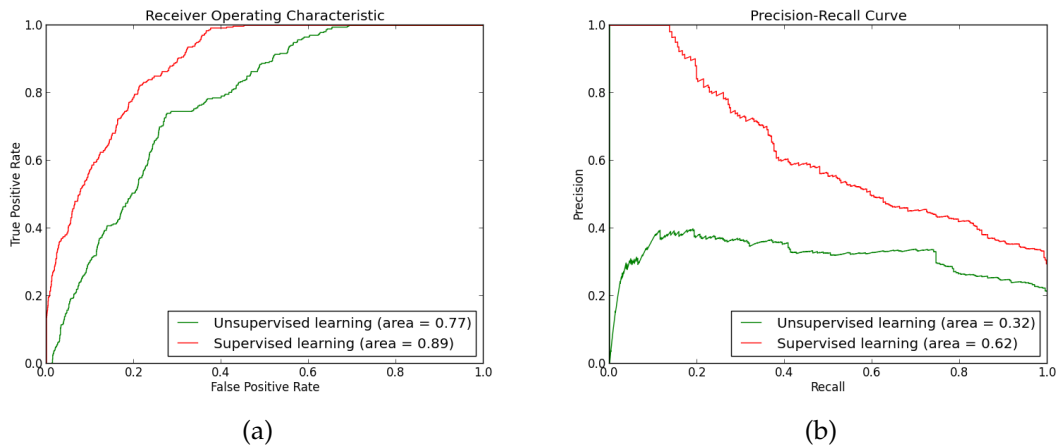


Figure 5.2: Comparison of supervised (SVM) with unsupervised (GMM) learning approaches using different metrics: a) ROC curves b) Precision-Recall curves c) temporal unusual event probabilities

maximizing the  $F1$  score:

$$\arg \max_{th} F1(P > th) \quad (5.9)$$

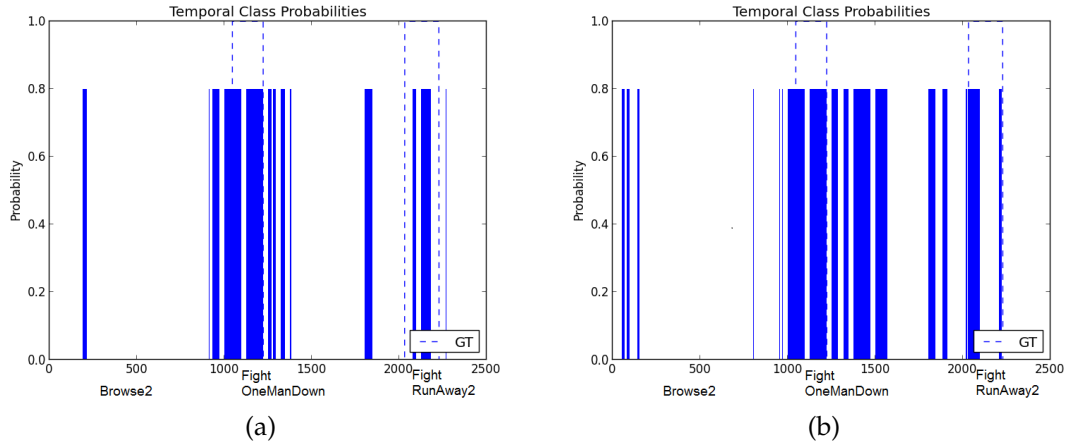


Figure 5.3: Unusual event detection results thresholded using the optimal F1 score for each classification method independently: a) supervised learning (SVM) b) unsupervised learning (GMM)

where  $P$  are the resulting probabilities and  $th$  is the threshold. The  $F1$  score is calculated as follows:

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (5.10)$$

where  $TP$  is the number of frames correctly identified as part of unusual events,  $FP$  is the number of usual frames incorrectly identified as part of unusual events, and  $FN$  is the number of unusual event frames identified as usual. It is interesting to note that when comparing thresholded results of both methods the scenes causing a decrease in precision for the unsupervised GMM classification method are detected as unusual by both unsupervised GMM and supervised SVM classification methods. The only difference is that both methods detected those scenes as unusual with different probabilities. The GMM approach assigned much higher probabilities than the SVM method causing the disparity between the precision-recall curves (Figure 5.2b). When considering the binary results of both methods, it can be seen that the unusual events were detected by both methods equally well, but unsupervised GMM classification results show more false detections compared to the supervised SVM approach. Tables 5.4 and 5.5

show the breakdown of the thresholded quantitative results for the SVM and the GMM approaches respectively. Precision, recall and the F1-scores are recorded in the table for usual and unusual class separately.

Class	precision	recall	f1-score
0	0.94	0.84	0.89
1	0.46	0.72	0.56
avg	0.86	0.82	0.83

Table 5.4: Unsupervised (SVM) classification using optimal F1-score threshold; 0 - usual class; 1 - unusual class

Class	Precision	Recall	F1-score
0	0.94	0.72	0.82
1	0.34	0.74	0.46
avg	0.84	0.73	0.76

Table 5.5: Supervised (GMM) classification using optimal F1-score threshold; 0 - usual class; 1 - unusual class

Precision for the *usual* class (indicated by number 0 in the first column of the tables) is equal for both SVM and GMM approaches. A 0.94 precision value for the *usual* class means that only 6% of the items marked as *usual* are incorrect. Due to the imbalance of the testing dataset and due to the properties of the application, the precision of the *unusual* class is more important than that of the *usual* class. From the precision rates of the *unusual* class (indicated by number 1), it could be said that using both methods more than 50% of the detected instances were not part of the *unusual* events in the ground-truth annotations. This may appear to be an unsatisfactory result, but if the recall of the *unusual* class is considered, which is above 70% using both methods, it can be seen that a significant portion of the overall *unusual* instances has been detected.

Figure 5.4 shows the recall from the two fighting events in the test dataset. The bars show detection results of the ground-truth fighting events where the resulting probabilities are thresholded using the threshold obtained from the

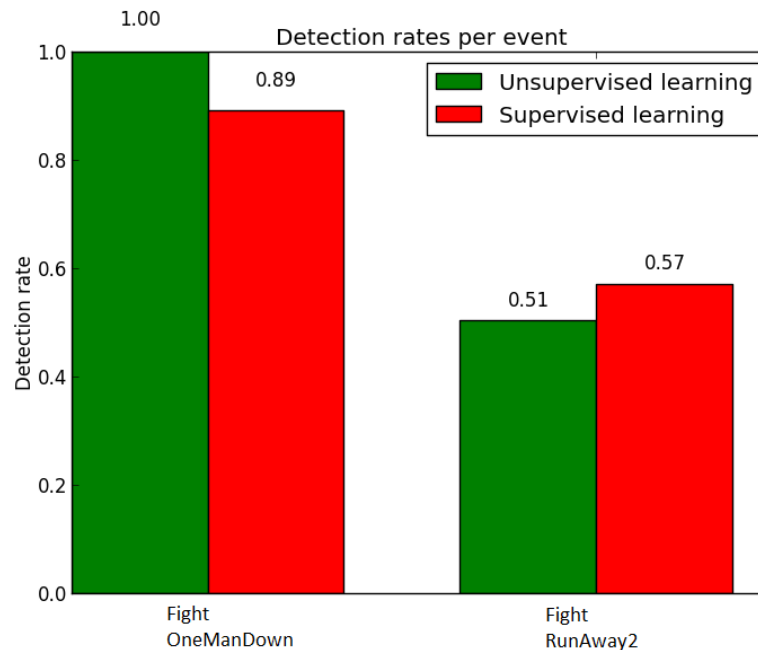


Figure 5.4: Recall of the fighting events

maximum F1-metric (Equation 5.9). It shows results from the event detection point of view where events of interest are either detected or not, and how well each event is detected. It can be concluded that the results are positive as more than 50% of both ground-truth fighting events are correctly identified as unusual, and the first fighting event is almost fully detected by both methods. The Figure also shows that the unsupervised approach has better recall of unusual events (which can also be seen in Tables 5.4 and 5.5) which is at the expense of the higher rate of the false detections. The amount of the false positives (FP) detected using the GMM method was found to be 543 segments or 27% of all negative segments. Using the SVM method 322 segments were falsely detected as unusual, which is 16% of overall negative segments. When temporarily grouped, the segments formed 5 FP events for GMM method and 2 FP events for SVM method. The qualitative evaluation of those events is further analyzed in Section 5.8.

## 5.5 Incremental vs batch learning

In the previous section, the unsupervised event detection approach was compared to the supervised event detection approach, and the results showed comparable accuracy when only events of interest, namely fighting events, were evaluated. The results showed that more than 50% of each fighting event were found to be unusual by both methods. In further experiments, an online learning version of the unsupervised event detection approach is investigated and compared to the batch based GMM unsupervised classification approach.

There are three features of the online learning approach that make it suitable to real-world scenarios. First, the batch processing based approaches have an increasing complexity as more data is used for training and are not scalable for training with the very large amounts of data, which is typically the case in surveillance applications. On the other hand, the online learning algorithm has the same complexity with any amount of data. It trains the model one sample at a time, and after adapting the model with the data sample it throws the sample away.

The second benefit of applying an online learning approach rather than the batch approach is the ability to make predictions from a very early stage of the training phase. Even if early predictions are less accurate and give more false alarms, it may be more acceptable in real-world applications than not getting any results for a period of time while training is taking place.

The third benefit of online learning is its ability to learn continuously and indefinitely. It is not the case with a static model that may become obsolete after some time. The online approach keeps incorporating new data into the model making it more relevant to the current state of the environment as time passes.

Taking all this into account and extending the baseline approach introduced in Chapter 3, an agglomerative on-line algorithm (AGG) is implemented so

that, similarly to the GMM batch processing algorithm, it gathers the statistical information about the data in a number of clusters.

An online approach is implemented based on the agglomerative clustering algorithm (AGG) introduced in Chapter 3. The algorithm is based on the clustering algorithm for non-stationary data proposed by Guedalia et al. (1999). The algorithm is adapted for the targeted environment, therefore the weighting metrics and the cluster fusion approach is modified. A summary of the algorithm is provided next. In this approach, each cluster  $k \in 1, 2, \dots, K$  is composed of three attributes: cluster centroid  $\mu_k$ , element count  $n_k$  and age  $a_k$ . A model is initialized by collecting samples from successive data points until the maximum size  $K$  is reached. To unify the representation, every new sample  $x$  is created as a cluster  $k + 1$  with  $\mu_{k+1} = x$ ,  $n_{k+1} = 1$  and  $a_{k+1} = 1$ . When a new sample comes, it is directly added to the model. Then, the most similar pair of clusters  $(i, j)$  is found by comparing the distance between the  $\mu$  value of every two clusters within the model. Cluster with the smaller weight  $w$  (lets assume its  $j$ ) is merged into the other ( $i$ ). After the merging procedure, the weaker cluster ( $j$ ) is removed from the model to ensure that the model does not increase the size with the time.

The same experimental setup is kept as in the previous experiments. Both GMM and AGG are composed of 100 components, and AGG is implemented with the Euclidean distance.

### 5.5.1 Evaluation

The results of both classification methods can be compared by evaluating the probabilities of the data being usual or unusual, or by evaluating the binary results.

Receiver Operating Characteristic curves evaluation is based on probabilities and it evaluates the relationship between True Positive Rates (TPR) and the False

Positive Rates (FPR). Here, TPR represents segments of correctly detected unusual events, and the FPR represents segments of usual events that were incorrectly detected as unusual. The relationship between TPR and FPR is always positively correlated, meaning that when the TPR is rising, the FPR also goes up. It is desirable to achieve high TPR rates at the same time as the low FPR. In the Figure 5.5a, it can be observed, that if 100% TPR is required, 70% of the detected segments would belong to usual events. Even though the numbers are high, the segments might be temporarily close to unusual events. Temporarily close segments can also be grouped and collectively discarded therefore reducing the final effort of identifying truly interesting events. If more than 50% of the unusual event segments are required to be detected, the GMM approach provides better TPR to FPR ratios by 10-20 %. The same trends can be seen in Precision-Recall Curves (PRC) depicted in figure 5.5b. To achieve a recall higher than 50 %, the GMM approach provides higher precision. On the other hand, if it is sufficient to have a recall below 50%, the AGG approach provides high precision values. The results can be clarified by plotting the probabilities on the time axis as can be seen in Figure 5.5c. In the Figure, the x-axis represents the temporarily ordered segments. The y-axis represents the actual probabilities and the dashed lines mark the unusual events happening. It can be observed, that the with the Incremental learning approach (AGG) probabilities above 50% would give negligible amount of false positive segments (FP), therefore yielding very high precision values. On the other hand, if the threshold would be chosen below 40%, then the number of FP segments is significantly increased. For the batch processing approach (GMM), the threshold at 40% would give high FP rate and low true positive (TP). Thus, when the threshold is 0.05, the TP rate is as high as with AGG method, but the FP rate is significantly smaller.

The results show, that the threshold selection significantly affect the classification results. As before, the threshold yielding the highest F1-metric (Equation



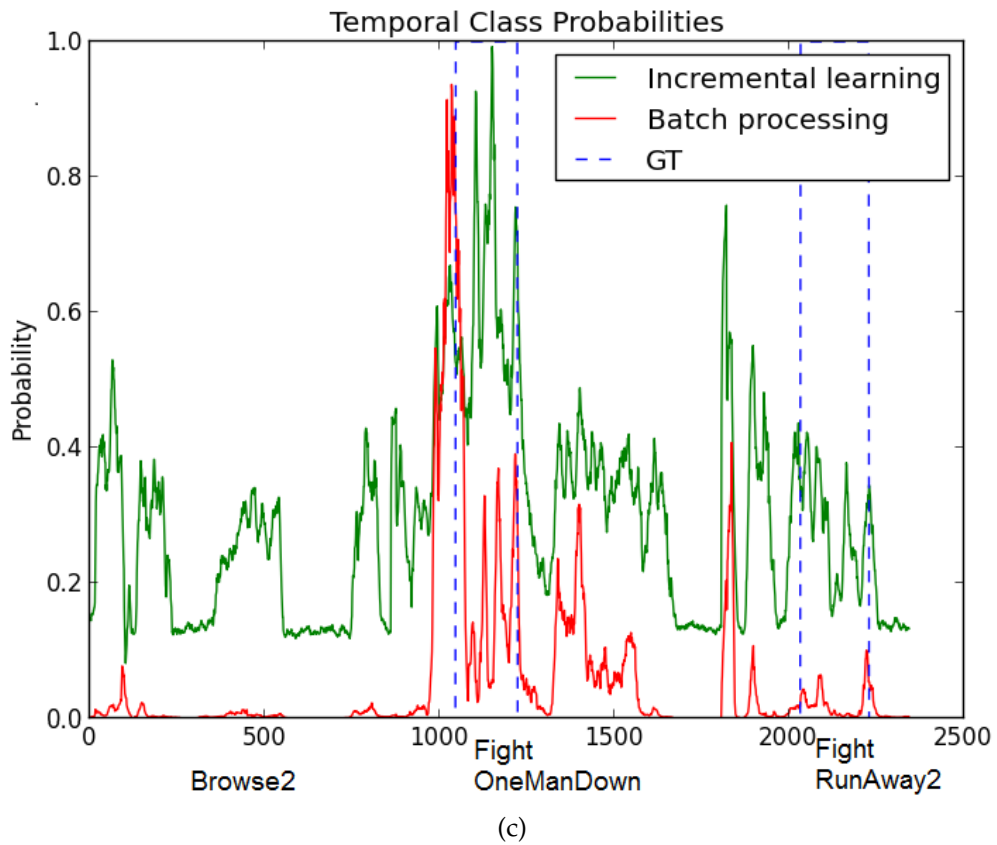
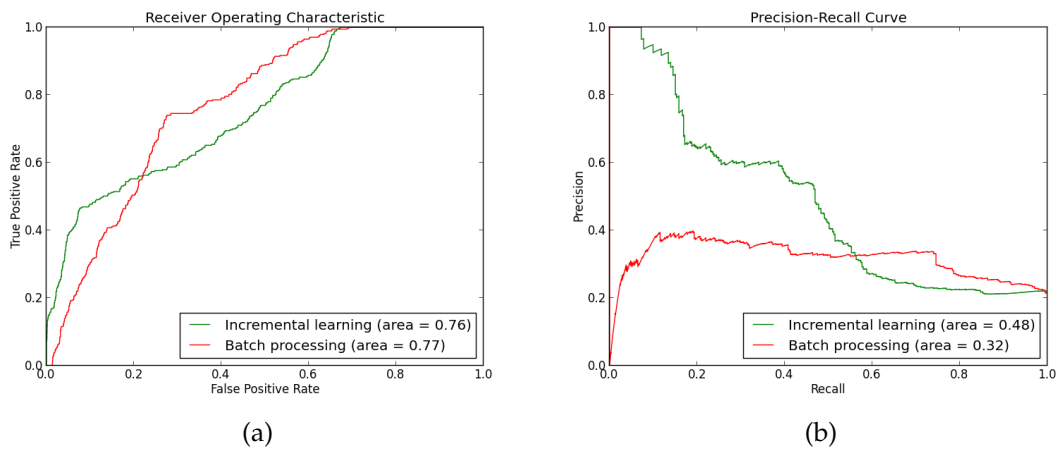


Figure 5.5: Comparison of incremental (AGG) and batch (GMM) approaches using different metrics: a) ROC curves b) Precision-Recall curves c) Temporal unusual event probabilities

5.9) is used to obtain the binary results. Binary classification results can be seen in Figure 5.6. The second fighting event was missed by the online approach (Figure 5.6a) but was partially detected by the GMM approach (Figure 5.6b). Nevertheless,

by comparing the two Figures it can be noticed that the batch approach (GMM) produced significantly more false detections than the online approach (AGG). The

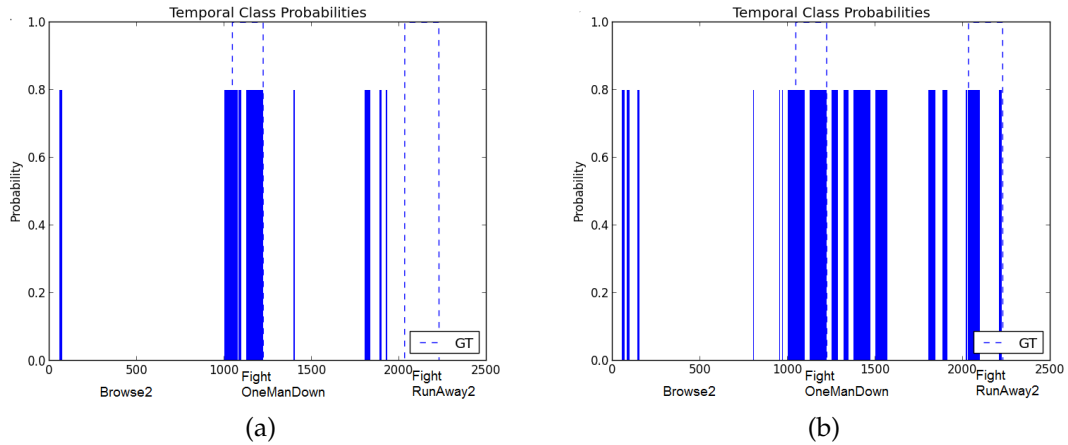


Figure 5.6: Unusual event detection results thresholded using the optimal F1 score for each classification method independently: a) incremental learning (AGG) b) batch learning (GMM)

Class	precision	recall	f1-score
0	0.90	0.92	0.91
1	0.53	0.47	0.50
avg	0.84	0.85	0.85

Table 5.6: Online classification (AGG) results thresholded with optimal F1-score; 0 - usual class; 1 - unusual class

Class	precision	recall	f1-score
0	0.94	0.72	0.82
1	0.34	0.74	0.46
avg	0.84	0.73	0.76

Table 5.7: Batch classification (GMM) results thresholded with optimal F1-score threshold; 0 - usual class; 1 - unusual class

findings are supported by the breakdown of the results in Table 5.6 and Table 5.7, where the detection precision of the frames belonging to the usual class (indicated by the first column of the class 0) is lower for the incremental learning approach. It indicates that the online approach missed a higher amount of segments of the unusual events. The higher precision of the unusual class shows (indicated by the

first column of the class 1) that a smaller amount of false events is detected by the online approach when compared to the batch learning method.

When looking only at the recall of the ground truth unusual events (Figure 5.7) it can be seen that although the first event is almost fully detected by the online agglomerative algorithm, the second fighting event is missed. The GMM approach marked more than half of the segments belonging to this event as unusual. The F1 score treats the detection rate of true events and the precision of

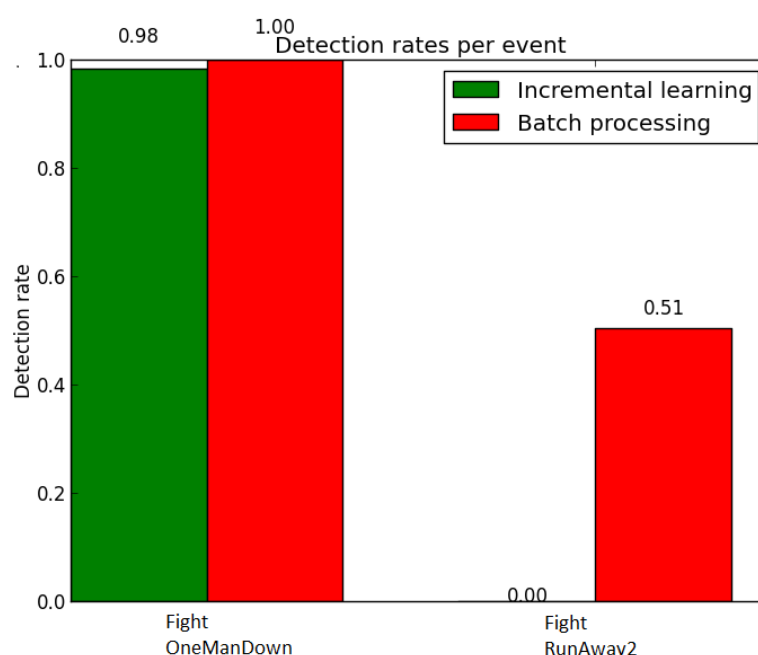


Figure 5.7: Recall of the fighting events

the results equally, thus, it adds more weight to the recall of the results. This yields more true events to be marked as unusual and reduces the precision as shown in the precision-recall curves (Figure 5.5b). Due to the *F1*-score requirements of the ground-truth annotations, the search for a better threshold is investigated in the next section.

## 5.6 Dynamic threshold

In order to acquire binary classification results in a continuous environment, constraints such as the lack of annotations and extended run-time of the system need to be considered. Whilst the F1-measure could be used to optimize the threshold, it requires data with annotations that are not always available in surveillance scenarios. Moreover, a predefined threshold might not be an ideal solution because of the dynamic nature of real-world surveillance scenarios. After an extended period of time, the static threshold will become obsolete. Thus, an adaptive threshold is a preferred approach in real-world surveillance applications. Three adaptive thresholding approaches are tested.

The first method defines the threshold by extracting the *mean*  $th_t = \mu(D)$  of the distribution of distances  $D$ . The distribution represents the distances between the model and the data instances collected until the time  $t$ . In this approach, the threshold is reevaluated for each data instance. If the distribution of distances is more or less flat, the method is expected to identify large numbers of events as unusual. Results for this approach can be seen in Table 5.8. The second

Class	precision	recall	f1-score
0	0.99	0.37	0.54
1	0.23	0.98	0.37
avg	0.87	0.46	0.51

Table 5.8: AGG classification results thresholded with *mean* threshold; 0 - usual class; 1 - unusual class

method defines the threshold by the mean of the distribution plus two standard deviations, *mean+sd*:  $th_t = \mu(D) + 2 * \sigma(D)$ , where  $\mu(D)$  is mean of the distances' distribution  $D$  and  $\sigma(D)$  is the standard deviation. This approach defines higher threshold values than the *mean* approach causing less false positives. Results for this approach can be seen in Table 5.9. The third approach is based on the expected fraction of unusual events in the dataset and defines the threshold by

Class	precision	recall	f1-score
0	0.91	0.60	0.72
1	0.24	0.67	0.36
avg	0.80	0.61	0.67

Table 5.9: AGG classification results thresholded with *mean+sd* threshold; 0 - usual class; 1 - unusual class

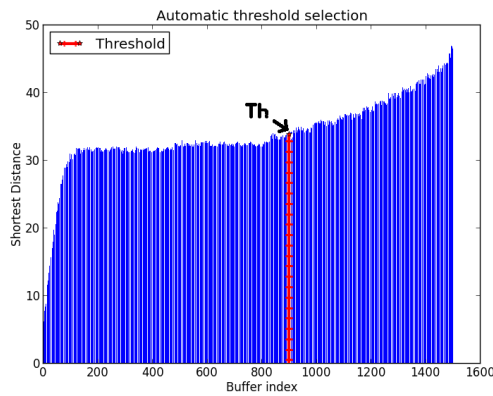
selected the high value from the distances' distribution:  $th_t = D((1 - \alpha) \cdot N)$ , where  $\alpha$  is a fraction of the data expected to be unusual and  $N$  is the number of distances in the distribution. In the test dataset, 2% of the data is known to be unusual leading to the value of  $\alpha$  to be 0.02. This approach is named *unusual fraction*, and the results can be seen in Table 5.10.

Class	precision	recall	f1-score
0	0.90	0.83	0.86
1	0.37	0.53	0.43
avg	0.82	0.78	0.79

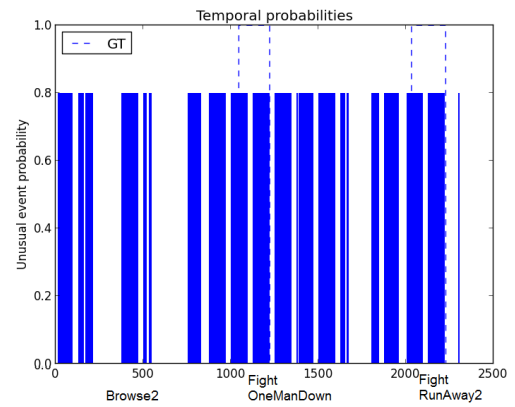
Table 5.10: AGG classification results thresholded with *mean* threshold; 0 - usual class; 1 - unusual class

### 5.6.1 Evaluation

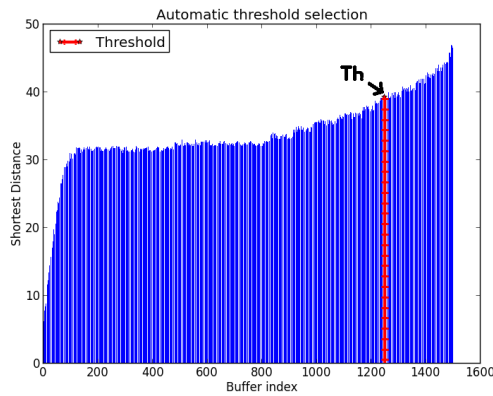
Figure 5.8 shows the distance distributions with the threshold values from the three methods marked with a red line (Figures 5.8a-5.8e), and the corresponding unusual event detection results (Figures 5.8b-5.8f). The threshold selection based on the mean of the distribution of distances (*mean*) can be seen in Figure 5.8a. The classification results based on this threshold can be seen in Figure 5.8b. This method yields a 98% recall rate, but only 23% precision for the unusual class as can be seen in Table 5.11. Despite high recall rates, this approach suffers from high rates of false alarms which is reflected in the precision value. The threshold selection from the distribution of distances can be seen in Figure 5.8c with the



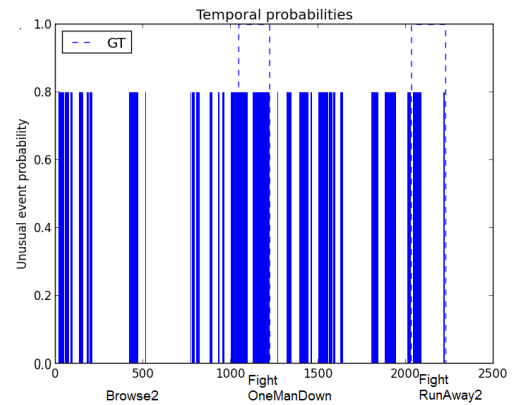
(a)



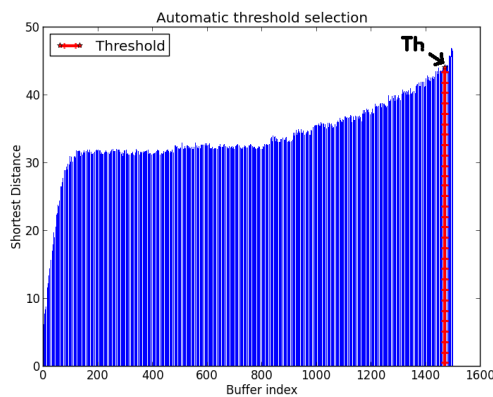
(b)



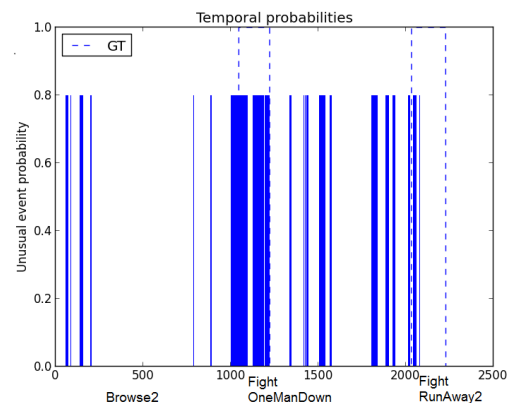
(c)



(d)



(e)



(f)

Figure 5.8: Top row: distances' distributions with the threshold value marked with the read line; bottom row: binary results. The results are evaluated using three methods: (a),(b) *mean*; (c),(d) *mean+sd*; (e),(f) *unusual fraction*;

Th method	Precision	Recall	F1-score
<i>mean</i>	0.23	0.98	0.37
<i>mean+sd</i>	0.24	0.67	0.36
<i>unusual fraction</i>	0.37	0.53	0.43

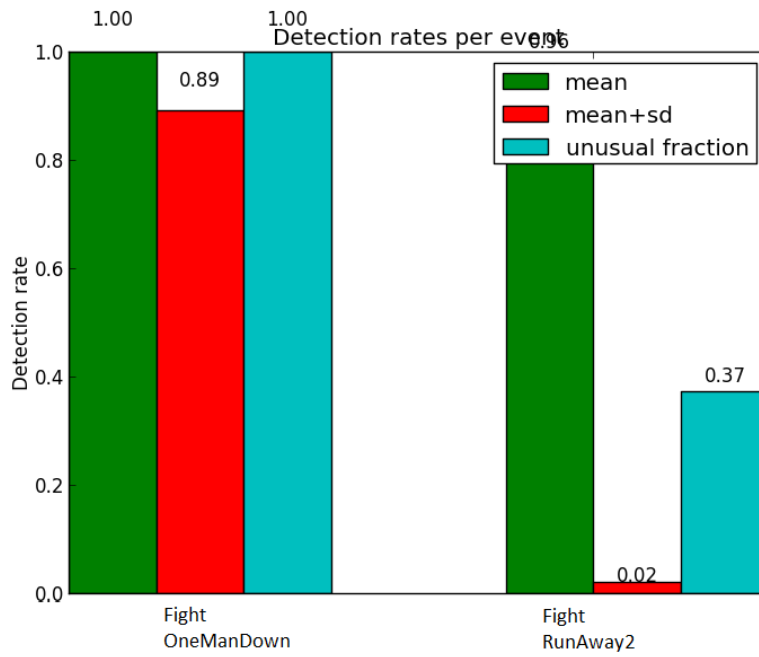
Table 5.11: Comparison of the unusual event class precision, recall and f1-score values of threshold selection methods

corresponding binary results in Figure 5.8d. When the *mean+sd* approach is used, the threshold is more sensitive therefore the precision is increased by 1%, but the recall of the unusual event frames is reduced to 67% as can be seen in Table 5.11.

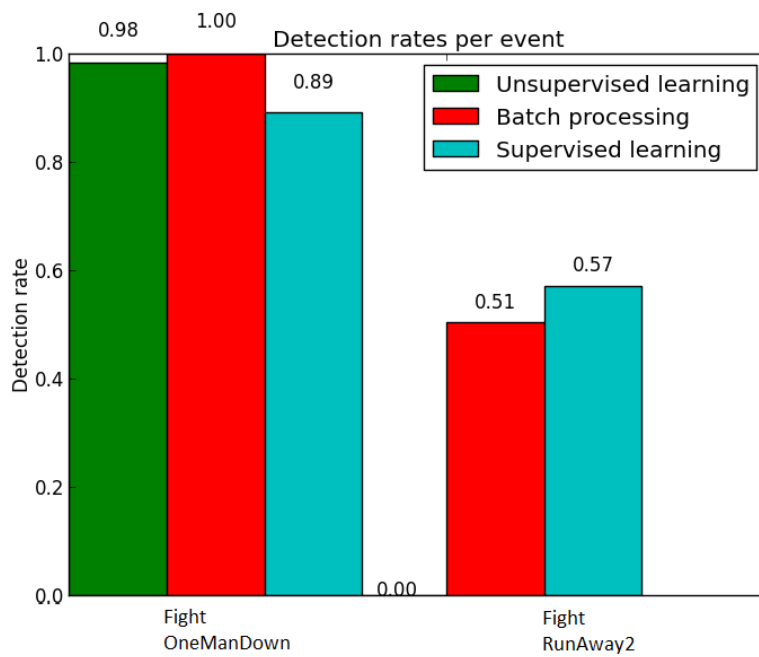
The table shows that f1-score, which combines precision and recall metrics, is 10% higher when *unusual fraction* methods is used when compared to *mean+sd*, and 28% higher when compared to *mean* method. The improved accuracy of this method outweighs the requirements for a priori information about the expected rate of unusual events in the captured environment. This information can be provided by experienced security personnel.

The fraction of the unusual events  $\alpha$  could be estimated by the experienced security personnel. The threshold selected from the distance distribution using *unusual fraction* approach which integrates  $\alpha$  in to estimation can be seen in Figures 5.8e. Figure 5.8f shows the corresponding binary unusual event detection results.

When compared with the other two methods, the precision of this method is significantly increased as can be seen in the Table 5.11. Recall values suffered from the increased precision. Therefore, to prove the advantage of this method the recall of the two individual unusual events that were present in the test dataset (each consisting of approximately 200 segments) can be seen in Figure 5.9a. The graph shows how well two unusual events are detected. The two events are video segments depicting two fighting events from the dataset (described in Section 4.3.1), namely *Fight OneManDown* and *Fight RunAway2*.



(a)



(b)

Figure 5.9: Recall of the unusual events (a) The automatic thresholding approaches applied to the online AGG results (b) F1 threshold selection for online unsupervised learning (online AGG); batch unsupervised learning (GMM) and supervised learning(SVM);



The graph shows, that the *mean* method identifies both unusual events with highest accuracy when compared to other thresholding methods. The precision of this method is lowest, therefore the overall result is not satisfactory (see Figure 5.8b). The *mean+sd* method identified the first unusual event with 89% accuracy, but nearly missed the second event. The *unusual fraction* method detected the first unusual event with 100% accuracy, and detected 37% of the second unusual event. Taking into account higher precision rate by 19% when compared to the *mean* approach, this method shows the most satisfactory result.

In addition to the comparison of the unusual event recall values obtained by the three automatic thresholding methods (Figure 5.9b), the recall values are plotted for all the evaluated classification methods with the F1-metric based threshold (Figure 5.9a). The recall rate for the unsupervised approach (online AGG) is improved with all three automatic threshold methods, when compared to the F1 threshold which was originally used. Moreover, it can be observed that when the F1 based threshold is applied, the second event is not detected at all using this method, while the proposed automatic thresholding techniques were able to identify at least a fraction of the second event. The *mean* approach showed the highest recall, but with the drawback of having a high false positive rate. The *unusual fraction* approach showed reasonable false positive rate (Figure 5.8f) and showed 37% improved when the unusual event recall is compared to the F1 based thresholding approach.

The *unusual fraction* thresholding approach yields the recall 9% lower than the supervised approach (SVM) with the F1 threshold and 14% lower recall than the batch unsupervised approach (GMM) with the F1 threshold. However, the precision value is only 1% lower than the SVM approach and 11% higher than the GMM approach. The results show that the precision of the results is improved by applying the *unusual fraction* thresholding approach.

## 5.7 Proposed Video Processing Flow

It can be concluded, that the threshold estimated from the data itself (based on the results of the three proposed methods) provides better unusual event detection results than the F1 optimized threshold which requires true labels of the data. The advantage is due to its ability to adapt to the characteristics of the data. The threshold estimated from the data also provides detection rates that are more comparable to the supervised approach (SVM), which in the previous sections was identified as a benchmark for unusual event detection. Based on the results from the previous experiments, the online agglomerative algorithm (online AGG) with dynamic threshold estimated using the *unusual fraction* approach has comparable capability to identify unusual events to the SVM approach. The ideal unusual event detection method needs to conform to the requirements of real-world surveillance applications identified at the beginning of this Chapter, namely (a) ability to learn without the labels of the training data (b) ability to process huge volumes of continuous data (c) ability to learn parameters of the algorithm from the data itself. The approach that takes into account the requirements is as follows:

- **Temporal segmentation:** Video segmentation using windowing technique with overlapping windows (15 frames length and 1 frame shift in these experiments)
- **Feature vectors:** Dense trajectory based local regions described using motion boundary histogram methods and quantized using bag of visual words approach
- **Training:** Online agglomerative clustering method
- **Classification:** Unusual fraction threshold estimated from the past data

The results acquired using this approach are further analysed in the next section that focuses on the analysis of the falsely identified data instances and provides a qualitative evaluation of the results to aid in better understanding of the results obtained.

## 5.8 Qualitative evaluation

In this section, a qualitative evaluation of the binary results is presented, acquired using the unusual event detection approach summarized at the end of the previous section. A temporal averaging filter of width 10 is applied to join the frames that are consequently detected as unusual and to remove the frames that are detected in isolation. Figure 5.10 shows the resulting 12 video segments, where events

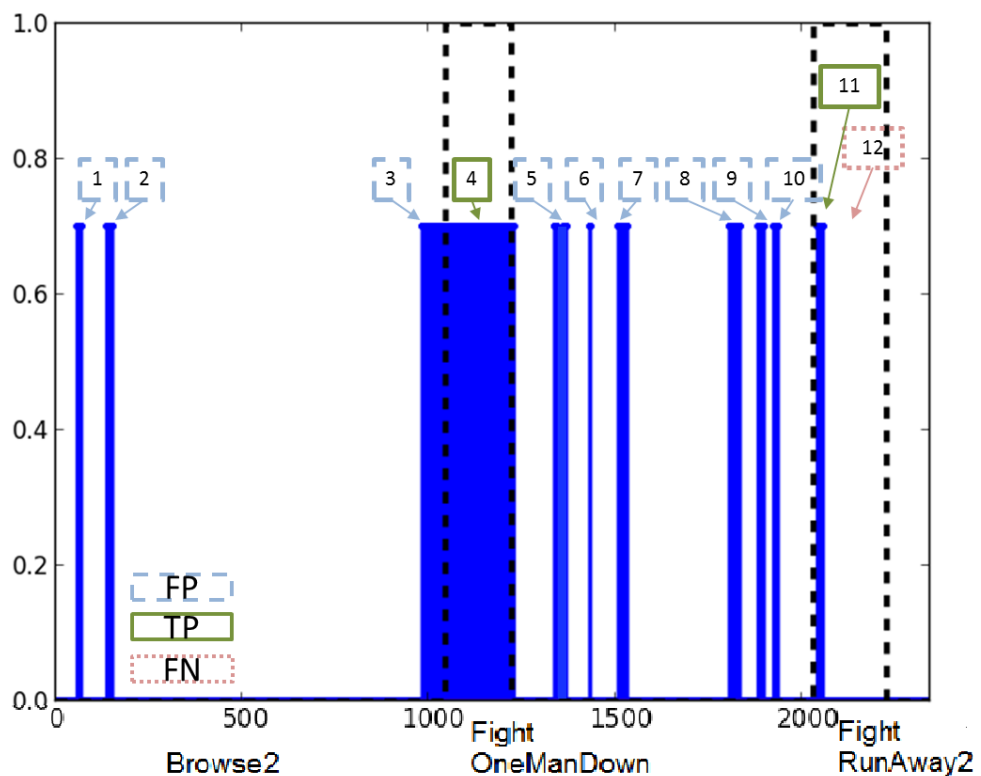


Figure 5.10: Event detection results for qualitative evaluation

enumerated with the numbers 4 and 11 correspond to the correctly identified

unusual instances (the true positive (TP)). The incorrectly identified instances (the false positive (FP)) correspond to events identified by numbers 1 to 10 (except 4). The missed part of the ground-truth fighting event, (the false negative (FN)) is identified by number 12. In Figures 5.16-5.15 representative images are shown for each identified instance. The samples of the correctly identified usual event frames (the true negative (TN)) are shown in Figures 5.13 and 5.14.

**True Positives (TP)** are the video segments that were correctly identified as unusual. Figures 5.11 and 5.12 show sequences of frames that represent the fighting event that are correctly identified as unusual events. The events are distinguishable by excessive hand movements, falling on the ground activity (Figure 5.11) and the fighters approaching each other at a fast pace (Figure 5.12).



Figure 5.11: TP - (4) - A fight with one man falling down on the floor (7.6s)



Figure 5.12: TP - (11) Two fighters approaching each other (1.3s)

**True Negatives (TN)** are the video segments that were correctly identified as usual. The frames that belong to the usual events correspond to empty scenes and people walking around. A few samples of the correctly identified usual event of walking people can be seen in Figures 5.13 and 5.14.

**False Negatives (FN)** are the video segments that belong to unusual events but were missed by the algorithm. Figure 5.15 shows a sequence of frames that belong



Figure 5.13: TN A person browsing (20s)



Figure 5.14: TN - A person walking out of the scene (3s)

to the fighting event according to the ground-truth but that was missed by the proposed algorithm. The missed sequence is part of the fighting scene, but has an inadequate amount of motion in the fighting action (event 12 from the figure 5.10).



Figure 5.15: FN - (12) A person attacking another person (7.2s)

**False Positives (FP)** are the video segments that are falsely identified as unusual. Figures 5.16 - 5.24 show the sequences of the frames that were falsely detected as unusual events.



Figure 5.16: FP - (1) coming towards the middle, turning towards the camera and picking up both hands (active gestures) (1.1s)



Figure 5.17: FP - (2) facing the camera and lowering hands towards sides (after raising them) (1.4s)



Figure 5.18: FP - (3) - Both fighters enter the scene from with a fast pace (3s)



Figure 5.19: FP - (5) A person sitting up after the fight (1s)



Figure 5.20: FP - (6) A person standing up after the fight (1s)



Figure 5.21: FP - (7) A person walking after standing for a while (0.8s)

Events 1 and 2 from Figure 5.10 depicted waving to the camera as can be seen in Figures 5.16 and 5.17. A scene after the fight where a person is standing up from the floor (event 5 and 6 from Figure 5.10) can be seen in Figures 5.19 and 5.20. Event 7 from Figure 5.10 is an incorrectly classified event that represents a person





Figure 5.22: FP - (8) A person walking in a fast pace (1.7s)



Figure 5.23: FP - (9) A person waving a paper towards the camera (1.3s)



Figure 5.24: FP - (10) Walking away while waving a white paper in his hand (1.2s)

walking as can be seen in Figure 5.21. A person walking at a fast pace, event 8 in Figure 5.10, can be seen in Figure 5.22. A person waving a piece of paper and walking away, events 9 and 10 in Figure 5.10 can be seen in Figures 5.23 and 5.24. Visual inspection of the frames falsely detected as unusual shows that most of the false detections are in fact outliers in the dataset. For example, a person standing from the middle of the floor, a person waving with his hands or a piece of paper. Only the event 7 shown in Figure 5.21 can be marked as true FP events, as it has a person walking activity which is a usual scene in the dataset and is not of interest.

## 5.9 Conclusions

The experiments conducted in this chapter were designed to evaluate the trade-offs in accuracy that need to be taken into consideration when the constraints of

the real-world surveillance applications are considered when implementing an unusual event detection system.

The first constraint identified for real-world applications is the lack of annotated training data. To evaluate how the accuracy differs when the training data annotations are provided and when the algorithms are trained without the data annotations, supervised (SVM) and unsupervised (GMM) training methods are applied. The results showed comparable accuracy of unusual event detection, but the unsupervised method suffered from high false detection rate. The false detections reduced the overall accuracy of the results, when measured by the area under curve (AUC-ROC) measure, by 8%.

Two unsupervised algorithms are compared (GMM and AGG) to investigate the second trade-offs of real-world unusual event detection. One of the algorithms (GMM) learns the model through an iterative optimization process called expectation maximization (EM) in batch mode. This approach requires an entire training dataset at each optimization step. The second algorithm (AGG) learns the model through incremental updates. This method gradually learns the model making it available for classification right after its initialization. The results showed that while the AGG method improved the precision of the results by 19% when compared to the GMM approach, it missed one important event which would not be acceptable in a real-world environment. Further analysis of the results revealed that the missed event could have been detected if the threshold value for unusual events was lowered.

Further experiments were carried out to find the optimal threshold value. Three threshold methods that are estimated directly from the data were investigated in combination with the AGG classification approach. The binary results were compared with the results acquired by a threshold method that requires labeled data to do the optimization. The results showed that the threshold estimated from the data improved the recall of the unusual events, while reducing the



precision of the unusual event detection by 10%. To investigate the reasons that caused lower precision values, a qualitative evaluation of the results was carried out. The conclusion was made that the false detections affected the precision of the results. The falsely detected events were indeed deviations from the normal scenarios in the dataset, but were not part of the fighting events that made up the ground-truth unusual class.

It can thus be concluded that the unsupervised approaches are valuable in detecting outlier events that are not known in advance. In surveillance applications, it is reasonable to assume that the user input becomes available at some stage of the processing due to the typical setup of the visual surveillance. Therefore, the annotations can become available as well, and the supervised method can be utilized to identify events that have already happened. A combination of the supervised and unsupervised methods, when both methods are implemented in an online mode, is expected to improve the overall accuracy of the unusual event detection system. The combined system would be able to identify unusual events that are not known in advance using the unsupervised approaches, and to identify the unusual events that are known to happen in the monitored environment. The next chapter is dedicated to investigating the fusion of the classifiers and the requirements that need to be taken into account when integrating a combination of the supervised and unsupervised classification methods into real-world surveillance systems.

## **5.10 Summary**

The experiments discussed in this chapter investigated online and batch based training methods, as well as supervised and unsupervised ones. The accuracy differences based on the AUC measure of ROC curves showed that classification accuracy of a method based on an online training is 1% lower when compared

to the batch training approach. The difference in classification accuracy between supervised and unsupervised approaches is found to be 8%, where the supervised approach is superior. Therefore, there is a 9% accuracy trade-off to be made when implementing unusual event detection algorithms with the constraints imposed by real-world applications. Using a threshold that is estimated from the data itself showed better results than the F1 threshold estimation method that requires true labels of the data. The accuracy of the results is showed on a per-segment basis due to the small number of available interesting events. When the accuracy is evaluated on this granularity, it could be compared to the object detection evaluation per-pixels rather than per-object lower accuracy numbers are observed in this situation, but more detailed information about the actual performance of the algorithm can be seen. The real world data is expected to show similar accuracy levels. Moreover, qualitative analysis of the online unsupervised approach showed that the accuracy measures mostly reflect the ability to detect the predefined ground-truth (fighting) events. Unusual events that were identified as false detections in the experiments were somewhat different from the standard scenes in the dataset. In real-world unusual event detection applications, those events may be of interest to the user. Due to the ability of the supervised classification method to identify known events, and the ability of the unsupervised method to identify interesting, but unknown, events, the following chapter reports on experiments investigating a hybrid classification approach.

# Chapter 6

## Combining classifiers

### 6.1 Overview

In previous chapters, a number of machine learning techniques for unusual event detection were presented. The trade-offs that need to be made for them to be suitable for real-world applications were investigated. The conclusions were that both supervised and unsupervised methods have complementary properties that could be utilized via a combination of these methods. The experiments showed that unusual event detection accuracy using supervised classification approach was 9% higher than the accuracy of the online unsupervised method. Nevertheless, through the qualitative evaluation of the results it was found that the accuracy of the unsupervised methods was affected by the falsely detected unusual events that were salient in the dataset and could be of interest in real-world scenarios. This chapter explores the potential improvement of the overall accuracy of the system. First, the stability of the unsupervised classification approach is analyzed using a bootstrap technique, and a bootstrap aggregation technique is investigated for stabilization of the method. Stability analysis using a bootstrapping techniques is a novel approach to the evaluation of the algorithm. Then the combination of both methods is explored in order to improve the results

of the overall system. Combination of supervised and unsupervised visual event classification algorithms is novel in research on visual surveillance.

## 6.2 Motivation

There are two factors in favor of applying the online unsupervised classification approach. One factor is the ability to make the predictions without having the labeled training examples of the events of interest. The second factor is the ability to continue learning for an extended period of time without increasing the computational complexity or the demand for data storage. The first factor is satisfied by the classification methods that are unsupervised, and the second factor is satisfied by the online learning methods. The combination of the two creates a solution applicable in real-world applications. One more important factor of a learning algorithm is its sensitivity to small changes to its input data. The affect these small changes have on the results can be explored by evaluating stability. Stability evaluation is an important part of algorithm development as it shows how well the learning algorithm can generalize.

The experiments conducted so far reported on the accuracy of the online unsupervised classification method when compared to the state-of-the-art supervised classification approach. The analysis of the results showed that higher accuracy is achieved with the supervised approach when detecting unusual events that are known in advance while the unsupervised approach is superior in identifying unusual events that are not known in advance. Based on this, a combination of the methods is proposed. The combination is expected to be able to identify unknown unusual events, and the unusual events that are already known.

Due to the lack of the information about the usual and unusual events in the captured environment, initially the predictions are made solely by the online unsupervised method. With the progress of time, the user can add labels to

certain usual and unusual events. To optimize the use of user input, active learning techniques can be applied. Active learning algorithms focus on choosing the data points that would give the most valuable information to the learning algorithm. The user is then asked to label only a small subset of overall data points. A list of active learning techniques are summarized and compared by Settles (2010).

Once labeled examples for both classes become available, the supervised classification method can be trained to identify known events. At this stage, the knowledge of the supervised method can be integrated into the system.

First, the stability of the online unsupervised methods is investigated in order to test the proposed idea. Next, incremental learning of supervised and unsupervised methods is considered followed by a proposal of approaches to combine the two methods. Experiments are conducted to validate the proposed ideas on the benchmark surveillance video dataset.

### **6.3 Algorithm Stability**

Most of the unsupervised activity detection methods focus on the accuracy of the results rather than the stability of the algorithm (Zhong et al., 2004; Sillito and Fisher, 2008; Breitenstein et al., 2009; Loy et al., 2010). In addition to the overall accuracy of the classification algorithms, analysis of the stability can show how variations of the input can influence the output of the system. Bousquet and Elisseeff (2002) showed that the stability of the algorithm in certain cases ensures good generalization. The stability of the classification algorithm can be evaluated by investigating the variance of the results with the perturbed training data. The affect of the changes to the results of the classification reveals the stability of the algorithm. Luxburg (2010) lists five different methods to perturb the dataset for stability investigation:

1. Draw a random subsample of the original dataset without replacement.
2. Add random noise to the original data points.
3. Reduce the dimensionality of the data using random projections into low-dimensional space.
4. Sample data from the model, if the generative model is known for the dataset.
5. Draw a random sample of the original data with replacement.

In all cases, precautions must be taken when choosing the amount of the additive noise or the size of the training datasets. For example, if too much noise is added or the subsample of the dataset is too small, the structure that the algorithm is trying to find might be destroyed. If the changes in the dataset are too small, then the observed stability would be trivial as the algorithm will always obtain the same result. The fifth approach from the list is known in the literature as a bootstrap and is advantageous when compared to other methods. It does not require setting of the size of the subsample for the perturbed dataset.

The bootstrap method involves creating multiple training sets  $X_b$  from the original training dataset  $X = x_1, \dots, x_N$ , where  $N$  is the number of training data points. Each set is created by randomly drawing  $N$  samples from  $X$  with replacement. This means that the same training item can appear more than once, whereas other items may be left absent from  $X_b$ . Each set has the same size as the original training dataset. Each bootstrap dataset  $X_b$  is then used to train a separate copy of the predictive model  $M_b$ . The stability of the method is assessed through the variation of the predictions between the bootstrap models  $M_b$ .

In addition to the evaluation of the classification algorithm's stability, the bootstrap models can be combined to improve the stability of the algorithm (Parker, 2010). A final prediction can be given by averaging the predictions of  $M$

models (Bishop, 2006):

$$y_{com}(x) = \frac{1}{M} \sum_{m=1}^M y_m(x) \quad (6.1)$$

where  $y_{com}$  is the final prediction and  $y_m(x)$  is the prediction of a bootstrap model  $m$  for instance  $x$ .

### 6.3.1 Experimental setup

The SVM classification approach is excluded from the stability analysis as its stability is well analyzed in the literature (Bousquet and Elisseeff, 2002). For the online agglomerative clustering method (AGG), the variation between classification results produced by the bootstrap models is assessed. Aggregation of the bootstrap models is evaluated in order to improve the stability of the algorithm. The AGG model is set to have 100 clusters. The reasoning is that as long as the number of clusters is reasonably big, the inherent clusters of the sample space is covered with some redundancy without significantly affecting the model.

Video events are represented by the temporal segments extracted using a windowing technique with a window 0.6s long and 0.04s shift. Each segment is represented by the occurrences of the spatio-temporal regions in the codebook consisting of 1000 instances, based on the previous experiments. Each spatio-temporal region in a temporal segment is represented by the motion boundary histogram (MBH). MBH is extracted from multiple scale regions around the trajectories of local points (for more details about the descriptors see Chapter 4). Starting with a single bootstrap model, aggregation of up to 50 bootstrap models is tested. Each aggregation is repeated 10 times to evaluate the variance of the final results. The CAVIAR<sup>1</sup> dataset is used for the experiments using the same setup as in the Chapter 5. The models are trained using 80% of the overall data.

---

<sup>1</sup>Data from EC Funded CAVIAR project/IST 2001 37540, found at URL: <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>

The rest of the data is used for testing. The unusual class data is divided into training and testing sets so that 50% of it goes to the training data (comprising 2% of the data), and the rest of it is left for testing.

### 6.3.2 Evaluation

Variance of the online agglomerative clustering algorithms is assessed using an area under curve (AUC) measures for both receiver operator curves (ROC) and precision-recall curves (PR). Figure 6.1 shows the results, where Figure 6.1a shows the AUC-ROC statistics and Figure 6.1b shows the AUC-PR statistics. Both

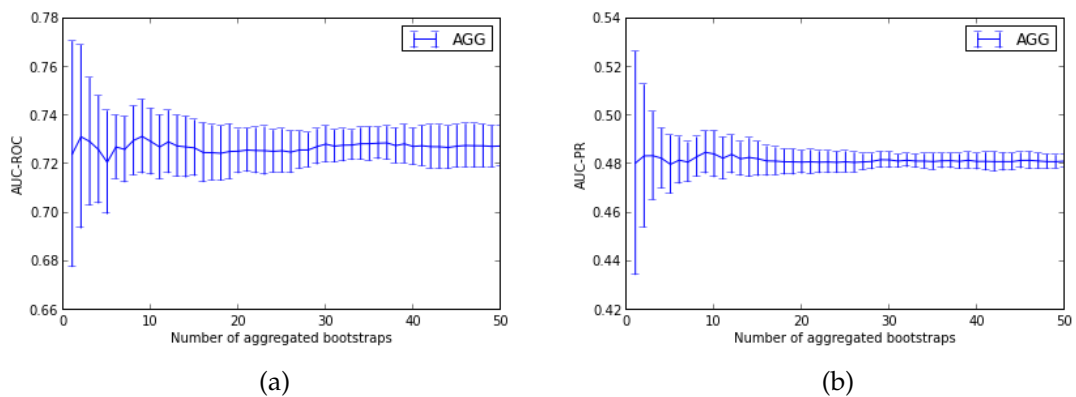


Figure 6.1: Incremental bootstrap aggregation from 1 to 50; variance over 10 random experiments

figures show the variance of the bootstrap aggregation results starting from single bootstrap, up to 50. Each increment in the bootstrap aggregation is repeated 10 times and corresponding variances are shown using the vertical error-bars. The stability of the AGG method is shown in the first error-bar of the both figures, where no aggregation is applied. Both AUC-ROC and AUC-PR metrics show 9% variance, which is acceptable taking into account the variety of events used to train the algorithms (see Table 6.1). The aggregation of results from multiple bootstrap models reduces the variance to 0.5% and 1.7% when measured by AUC-PR and AUC-ROC metrics respectively. The average accuracy of the results does



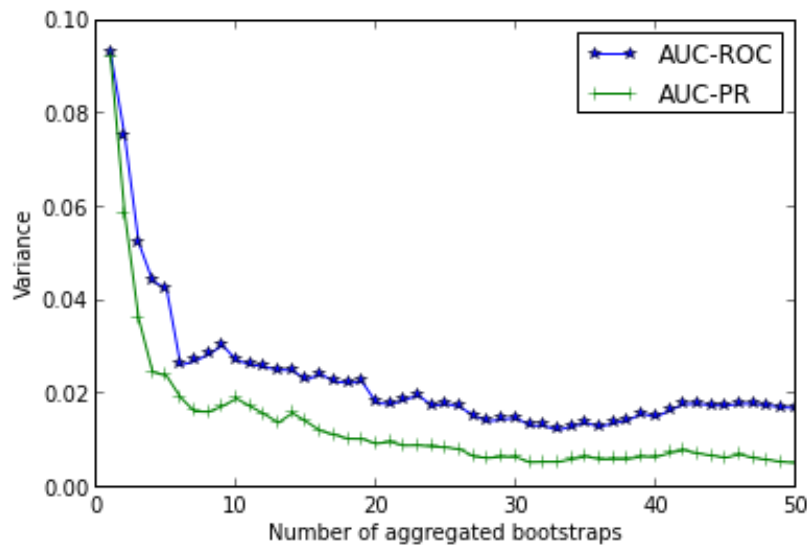


Figure 6.2: Variance of the event detection results with a model combining an increasing number of aggregated bootstraps

not change with the aggregation of more bootstrap models, and it stays at 72.8% for AUC-ROC and 48.1% for AUC-PR.

When improving stability by combining multiple models acquired from the same training dataset using bootstrapping, it is important to note that each bootstrap added to the decision making increases the complexity of the model. Therefore, it is important to minimize the number of bootstraps used. Figure 6.2 shows the variance trends with an increasing number of bootstraps starting from a single bootstrap, and finishing with the variance of the results from the model that aggregates 50 bootstrap models. It can be observed from both AUC-ROC and AUC-PR statistics that the variance steeply decreases for both metrics when up to six bootstraps are aggregated. Results from the aggregation of six bootstrap models reduces the variance by 72% from 9.3% to 2.6% for AUC-ROC, and by 80% from 9.2% to 1.9% for AUC-PR. Increasing the number of bootstraps up to 50 reduces the variance by further 10% and 5% AUC-ROC and AUC-PR metrics. A further reduction of the variance is not significant compared to the increased complexity when 50 models are created.

It can be concluded that combining bootstrap classifiers does not increase the overall accuracy of the results, but increases the stability of the model. Combination of the models that capture different aspects of the data is proposed to improve the accuracy of the results, where diverse models can bring complementary information for the decision making.

## 6.4 Incremental learning

Incremental learning is a machine learning paradigm where the learning process takes place whenever new examples emerge to adjust what has been learned according to the new examples (Ade and Deshmukh, 2013). There are three main advantages of using incremental learning in real-world surveillance applications: 1) it works well in limited memory and processing power scenarios 2) it can deal with sequential flow of information 3) it can adapt to the changes of the data after the learning phase is over.

There is a need to scale up learning algorithms to handle more data as the databases used for modeling the data increase in size. Incremental learning is one of the solutions to the scalability problem, where data is processed in parts, and the results combined so as to use less memory (Syed et al., 1999a). Incremental learning is also used in applications where all the data is not available at once, and it is streamed one sample at a time or in chunks of data at a time. Syed et al. (1999b) first proposed incremental learning of the SVM. Instead of retraining the model at every increment with all the data, the approach only retrains the model using only the support vectors and the new data. Experimental results showed that compared to the standard SVM approach, 0.5% loss of accuracy is introduced. This method was applied by Lu et al. (2014) for human recognition task. The additional condition was added to preserve the Karush-Kuhn-Tucker (KKT) condition (Pontil and Verri, 1998) on the reused training data. Incremental learning is also

a frequent choice for object tracking algorithms to adapt gradually to the changing representation of objects. Ross et al. (2007) applied an incremental principal component analysis (PCA) algorithm for the representation of the tracked objects, and showed that it is faster and more robust to model object appearance via incremental rather than batch PCA approach. Mixture of probabilistic PCA (MPPCA) was used by Kim and Grauman (2009) to identify unusual activities, where the initial MPPCA model was learned from a small sample of annotated data, and the model was updated incrementally with each new data sample. Xiang and Gong (2008) constructed an incremental HMM model to represent normal visual behaviour at the entrance to the restricted area. They initialized the model using a small bootstrap dataset and continued learning incrementally whenever a new behaviour pattern was captured. The model structure was adapted incrementally to accommodate changes in the definition of normality/abnormality when the visual context changes. Similarly, Ouivirach et al. (2013) applied incremental maximum likelihood (IML) algorithm to HMM model that requires updates only of the sufficient statistics as new events occur. Breitenstein et al. (2009) applied incremental clustering algorithm to continuously train the model of the scenes captured by a web-cam in the Time Square, New York. In their approach, with every new data sample the closest clusters are merged and only the statistics of the clusters are retained. It is important for real-world surveillance applications to investigate the properties of the algorithms when the amount of training data is increasing. The following experiments are conducted to investigate the continuous learning effect on supervised and unsupervised classification approaches.

#### **6.4.1 Experimental setup**

The experiments evaluate how the supervised and unsupervised methods respond to increasing amount of training data. The two algorithms that were investigated

Clip ID	Title of The Video Clip	Duration
1	Browse WhileWaiting1	31.1s
2	Browse WhileWaiting2	1min 15.2s
3	Fight Chase	5.32s
4	Fight OneManDown	7.8s
5	Fight RunAway1	8.24s
6	Fight RunAway2	7.84s
7	Browse1	41.8s
8	Browse2	35s
9	Browse3	54.6s
10	Browse4	45s
11	LeftBag AtChair	21.5s
12	LeftBag	57s
13	LeftBag PickedUp	53.6s
14	LeftBox	34s
15	Meet Crowd	19.1s
16	Meet Split 3rdGuy	36.4s
17	Meet WalkSplit	24.4s
18	Meet WalkTogether1	27.7s
19	Meet WalkTogether2	32.5s
20	Rest FallOnFloor	39.7s
21	Rest InChair	39.7s
22	Rest SlumpOnFloor	35.9s
23	Walk2	41.6s
24	Walk3	54.6s

Table 6.1: List of events used for training

in the previous experiments are used. The SVM classification algorithm with a Radial Basis Function (RBF) as a kernel with parameters  $C$  and  $\gamma$  set to 1 and  $1/d$  respectively ( $d$  is the number of features). The online agglomerative clustering method (AGG) composed of 100 clusters. The parameters are the same as in the previous experiments. The representation of video events is the same as in the previous section. The CAVIAR dataset is utilized for the experiments as it is a good representation of surveillance scenarios. The training data is incremented using video clips of approximately 35 second length until all the data is observed. Each increment of data corresponds to the video-clips that are listed in Table 6.1. Testing is performed on the 80% of the overall data - the same for all the increment

stages. The experiments are randomly shuffled ten times for cross-validation.

## 6.4.2 Evaluation

AUC-ROC and AUC-PR based performance evaluation results can be seen in Figure 6.3a and Figure 6.3b respectively. In the graphs, the x-axis represents the

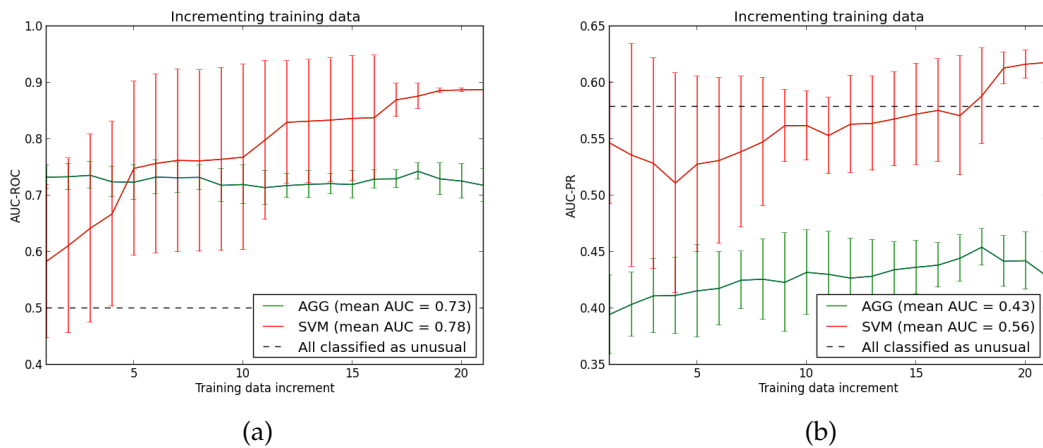


Figure 6.3: Incrementing training data experimental results a) AUC-ROC statistics b) AUC-PR statistics

increasing number of video-clips used for training. The accuracy measure is shown on the y-axis. The error bars at each increment show the performance variation between cross-validation experiments.

To be able to create a model using the SVM approach, examples of both usual and unusual events are required. When only a few training examples are available, it is very likely that all randomly chosen clips belong to the usual class. In this scenario, the supervised classification method is not able to create a model because it requires examples from both classes to be available for training. A solution to the situation where the model cannot be created is to set all the results to 1. In other words, in the case of the unavailability of the model, all the events are predicted to be unusual with maximum probabilities. It is a reasonable assumption for a real-world scenario, meaning that if the classifier is not able to make a prediction

on the data, this data has to be sent to the user for the inspection. The dashed lines in Figure 6.3a and Figure 6.3b show the accuracies achieved when the model is absent. The all-unusual results produce 0.5 accuracy with AUC-ROC metric, and 0.579 accuracy with AUC-PR metric. The accuracy is slightly higher with AUC-PR metric because the true-negatives<sup>2</sup> are discarded from the equation.

The accuracy of the AGG method is stable considering the AUC-ROC metric, but there is no significant improvement on the average accuracy with increasing amount of training data. The ROC metric is affected by the true-negatives due to the domination of the usual data in the dataset. Thus, the representation of the usual class remains stable with the introduction of more data. A 3% increase in average accuracy is observed with the AUC-PR metric. The PR metric does not take into account the true-negatives and indicates how well the unusual class instances are detected. The slowly increasing AUC-PR value in Figure 6.3b shows, that the accuracy of the unusual event detection algorithm increases when more data is available for training. On the other hand, a significant increase in accuracy can be seen with increasing amount of training data provided to the SVM modeling approach. Using the SVM approach, the AUC-ROC metric is improved by 20% if trained with all the available data, when compared to training with a single video clip. The result indicates that the *usual* class samples are better modelled with more training data. The AUC-PR metric is improved by 7% with the same data increments, showing that the data instances of the *unusual* class also benefits from more training samples. The lower rise of the AUC-PR metric is due to the low probability of the *unusual* data in the dataset. SVM method shows high dependency on the training data which can be seen by error-bars in Figure 6.3a. When more training data is used, the dependency on training data decreases. Therefore, to ensure stability of the supervised training algorithm, training samples have to be chosen carefully.

---

<sup>2</sup>The true-negatives are the *usual* class samples correctly identified as *usual*

The result shows that both methods benefit from the increasing amount of training data even if the data is dominated by the usual class events. The ability to identify *usual* events does not change when more training data is used with the unsupervised method (AGG), but the ability to identify *unusual* events slightly increases. On the other hand, the accuracy of the supervised method (SVM) significantly increases when more data is introduced. The higher increase of the supervised approach accuracy rather than the unsupervised approach is due to availability of the labels for the data. The next section investigates combination of the SVM and GMM methods in order to increase the overall performance of an unusual event detection system.

## 6.5 Combining classifiers for surveillance applications

A key goal of the combination of the models created for the same problem is to obtain a better composite global model, with more accurate and reliable estimates or decisions. The underlying assumption of fusing multiple classifiers is that each participating has a merit that deserves exploitation (Smits, 2002). The presumption in classifier selection is that each classifier is an expert in some local area of feature space and the classifiers are considered complementary. The novelty of the fusion approach proposed in this section is that it combines an unsupervised classification method with a supervised classification method. The fusion method is also adapted to an online data stream and only information from the past is used to learn weights of the fused components.

A general approach to combining classifiers is the fusion of the results via merging schemes, also called decision level fusion. Among the most popular approaches to decision level fusion are linear, logarithmic, and voting approach (Sinha et al., 2008). Linear fusion is popular due to its simplicity. Its output is a

weighted sum of the probabilities produced by each model:

$$P_{linear}(A) = \sum_{i=1}^K \alpha_i P_i(A) \quad (6.2)$$

where  $P_{linear}$  is the combined probability from a set of models;  $\alpha_i$  is the weight given to the  $i$ -th model;  $P_i(A)$  is the probability of the  $i$ -th model for the event  $A$ ;  $K$  is the number of combined models.

An alternative to linear fusion is logarithmic fusion. It is different from the linear approach, as within this formulation, the combined probability is zero if any model assigns a probability of zero. This approach consists of a weighted product of the model outputs:

$$P_{log} = \prod_{i=1}^K P_i(A)^{\alpha_i} \quad (6.3)$$

Another simple method for combining the results of multiple models is to use a voting procedure where each model must generate a decision instead of a score. Among the popular voting techniques are majority vote, maximum, minimum and median votes.

A less intuitive approach of combining the results is to use a machine learning method, such as SVM, and to treat the opinions of the experts as data themselves. Therefore, the opinions form an input to the classifier and a function for the final decision making is learnt.

When combining supervised and unsupervised classifiers, each method has expertise in different problems and brings complementary information to the system. When the supervised method is to be used in real-world surveillance applications, examples from both usual and unusual classes are required. Fortunately, most of the surveillance applications have an advantage of having a user at the other end. The surveillance application setting usually has a user in the



loop who is either dedicated to monitor video data in real-time, or someone who occasionally looks at the recorded data to see what has happened. Therefore, the knowledge about the events becomes available and can be made accessible to classification algorithms.

### 6.5.1 Experimental setup

Two decision level fusion methods are investigated in the further experiments: *MAX-fusion* and *SVM fusion*. These two methods are chosen based on the application targeted by the research. The main feature of unusual event detection in visual surveillance applications is that missed events are more serious than falsely detected ones. First method is *winner-takes-all* or *MAX-fusion* method. *MAX-fusion* method compares the predictions of the participating classifiers and the highest probability value for being unusual is set to be the final result. This method is suitable for an event detection task where missed unusual events are less tolerable than false detections. This method is based on the voting approach mentioned in the overview. The second method is an *SVM fusion* approach. It takes the output from the two classifiers together with the labels acquired from the user, and learns the decision function for the final prediction. This method is motivated by the unknown relationship between the two methods that can be learnt from the past predictions.

The parameters for the SVM and AGG methods are kept the same as in the previous section. The *SVM-fusion* method uses Radial Basis Function (RBF) as a kernel. Parameters  $C$  and  $\gamma$  are set to 1 and  $1/d$  respectively, where  $d$  is the number of features. The incremental learning is implemented as described in the previous section.

## 6.5.2 Evaluation

A comparison is made between the *MAX-fusion* approach and the *SVM-fusion* approach, as well as between the predictions acquired using the SVM and AGG models independently. The results from all the classifiers are shown in Figure 6.4. The results are displayed using the AUC-ROC metric in Figure 6.4a. AUC-PR metric results are showed in Figure 6.4b. The results are plotted on the training increments to simulate the real-world scenario. The results show that once the

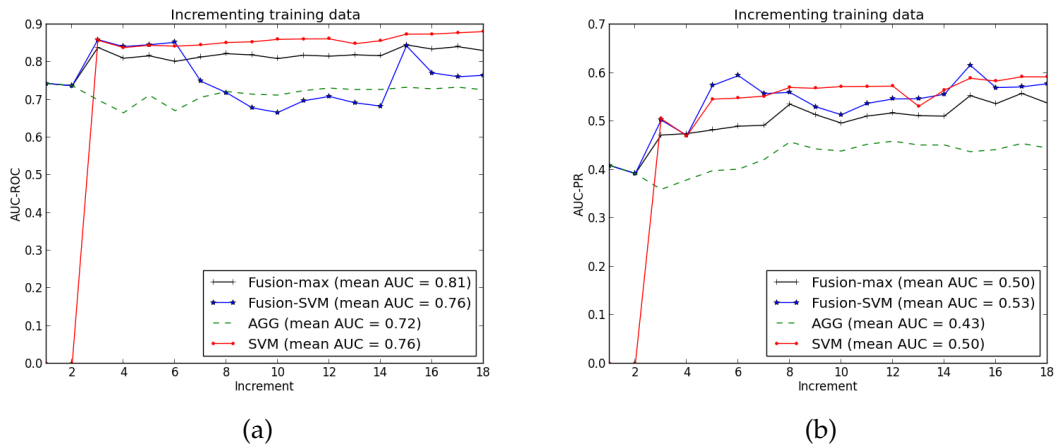


Figure 6.4: Combination of supervised and unsupervised classification methods with different weighting schemes

supervised classification approach (SVM) has obtained enough information to train both usual and unusual class models (number three on the X-axis of the Figure 6.4, at which point three video sequences were used for training the algorithms), the results of both fusion approaches show improvements to the unsupervised method (AGG) classification alone. The two fusion methods differ in their results. The overall accuracy of the results, when both usual and unusual classes are treated with the same importance, which is represented by the AUC-ROC metric (Figure 6.4a), the *MAX-fusion* results in higher accuracy to the machine learning based *SVM-fusion*. On the other hand, when the detection of unusual events is given more importance, represented by the AUC-PR metric (Figure 6.4b),

the *SVM-fusion* approach shows superior results. The unusual event detection recall (Figure 6.5) is shown to be 13% better for the *SVM-fusion* approach than the *MAX-fusion* approach. The result is only 4% lower than the benchmark result of

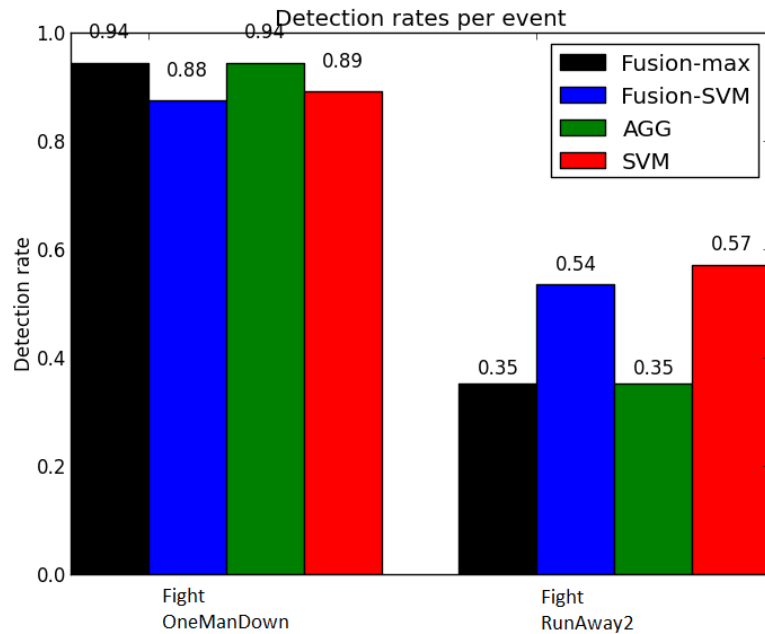


Figure 6.5: Recall of the unusual events

the SVM classification and as can be seen from the temporal plots of the results, it detected most of the unusual events that were identified as outliers after the qualitative evaluation in Section 5.8 (See Figure 6.6). The results show that all of the unusual events identified during qualitative evaluation have a sample of segments identified as unusual. In addition, both of the fighting events have a high recall rates.

## 6.6 Validation of algorithm invariance

The Ilids dataset (U.K. Home Office, 2011) is chosen to validate the invariance of the proposed algorithms to changes in the environmental conditions. The dataset comprises of CCTV video footage in real operating conditions with potential

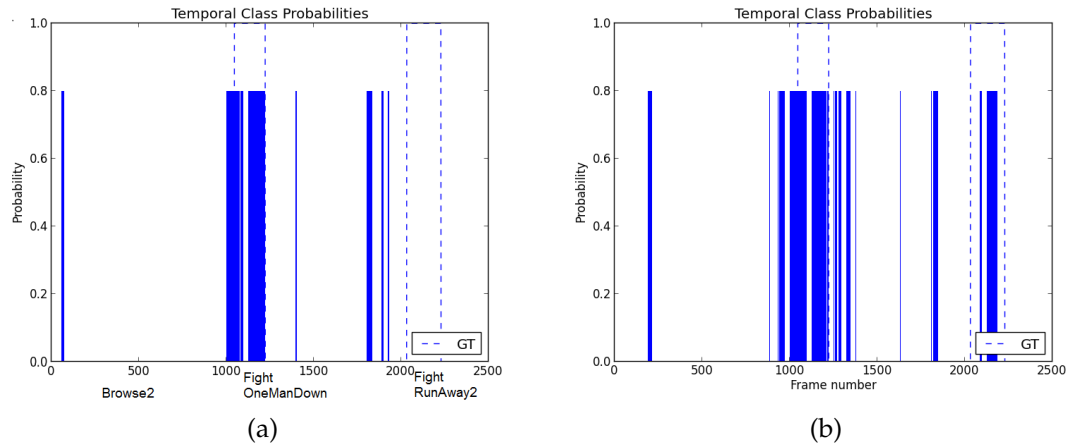


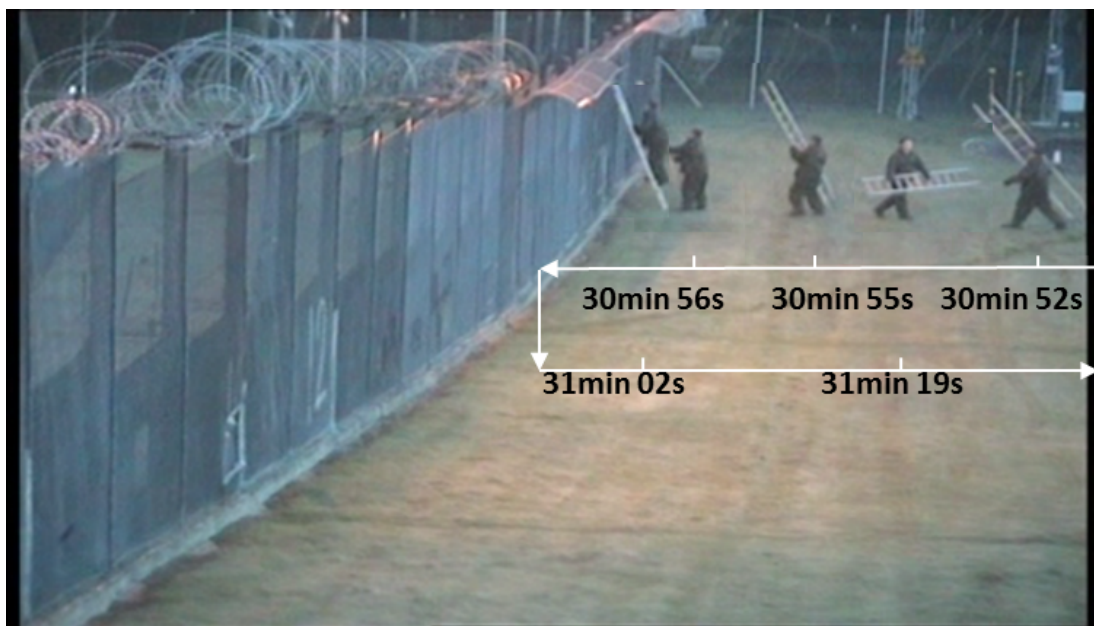
Figure 6.6: Unusual event detection results thresholded based on the optimized  $F - 1$  measure a) AGG b) SVM-fusion

threats captured. A video from sterile zone monitoring is chosen as a validation dataset because of its definite interpretation of the events. The duration of the events captured in the dataset span from 31sec to a 2min 29sec. The sample frames from the dataset can be seen in Figure 6.7. Two sample events are showed in the figure with the superimposed grade of time to allow for the perception of time-span. Event (a) in the Figure is captured at night time and lasts for one minute. It depicts a person escaping through the fence by slowly approaching, cutting a hole and escaping to another side through it. The second event (b) lasts for 31 seconds and depicts a person bringing a ladder, trying to climb over the fence and due to unsuccessful attempt turning back to where he came from.

The algorithm is validated with the subset of the Ilids sterile zone dataset of 37min 11sec long and consists of nine events depicting a person or people trying to cross the wall. Five minutes of data was used to train the model, and the rest of the data was used for testing. Results are shown in the Figure 6.8. The results confirm the applicability of the unusual event detection algorithm to the sterile zone environment based on Figure 6.8a. The dashed line shows the boundaries of the ground-truth unusual events. Because the descriptors used to describe visual events are based on motion, only the moving part of the unusual



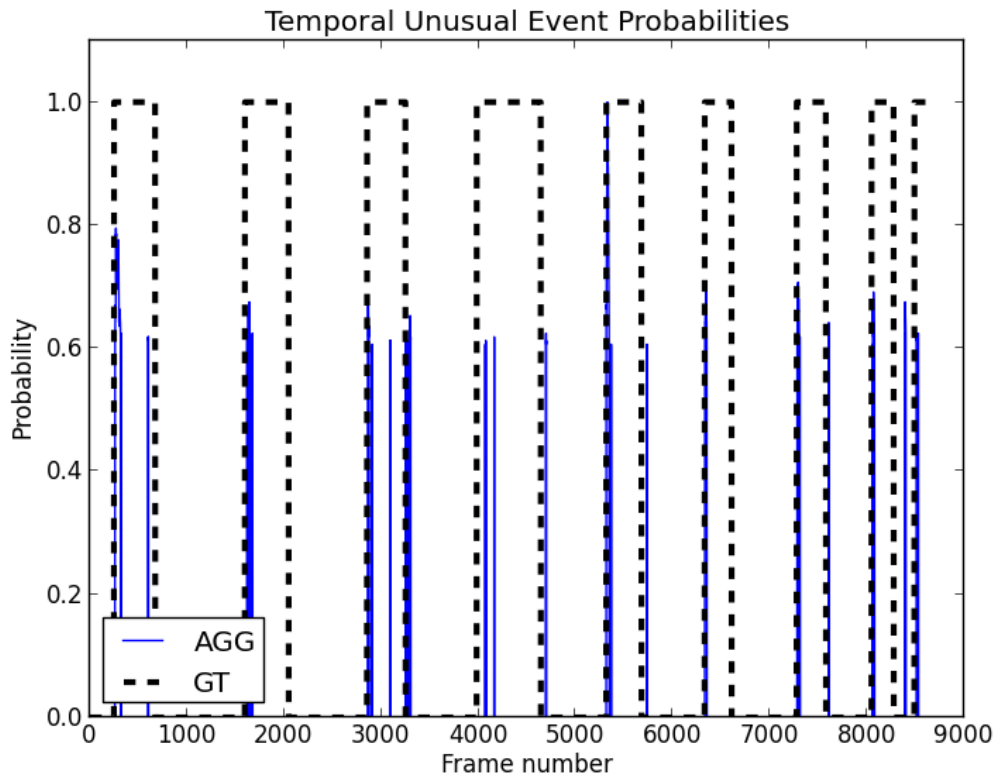
(a)



(b)

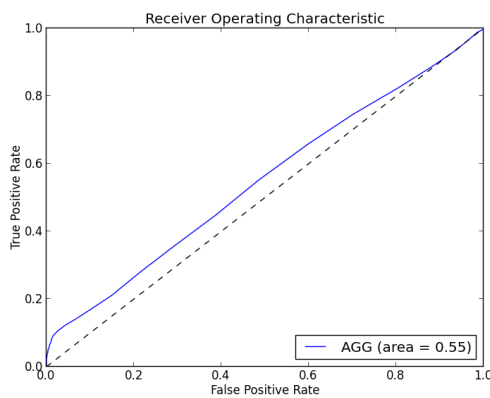
Figure 6.7: Sample events superimposed on single frames: taken from Ilids dataset sterile zone video data

event is detected. The figure shows that the beginnings of each unusual event are identified as unusual with high probabilities. The remainder of the events are static - depicting a person cutting the fence. Because the person is always facing the back to the camera, no motion is captured in this period of time. There

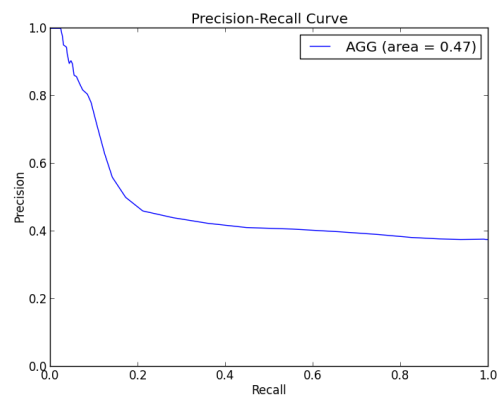


(a)

Figure 6.8: a) ROC curve b) Precision-Recall curve c) Thresholded temporal unusual event probabilities



(a)



(b)

is also some motion at the end of the events where the person is moving to the other side of the fence. Figure 6.9a depicts the ROC curve based on segment detection. Low accuracies showed in the ROC graph are affected by the unusual event parts that have no motion. The descriptors based on spatial information

need to be integrated to the preprocessing step to identify those areas as unusual. Precision-recall curve (Figure 6.9b) shows that only the segments belonging to the unusual events are classified with high probabilities to be unusual based on the high level of precision when the high threshold is applied.

The classification results acquired using Ilids dataset show that the algorithm is invariant to different scenarios as all the parameters are left unchanged when running the experiments.

## 6.7 Conclusions

In the previous chapter, it was concluded that the supervised classification approaches are optimized to detect known events, whereas the unsupervised approaches are valuable in detecting outlier events that are not known in advance. The combination of the two methods was suggested to improve the overall accuracy of the unusual event detection system.

First, stability evaluation was conducted to investigate the issue of the stability of the online unsupervised classification approach (AGG). The results showed that the variance of the AGG approach is 9%. Aggregation of bootstrap models is proposed to improve the stability, and the experiments showed that by aggregating six bootstrap models to get a final decision the variance is reduced to approximately 2%. Aggregation of more bootstrap models proved to increase the stability only slightly with a high computational overhead.

The incremental learning approach has to be implemented in real-world application scenarios due to its ability to keep complexity levels constant with increasing amounts of data. Experiments were carried out to investigate the effect that incremental training has on the supervised (SVM) and the unsupervised (AGG) training methods. The results showed that the AGG method has stable accuracy from the early stages of learning, while the SVM approach has increas-

ing accuracy with increasing amount of training data. The bigger effect on the supervised method (SVM) is due to the annotations that are provided together with the data at each increment. It was concluded, that the unsupervised method (AGG) requires a longer period of training for the model to improve, whereas the supervised method (SVM) can improve much faster. The results are consistent with the real-world implementation, where the unsupervised method would integrate all the incoming data, while the supervised method would integrate only the annotated data which results in less data available for this approach.

Two fusion approaches are implemented to test the idea of combining the supervised (SVM) and the unsupervised (AGG) classification approaches. The results show that both fusion approaches result in increased unusual event detection accuracy when compared to the unsupervised approach applied on its own. The *SVM-fusion* showed the benefits of both classification approaches where all of the identified unusual events, that were not part of the ground truth in the initial experiments were detected together with high recall rates for the unusual events identified as the ground-truth data.

## 6.8 Summary

This chapter investigated a user feedback approach which allows for a supervised method to be incorporated into decision making in the event-detection system when data annotations become available over time. This information acquired over time allows the real-world surveillance application to use a supervised classification approach which when combined with the unsupervised approach improves the overall performance of the event detection system. The experimental results show the increased accuracy when the two methods are combined when compared to the unsupervised only approach. Therefore, making the application more reliable when deployed in the real-world environment. The



following chapter discusses the overall results and gives further suggestions for investigations and possible improvements of the results.

# Chapter 7

## Conclusions and Future Work

### 7.1 Overview

In this thesis, the requirements for real-world surveillance applications are identified, and experiments carried out to evaluate the trade-offs that need to be taken to accommodate these requirements. The objectives of the thesis shape the research questions and thesis contributions reported throughout the thesis. This chapter overviews the findings of each chapter, revisits the research questions in retrospect, and examines the research questions in the light of the experimental results. Suggestions for future work are also proposed.

### 7.2 Thesis summary

In Chapter 1, the thesis is introduced by providing motivation, a brief overview of the research area, and hypotheses. Motivation for carrying out the research is based on three facts. Firstly, due to the increased security concerns, the number of video cameras deployed in public and private places such as airports, railway stations, shopping malls, business and private residences, is growing. Moreover, monotonous tasks can be automated to improve the efficiency of the human

operator that unavoidably suffers from limited concentration span. Finally, most of the research in the literature is focused on optimizing different parts of event detection task and does not focus on a combination of these parts into a real-world applicable system. The combination of these facts forms the motivation for the research conducted in this work. A brief overview of the research in the visual surveillance area identified the dependence of the event representation in surveillance applications on object detection and tracking techniques. The techniques are reliable only in constrained environments where the moving objects are clearly separable from the surrounding scenes and each other for the duration of the activity. Furthermore, most of the classification algorithms rely on the examples of each event to be identified, which are not always available in the real-world applications. The belief is that the detection of unusual events would benefit visual surveillance systems, but it requires investigation and evaluation of the methods that are applicable to real-world surveillance applications. Following the brief overview of the research in surveillance applications, the hypotheses are further expanded into the research questions that are investigated throughout the thesis. The research questions are:

1. Could unsupervised classification techniques be applied to unusual event detection and would it yield comparable results to the state-of-the-art supervised classification techniques?
2. Event representation is an essential part of the event classification task. Can space-time visual events be efficiently represented without relying on detection of the moving objects, accuracy of objects' shape, and their complete motion trajectory?
3. Could online training techniques be used as an alternative approach for training in applications where optimization techniques are not feasible?

4. Could a supervised classification method be integrated into the unsupervised system to benefit from the advantages of both techniques?

A literature review of research in visual event detection for surveillance applications is reported in Chapter 2. The review starts with an overview of the research in visual surveillance. Then the unusual event detection research is summarized. An overview of the methods for the event detection steps that include event representation, segmentation, and classification techniques, is also presented. Standard evaluation techniques are also reported in this chapter. The methods applicable to real-world surveillance applications are identified. The review revealed that detection and identification of the events is commonly treated as a supervised task. Supervised classification methods applied in the literature (dynamic Bayesian networks, hidden Markov models, support vector machines) rely on the training data of the known events to be provided to the algorithms. Unsupervised approaches are preferred in identification of unusual events, that are diverse and can be difficult to define for some environments. Unsupervised approaches are usually based on clustering methods (k-nearest neighbor, k-means, agglomerative), or mixture of models (Gaussian mixture model, mixture of Markov random fields, mixture of hidden Markov models). Abstraction of the events which consist of event segmentation and event representation depend on the complexity of the scene. If captured scenes depict mostly empty scenes with only few moving objects at a time, object detection and tracking techniques are applied. On the other hand, if the captured scenes depict a crowded environment with an object moving in an unordered manner, salient regions of interest, rather than objects, are extracted and described by the properties of the regions such as pixel motion, edge orientations, grey level intensities, etc. In the literature, supervised classification techniques combined with the object based event representation are commonly combined approaches. The review showed that a combination of the unsupervised classification techniques with the local region descriptors, the

combination that is applicable to real-world surveillance scenarios, is the area that requires to be further investigated.

In Chapter 3, experiments are conducted on a combination approaches of event abstraction and modeling techniques. The techniques are restricted to the ones applicable in the real-world surveillance scenarios defined in this thesis. An online clustering algorithm is applied to model the usual scenes in the captured environment. This approach fulfills two constraints of real-world applications. First, the clustering approach does not require labelled training data for model creation as it is relying on the assumption that the usual events dominate the surveillance video data. A feature of real-world surveillance applications that introduces the second constraint is the continuous streaming of data. The incoming data has to be continuously integrated into the model. The incremental model update procedure allows continuous training of the model. It is also preferable not to rely on object detection techniques. Therefore, a spatio-temporal event representation based on motion is implemented to represent overlapping temporal segments of video data. The unusual event detection routine is tested on the video data captured in a university corridor. The unusual events that were performed by volunteers over the period of one week were detected with 88% accuracy. Most falsely detected unusual events were found to be due to the environmental conditions such as weather changes. The unusual events that were missed by the algorithm had insufficient representation with the descriptors used.

In order to identify suitable descriptors for surveillance applications, the experiments in Chapter 4 were conducted. The list of suitable descriptors is first narrowed to the local region descriptors that do not depend on object detection, but have sufficient information to differentiate between events. Based on the results reported in the literature, the dense trajectory region-of-interest detection technique is adopted. The region-of-interest descriptors in the experiments were aggregated for each segment using a bag-of-visual-words approach, which is

commonly used in the literature yielding good results. The experiments were performed on surveillance-like benchmark data and a supervised classification approach. The results showed that the region-of-interest description based on the boundary motion information is superior to the descriptions based on trajectory shape, optical flow and edge orientation.

Online and unsupervised unusual event detection approaches are evaluated on the benchmark surveillance dataset in Chapter 5. The evaluation is carried out in two steps. First, an unsupervised event detection approach is compared to the state-of-the-art supervised event detection approach where both of the methods are based on the batch learning procedure. The second step evaluated the performance of two unsupervised event detection approaches, the online approach based on incremental learning, and the batch learning approach. The structure of the experiments allowed for the evaluation of the intermediate results between state-of-the-art supervised approach and the real-world applicable unsupervised approach. The results showed that the unsupervised event detection approach has unusual event detection accuracy 8% lower than the supervised approach. The online implementation of the unsupervised classification method has accuracy 1% lower than the batch processing based approach. The lower accuracy of both unsupervised methods is mainly caused by the falsely detected unusual events. Further investigation into the falsely detected events showed that most of the false positives were caused by actual deviations from the usual scenarios. The results are showed to be highly dependent on the thresholding approach applied to the unusual event probabilities, where an automatic threshold extracted from the data itself performed better than the threshold that optimized the *F1* measure.

The supervised and the online unsupervised event detection approaches were found to have complementary characteristics. Therefore, the combination of the two approaches is proposed in Chapter 6. First of all, to evaluate the stability of the online unsupervised method, the bootstrap technique was applied to the

training data and the aggregation of the bootstraps is proposed for improvement of the stability. The variance of the method was shown to be 9%, which might be an acceptable measure in real-world applications due to the per-segment rather than per-event evaluation and the average event contains approximately 350 overlapping segments. The variance is shown to decrease to the 2% when aggregating six bootstrap models, but with six times computational overhead. The experiments were also carried out to evaluate how the supervised and online unsupervised methods are affected by increasing amount of training data. The results showed that the supervised training method benefited from the increasing amount of training data quicker than the online unsupervised method. On the other hand, the online unsupervised method has a constant accuracy from the beginning of the training. Finally, two decision level fusion techniques were investigated, both showed improvements when detecting known unusual events comparing to the online unsupervised event detection method applied alone. The unusual events, that are not part of the ground-truth, but were identified as the deviations from the usual scenes during the qualitative evaluations, had at least part of them identified as unusual. The detection is sufficient to set a flag to the operator in the real-world environment.

### **7.3 Analysis and discussion of hypotheses**

In this thesis, a number of research questions in conjunction with central hypotheses are explored to investigate how computer vision and machine learning techniques can improve the effectiveness of real-world visual surveillance applications. In this section, the research questions are examined with respect to the experimental results obtained.

#### **Research question 1**

**Could unsupervised classification techniques be applied to unusual event detection and would it yield comparable results to the state-of-the-art supervised classification techniques?**

The first part of the question is explored in Chapter 3, where a complete unusual event detection system, that conforms to the requirements of real-world applications, is implemented. An agglomerative clustering (AGG) algorithm is implemented to learn the usual environment. It is an unsupervised classification algorithm with the capability to continuously integrate data into the model. The system is tested on a week of continuous video data captured in a university corridor outside the laboratories. In the dataset, a number of unusual behaviours were performed by volunteers, and the ability to detect those events is investigated using the proposed pipeline. Results showed that 14 out of 16 events were correctly identified as unusual. Analysis of the missed events and the falsely detected events suggested improvements in event representation where necessary. The second part of the question is investigated in Chapter 5, where a number of classification methods are investigated and compared. Two conclusions were drawn from the comparison of the unsupervised classification approach with the state-of-the-art supervised approach. The first conclusion is that the unsupervised approach has 8% lower event detection accuracy than the supervised approach. The lower accuracy is mostly caused by the false positives - events that were identified as unusual by the system but were not part of the ground-truth unusual events. The second conclusion is that the two approaches have complementary properties where unsupervised approach is appropriate for detecting unknown unusual events while the supervised approach is appropriate for detecting unusual events that are already known. This observation motivates further experiments on fusing the two approaches.

**Research question 2**



**Event representation is an essential part of the event classification task. Can space-time visual events be efficiently represented without relying on detection of the moving objects, accuracy of objects' shape, and their complete motion trajectory?**

In Chapter 4 the representation of visual events is explored. Following the discussion about what constitutes an event, three essential components of visual event abstraction techniques are identified. Firstly, the identification of local regions is required to filter out visual information that is not part of the event. The second part of the visual event abstraction is the description of individual local regions that are identified in the previous step. The last part is responsible for aggregation of the local region descriptors into a single descriptor that represents the event of interest. Experiments are conducted using five different local region descriptors to find the most suitable representation for events in real-world surveillance applications. The local region detection method and aggregation method (step one and three) are kept static through the experiments. Appropriate representation for visual events in surveillance applications is suggested as follows:

1. Dense trajectory interest point detector;
2. Motion Boundary Histogram descriptor for each local region;
3. Bag of visual words vector quantization;
4. Event representation using overlapping segments of 15 frames length and 1 frame shift;

### **Research question 3**

**Could online training techniques be used as an alternative approach for training in applications where optimization techniques are not feasible?**

Comparison between online and batch processing is investigated in Chapter 5. A literature review showed that the majority of the supervised classification methods can be adapted to work online. The trade-offs of integrating online training rather than batch training are: it is scalable for training with large amounts of data; predictions can be made from a very early stage of the training phase; ability to learn continuously which allows the model to be up-to-date for an extended period of time.

Experiments are conducted to compare two unsupervised modeling approaches, one based on the online processing and another based on batch processing. The experimental results showed that 1% of accuracy is sacrificed when using an online method instead of the batch method for unusual event detection. Similar results are shown in the literature for state-of-the-art supervised classification approaches. The final conclusion is that online training can be a good alternative to the batch training approaches when large datasets, or streaming data is used.

#### **Research question 4**

**Could a supervised classification method be integrated into the unsupervised system to benefit from the advantages of both techniques?**

The final research question is addressed in Chapter 6. Further improvements to the unsupervised unusual event detection are proposed by fusing the classification results of the supervised and unsupervised approaches. Integration of the unsupervised approach into the decision making is only possible when the events of interest are known. Information about the events can be made available after the user confirms the unusual events detected by the unsupervised approach. User feedback also can be integrated via active learning techniques where the samples for annotations are chosen based on the amount information they would give to the classifier. Two decision level fusion methods are proposed, and experiments are conducted to test their event detection capabilities. Both methods showed improvements to the unusual event detection results when compared to

the unsupervised method applied alone. The winner-takes-all method showed results more similar to the unsupervised method, where mainly undefined unusual events are detected. The support vector machine approach to fusion of the classifiers showed similarity to the supervised method, where known unusual events are detected. With this method, undefined unusual events are also detected which is the desired outcome of the combined unusual event detection system.

## **7.4 Future Work**

An extensive literature review and the experiments conducted provided a basis for implementation of a real-world unusual event detection system in surveillance applications. The proposed solution conforms to requirements of real-world surveillance applications identified at the beginning of the thesis. However, several issues remain to be addressed. In addition to these, several research areas that are related to the unusual event detection but are excluded from the thesis are considered in this section.

Even though an unusual event detection system has been proposed, and its components were tested on benchmark surveillance dataset. However, evaluation of the implemented system would benefit from the application of the system in real surveillance scenarios where the experienced security officer could provide feedback. This feedback would be invaluable input for the further research perspectives in the area. An important part for gathering the feedback from real surveillance setting is the graphical user interface (GUI). It is a challenging task to convey event detection information to the user in a simple yet informative way. An example interface was designed and can be seen in Appendix A. The proposed GUI contains online and offline mode, with a live camera feed and the event detection results showed on the side. The interface provides the capability to manually annotate the detected events which would allow more specific event

detection results. The GUI is designed for a single video camera, but could easily be upgraded to contain multiple cameras. The live feed area would have to be divided into grid in order to accommodate multiple video feeds. The events detected from all the cameras would have to be placed into a single list and ordered by time of event occurrence. A filter for the events also could be integrated globally or per camera to reduce the amount of events.

Events represented by video sensor data are limited to the visual clues. Improved event representation might be achieved by integrating a vision based system with other modalities, such as an audio signal. Kumar et al. (2005) combined audio information with the video descriptors to describe events. Integrating different modalities, such as temperature, motion, inertia, was suggested by Turaga et al. (2008). Ho et al. (2012) also mentioned integration of different modalities such as thermal, infrared, audio and pressure.

Relevance feedback is frequently applied in image retrieval to adapt to the user's information needs and to reduce the effort required for query composition (Rui et al., 1998). Similarly, an active learning approach can be used as an alternative to the intensive manual labeling. Several active learning schemes have been proposed in the literature to accelerate the learning process. The most informative samples are selected from the unlabeled sample pool according to certain criteria and the users are requested to label them. Most of the active learning methods empirically apply the closest-to-boundary criterion and choose the most uncertain samples for user annotations (Campbell et al., 2000). The supervised classifier can be updated with the newly labeled samples and this is an approach that should be investigated in the future.

## **7.5 Summary**

This chapter summarized the thesis, provided interpretation for the overall results and suggestions for future work. Starting with an overview of the results chapter by chapter, each chapter is briefly overviewed and the research questions, raised at the start of the thesis, are revisited. The chapter is concluded with the suggestions for the future work in this research area.

# Appendix A

## Graphical User Interface

The graphical user interface (GUI) was designed with the help of Dr. Hoywon Lee who is an expert in human-computer interaction. The GUI encapsulates the following functionalities - the live event detection (Figure A.1 - A.4) and the off-line search and management of the events (Figure A.5 - A.6). In the live event detection view, the live video feed is shown in the center, while the list of sequential events, both usual and unusual, are listed on the right panel (Figure A.2). A clickable timeline with marked unusual events is displayed on the top of the window. The Right side panel menu has an option to switch between listing all events, listing only unusual events, and listing only events that have been annotated by the user. When the option to view only unusual events is selected (Figure A.3), only the unusual events, marked by the system or the user, are listed at the right side panel. The events can be labeled by clicking on them, where the menu would pop-up with the list of available events. The options are to mark the event as usual, to give one of the labels from the list or to create a new label for the event (Figure A.4).

In the off-line mode, the GUI provides a functionality of event retrieval by the date, time or event type (Figure A.5). The labels of the events can also be set

in the off-line mode. The video of the selected event can be played, paused and rewind in the main window (Figure A.6).

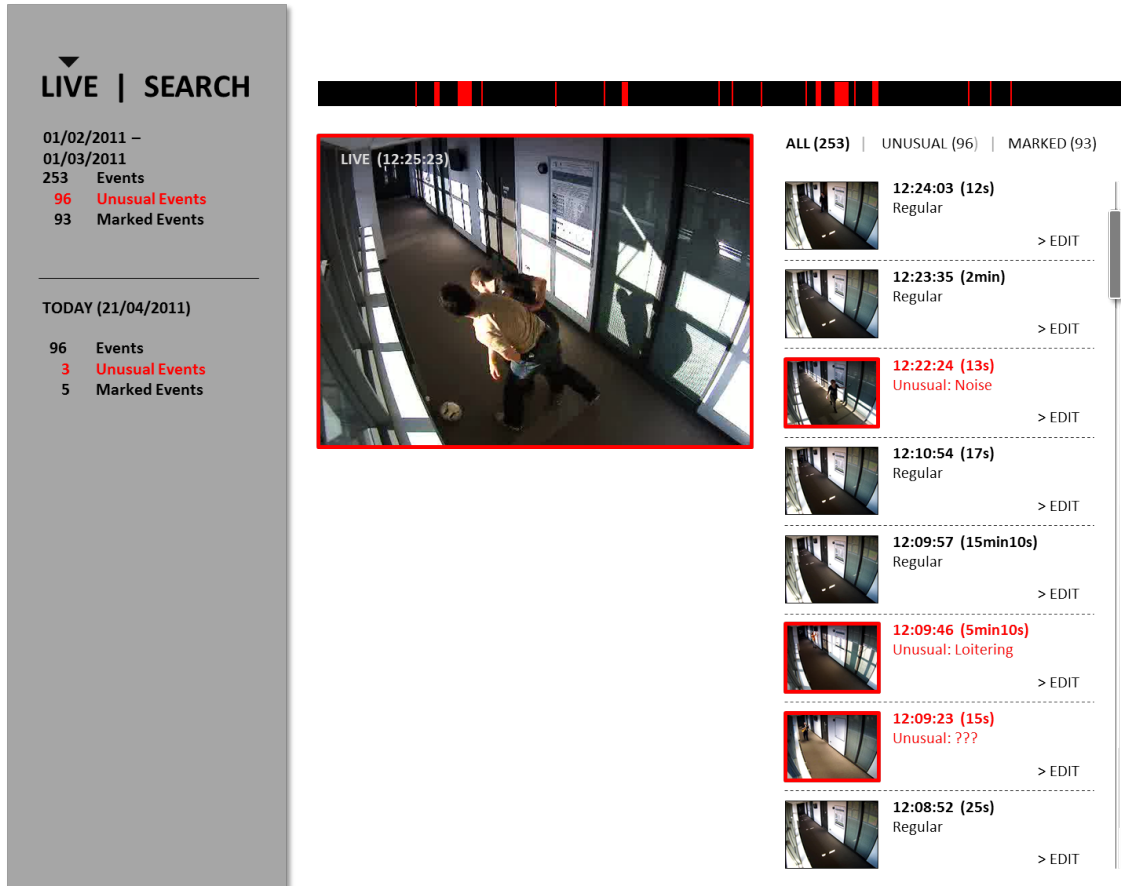


Figure A.1: Graphical user interface of the live unusual event detection system

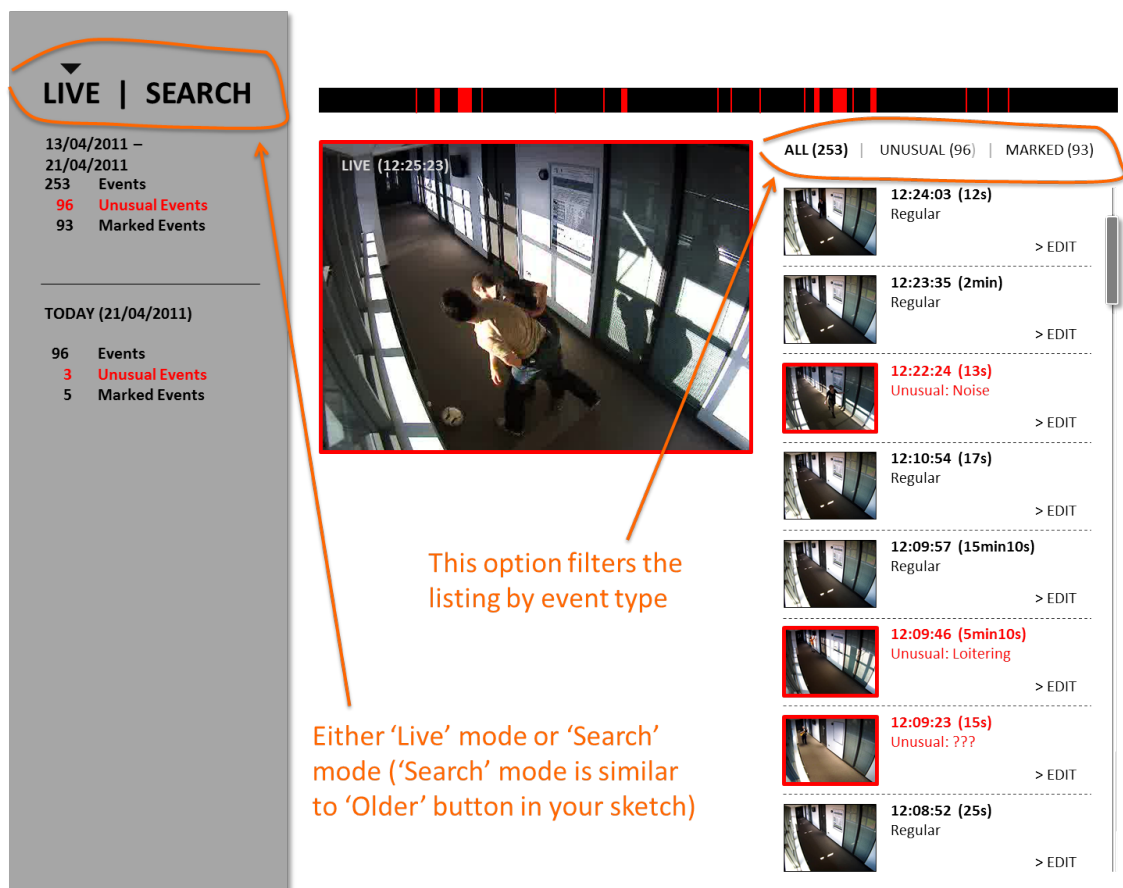


Figure A.2: Menu explanation of the live unusual event detection system GUI



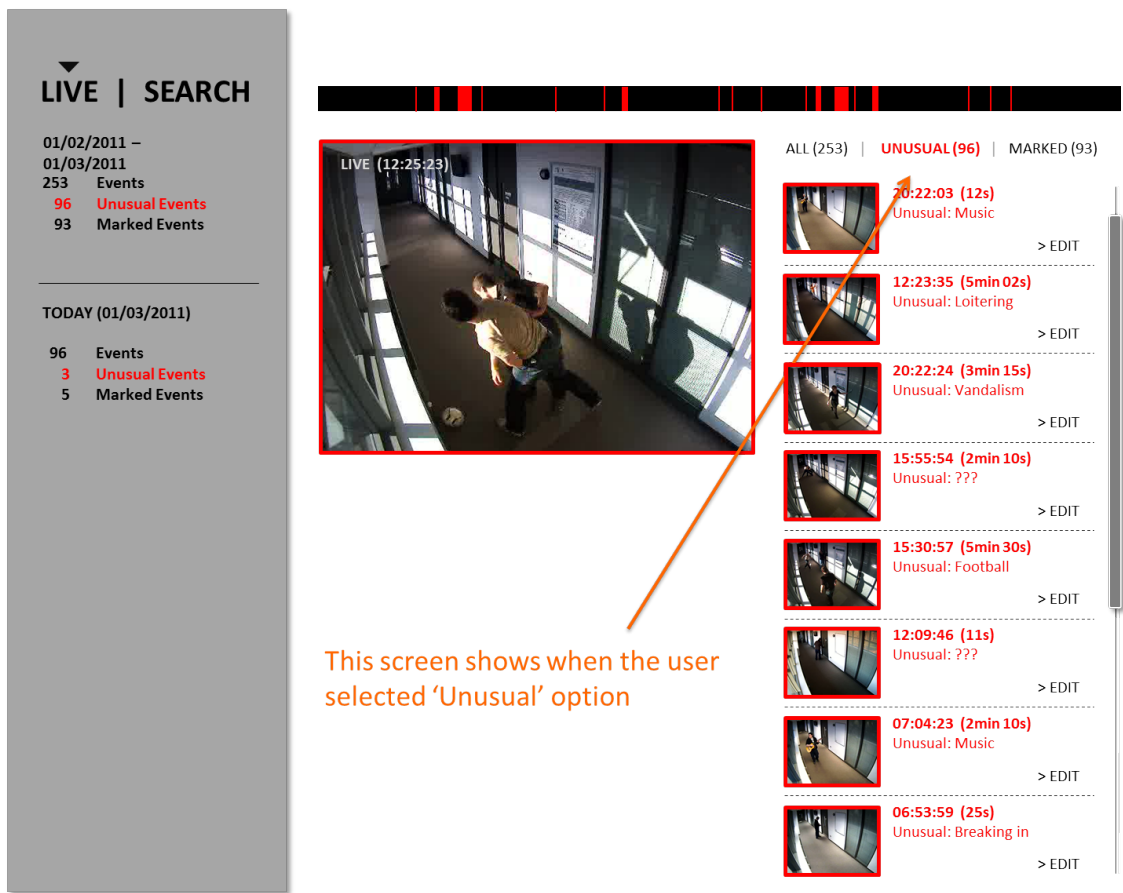


Figure A.3: Unusual events in the live unusual event detection system GUI

LIVE | SEARCH

01/02/2011 –  
 01/03/2011  
 253 Events  
 96 Unusual Events  
 93 Marked Events

---

TODAY (01/03/2011)

96 Events  
 3 Unusual Events  
 5 Marked Events

ALL (253) | UNUSUAL (96) | MARKED (93)

LIVE (12:25:23)

20:22:03 (12s)  
 Unusual: Music > EDIT

12:23:35 (5min 02s)  
 Unusual: Loitering > EDIT

20:22:24 (3min 15s)  
 Unusual: Vandalism > EDIT

Usual  
 Running  
 Bird  
 Stealing  
 Car  
 Light  
 Wind  
 Crowd  
 Ball  
 Add New (+)

07:04:23 (2min 10s)  
 Unusual: Music > EDIT

06:53:59 (25s)  
 Unusual: Breaking in > EDIT

Figure A.4: Changing event annotation in the live unusual event detection system GUI

**LIVE | SEARCH**

28/02/2011 – 01/03/2011  
 253 Events  
 96 Unusual Events  
 93 Marked Events

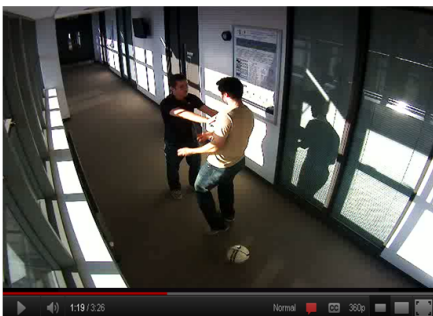
Date From  To

Time From  To

Event Type

**GO**

When the user selects one of the retrieved results, she/he can label it



12:22:24 (3min 26s)  
Unusual: ???

Label the unusual event by selecting type(s)


- Running
- Bird
- Fighting
- Car
- Light
- Wind
- Crowd
- Ball

▶

Fighting

Add New +


ALL (253) | UNUSUAL (96) | MARKED (93)



12:24:03 (12s)

Regular


> EDIT



12:23:35 (09s)

Regular


> EDIT



12:22:24 (13s)

Unusual: Fighting


> EDIT



12:10:54 (17s)

Regular


> EDIT



12:09:57 (08s)

Regular


> EDIT



12:22:24 (3min 26s)

Unusual: ???


> EDIT



12:09:23 (15s)

Unusual: Running

> EDIT



12:08:52 (25s)

Regular

> EDIT

Figure A.5: Off-line search in the off-line unusual event detection system GUI

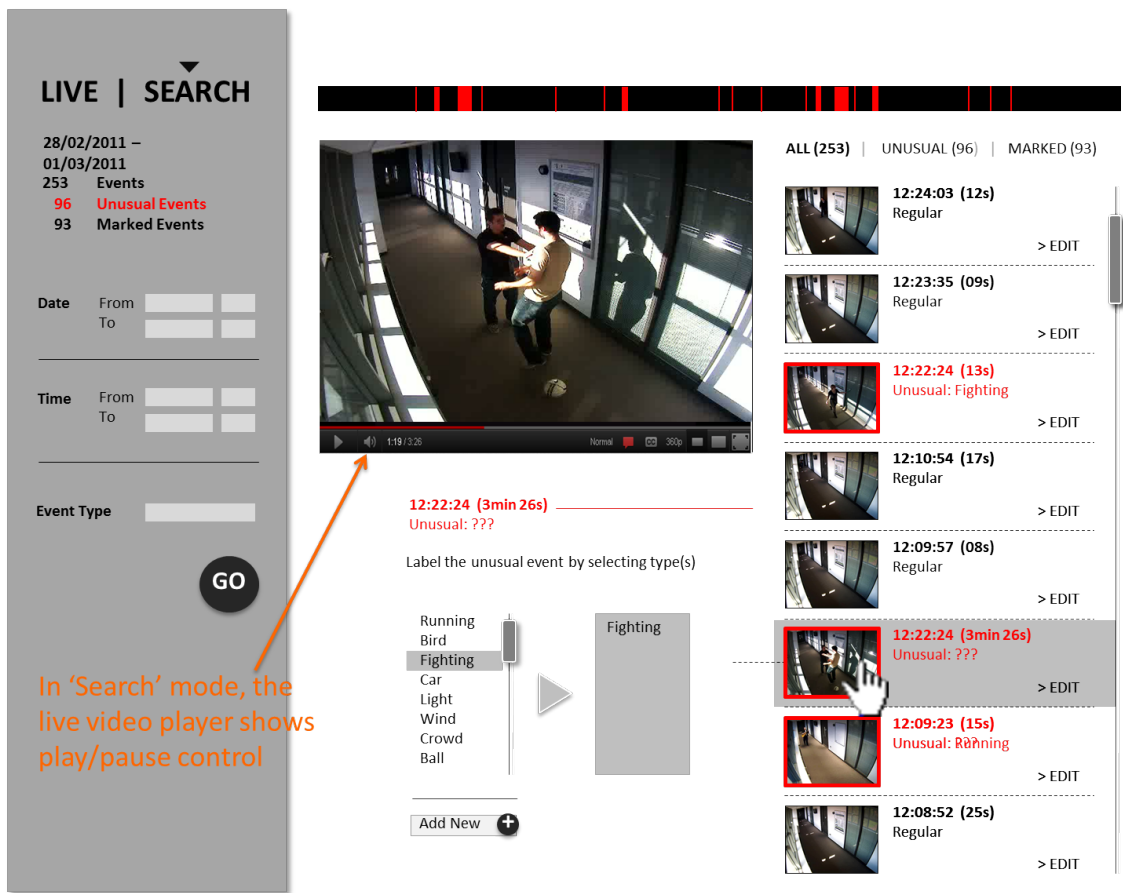


Figure A.6: Off-line search in the off-line unusual event detection system GUI

# Bibliography

- Acic (2014). <http://www.acic.eu/en/products/acic-activity-detection.html>. [Accessed 08/06/2014].
- Ade, R. R. and Deshmukh, P. R. (2013). Methods for incremental learning: a survey. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, 3(4):119–125.
- Agent Video Intelligence (2014). <http://www.agentvi.com/>. [Accessed 08/06/2014].
- AgilityVideo (2014). <http://www.vidient.com/>. [Accessed 08/06/2014].
- Ahad, A. R. (2011). *Computer Vision and Action recognition*, volume 5. Atlantis Ambient and Pervasive Intelligence.
- Aimetis (2014). <http://www.aimetis.com/>. [Accessed 08/06/2014].
- Albeshar, B., Kurugollu, F., Bouridane, A., and Baig, A. (2014). Cascaded multi-modal biometric recognition framework. *IET Biometrics*, 3(1):16–28.
- AllGoVision (2014). <http://www.allgovision.com/>. [Accessed 08/06/2014].
- AMI (2007). <http://corpus.amiproject.org/>. [Accessed 08/06/2014].
- Andrade, E. L., Blunsden, S., and Fisher, R. B. (2006). Modelling Crowd Scenes for Event Detection. In *18th International Conference on Pattern Recognition (ICPR)*, number August, pages 175–178. Ieee.

- Attwood, C. I. and Watson, D. A. (2004). Advisor-socket and see: Lessons learnt in building a real-time distributed surveillance system. In *IEE Intelligent Distributed Surveillance Systems*, number February, pages 6–11.
- AVSS (2007). [http://www.eecs.qmul.ac.uk/~andrea/avss2007\\_d.html](http://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html). [accessed: 08/06/2014].
- Basharat, A., Gritai, A., and Shah, M. (2008). Learning object motion patterns for anomaly detection and improved object detection. In *Computer Vision and Pattern Recognition (CVPR)*, number Anchorage, pages 1–8. IEEE.
- Ben-David, S. and Lindenbaum, M. (1997). Learning Distributions by Their Density Levels: A Paradigm for Learning without a Teacher. *Journal of Computer and System Sciences*, 55(1):171–182.
- Berger, M., Erlachert, F., Sommert, C., and Dresslert, F. (2014). Adaptive Load Allocation for Combining Anomaly Detectors Using Controlled Skips. *Computing, Networking and Communications (ICNC)*, pages 792–796.
- Bishop, C. M. (2006). *Pattern Recognition and Machine learning*. Information Science and Statistics. Springer.
- Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *Tenth International Conference on Computer Vision (ICCV)*, volume 2.
- Bobick, A. F. and Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis Machine Intelligence (PAMI)*, 23(3):257–267.
- Boiman, O. and Irani, M. (2005). Detecting Irregularities in Images and in Video. In *Tenth IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 462–469.

- Bousquet, O. and Elisseeff, A. (2002). Stability and generalization.
- Bowden, R. and Kaewtrakulpong, P. (2005). Towards automated wide area visual surveillance : tracking objects between spatially separated , uncalibrated views. *Image (Rochester, N.Y.)*, pages 213–223.
- Bregonzio, M., Gong, S., and Xiang, T. (2009a). Action recognition with cascaded feature selection and classification. In *3rd International Conference on Imaging for Crime Detection and Prevention (ICDP)*, number Dec, pages 1–6. Iet.
- Bregonzio, M., Gong, S., and Xiang, T. (2009b). Recognising action as clouds of space-time interest points. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number June, pages 1948–1955. Ieee.
- Breitenstein, M. D., Grabner, H., and Gool, L. V. (2009). Hunting nessie - real-time abnormality detection from webcams. In *Workshop On Visual Surveillance*, number September, pages 1243–1250.
- Brezovan, M. and Badica, C. (2013). A Review on Vision Surveillance Techniques in Smart Home Environments. In *19th International Conference on Control Systems and Computer Science (CSCS)*, number May, pages 471–478. Ieee.
- BRSLABS (2014). <http://www.brslabs.com>. [Accessed 14/06/2014].
- Buch, N., Velastin, S. A., and Orwell, J. (2011). A Review of Computer Vision Techniques for the Analysis of Urban Traffic. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):920–939.
- Campbell, C., Cristianini, N., and Smola, A. (2000). A Query learning with large margin classifiers. In *17th International Conference On Machine Learning (ICML)*, pages 111–118.

- Candamo, J., Shreve, M., Goldgof, D. B., Sapper, D. B., and Kasturi, R. (2010). Understanding transit scenes: a survey on human behavior-recognition algorithms. *IEEE transactions on intelligent transportation systems*, 11(1):206–224.
- CANDELA (2005). <http://www.multitel.be/~va/candela/>. [Accessed 08/06/2014].
- Carincotte, C., Desurmont, X., Ravera, B., Bremond, F., Orwell, J., Velastin, S. A., Odobez, J. M., Corbucci, B., Palo, J., and Cenocky, J. (2006). Toward generic intelligent knowledge extraction from video and audio: The EU-funded caretaker project. In *The Institution of Engineering and Technology Conference on Crime and Security*, number June, pages 470–475.
- Chakraborty, B., Holte, M. B., Moeslund, T. B., and González, J. (2012). Selective spatio-temporal interest points. *Computer Vision and Image Understanding*, 116(3):396–410.
- Chandola, V., Banerjee, A., and Kumar, V. (2007a). Anomaly Detection: A Survey. Technical Report August.
- Chandola, V., Banerjee, A., and Kumar, V. (2007b). Outlier detection: A survey. Technical report.
- Chandola, V., Banerjee, A., and Kumar, V. (2012). Anomaly detection for discrete sequences: a survey. *IEEE transactions on knowledge and data engineering*, 24(5):823–839.
- Chandola, V., Mithal, V., and Kumar, V. (2008). Comparative Evaluation of Anomaly Detection Techniques for Sequence Data. In *Eighth IEEE International Conference on Data Mining*, number December, pages 743–748. Ieee.
- Chaquet, J. M., Carmona, E. J., and Fernández-Caballero, A. (2013). A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, 117(6):633–659.



- Chen, D., Yang, J., and Wactlar, H. D. (2004). Towards automatic analysis of social interaction patterns in a nursing home environment from video. In *6th ACM SIGMM international workshop on Multimedia information retrieval (MIR)*, page 283, New York, New York, USA. ACM Press.
- CMU (2005). <http://www.informedia.cs.cmu.edu/arda/vaceI.html>. [Accessed 29/06/2014].
- CMU (2010). <http://www.cs.cmu.edu/~vsam/>. [Accessed 29/06/2014].
- Cognimatics (2014). <http://www.cognimatics.com/>. [Accessed 14/06/2014].
- Collins, R., Lipton, A., Fujiyoshi, H., and Kanade, T. (2001). Algorithms for cooperative multisensor surveillance. *Proc. IEEE*, 89(10):1456–1477.
- Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–22.
- Cucchiara, R. (2005). Multimedia surveillance systems. In *3rd ACM international workshop on Video surveillance & sensor networks (VSSN)*, pages 3–10, New York, New York, USA.
- Cucchiara, R., Grana, C., Piccardi, M., and Prati, A. (2001). Detecting objects, shadows and ghosts in video streams by exploiting color and motion information. In *Lecture Notes in Computer Science (ACCV)*, pages 360–365.
- Dalal, N., Triggs, B., and Schmid, C. (2006). Human Detection Using Oriented Histograms of Flow and Appearance. *Lecture Notes in Computer Science (ACCV)*, 3952:428–441.
- Dambra, C. (2014). <http://www.protectrail.eu/>. [Accessed 29/06/2014].

- Danafar, S. and Gheissari, N. (2007). Action recognition for surveillance applications using optic flow and SVM. *Lecture Notes in Computer Science*, 4844:457–466.
- DARPA (2002). <http://fas.org/irp/program/process/avs.htm>. [Accessed 29/06/2014].
- DARPA (2010). [www.darpa.mil/ipto/programs/virat/virat.asp](http://www.darpa.mil/ipto/programs/virat/virat.asp). [Accessed 29/06/2014].
- Davies, A. C. and Velastin, S. A. (1995). Crowd monitoring using image processing. *Electronics & Communications Engineering Journal*, 7(1):37.
- Davies, A. C. and Velastin, S. A. (2005). A Progress Review of Intelligent CCTV Surveillance Systems. In *IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS Workshop*, number September, pages 417–423.
- Davis, J. and Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *23rd international conference on Machine learning (ICML)*, pages 233–240, New York, New York, USA.
- Dawson, D., Derby, P., Doyle, A., Fonio, C., Huey, L., and Johnson, M. (2009). A Report on Camera Surveillance in Canada, Part two. Technical report, Social Sciences and Humanities Research Council of Canada.
- De Silva, L. C., Morikawa, C., and Petra, I. M. (2012). State of the art of smart homes. *Engineering Applications of Artificial Intelligence*, 25(7):1313–1321.
- Dee, H. and Hogg, D. (2004). Detecting inexplicable behaviour. In *British Machine Vision Conference (BMVC)*, pages 477–486.
- Dollar, P., Rabaud, V., Cottrell, G., and Belongie, S. (2005). Behavior Recognition via Sparse Spatio-Temporal Features. In *2nd Joint IEEE International Workshop*

*on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, number October, pages 65–72.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*.

Dvtel (2014). <http://www.dvtel.com/products-solutions/video-analytics/>. [Accessed 08/06/2014].

Efros, A. A., Berg, A. C., Mori, G., and Malik, J. (2003). Recognizing action at a distance. In *Ninth IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 726–733. Ieee.

Eptascope (2014). <http://www.eptascope.com/>. [Accessed 08/06/2014].

Evans, M. and Ferryman, J. (2010). Surveillance Camera Calibration from Observations of a Pedestrian. *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 64–71.

Feng, J., Zhang, C., and Hao, P. (2012). Online anomaly detection in videos by clustering dynamic exemplars. In *19th International Conference on Image Processing (ICIP)*, number September, pages 3097–3100.

Fernández-Caballero, A., Castillo, J. C., and Rodríguez-Sánchez, J. M. (2012). Human activity monitoring by local and global finite state machines. *Expert Systems with Applications*, 39(8):6982–6993.

Foresti, G. L., Marcenaro, L., and Regazzoni, C. S. (2002). Automatic detection and indexing of video-event shots for surveillance applications. *IEEE transactions on Multimedia*, 4(4):459–471.

Foucher, S., Lalonde, M., and Gagnon, L. (2011). A system for airport surveillance: Detection of people running, abandoned objects and pointing gestures. In *SPIE Defense & Security Symposium: Visual Information Processing*.

- Foxstream (2014). <http://www.foxstream.fr/>. [Accessed 14/06/2014].
- French, A. P. (2005). *Visual Tracking: From An Individual To Groups of Animals*. PhD thesis, University of Nottingham.
- Gao, Q. and Sun, S. (2013). Trajectory-based human activity recognition with hierarchical Dirichlet process hidden Markov models. In *IEEE China Summit and International Conference on Signal and Information Processing*, pages 456–460.
- Georis, B., Maziere, M., Bremond, F., and Thonnat, M. (2004). A video interpretation platform applied to bank agency monitoring. In *Intelligent Distributed Surveillance Systems (IDSS)*, pages 46–50.
- Goldgof, D. B., Sapper, D., Candamo, J., and Shreve, M. (2009). Evaluation of Smart Video for Transit Event Detection. Technical report.
- Gong, S. and Xiang, T. (2003). Recognition of group activities using dynamic probabilistic networks. In *Ninth IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 742–750.
- Guedalia, I. D., London, M., and Werman, M. (1999). An On-Line Agglomerative Clustering Method for Nonstationary Data. *Neural Computation*, 11(2):521–540.
- GVU (2004). <http://www.cc.gatech.edu/cpl/projects/hid/>. [Accessed 29/06/2014].
- Haering, N., Venetianer, P. L., and Lipton, A. (2008). The evolution of video surveillance: an overview. *Machine Vision and Applications*, 19:279–290.
- Hakeem, A. and Shah, M. (2004). Ontology and taxonomy collaborated framework for meeting classification. In *17th International Conference of Pattern Recognition (ICPR)*, volume 4, pages 219–222.
- Hamid, R., Johnson, A., Batta, S., Bobick, A., Isbell, C., and Coleman, G. (2005). Detection and Explanation of Anomalous Activities: Representing Activities

- as Bags of Event N-Grams. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1031–1038.
- Han, B., Comaniciu, D., Zhu, Y., and Davis, L. S. (2008). Sequential kernel density approximation and its application to real-time visual tracking. *IEEE transactions on pattern analysis and machine intelligence*, 30(7):1186–1197.
- Haritaoglu, I., Harwood, D., and Davis, L. S. (2000). W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830.
- Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey Vision Conference*, volume 15, pages 50–56. Manchester, UK.
- Ho, T. K., Matthews, K., O’Gorman, L., and Steck, H. (2012). Public Space Behavior Modeling With Video and Sensor Analytics. *Bell Labs Technical Journal*, 16(4):203–217.
- Hodge, V. J. and Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, (22):85–126.
- Honeywell (2014). <http://www.honeywellvideo.com/>. [Accessed 08/06/2014].
- Hu, W., Tan, T., Wang, L., and Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 34(3):334–352.
- Huo, J., Gao, Y., Yang, W., and Yin, H. (2014). Abnormal Event Detection via Multi-Instance. *International Journal of Neural Systems*, pages 76–83.
- IBM (2014). [http://www.ibm.com/smarterplanet/ie/en/transportation\\_systems/nextsteps/](http://www.ibm.com/smarterplanet/ie/en/transportation_systems/nextsteps/) [Accessed 29/06/2014].

- IndigoVision (2014). <http://www.indigovision.com/video-analytics>. [Accessed 08/06/2014].
- INRIA (2004). <http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>. [Accessed 08/06/2014].
- INRIA (2011). <http://www-sop.inria.fr/orion/ETISEO/>. [Accessed 28/06/2014].
- IntelliView (2014). <http://www.intelliview.ca/>. [Accessed 08/06/2014].
- IntelliVision (2014). <http://www.intelli-vision.com/products/intelligent-video-analytics>. [Accessed 08/06/2014].
- Iomniscient (2014). <http://www.iomniscient.com/>. [Accessed 14/06/2014].
- Ipsotek (2014). <http://www.ipsotek.com/>. [Accessed 08/06/2014].
- Jargalsaikhan, I., Little, S., Direkoglu, C., and OConnor, N. E. (2013). Action recognition based on the sparse trajectory. In *20th IEEE International Conference on Image Processing (ICIP)*, number September, pages 3982–3985.
- Johnson, A. Y. and Bobick, A. F. (2001). A Multi-view Method for Gait Recognition Using Static Body Parameters. *AVBPA*, pages 301–311.
- Junior, O. L., Delgado, D., Goncalves, V., and Nunes, U. (2009). Trainable classifier-fusion schemes: An application to pedestrian detection. In *12th International IEEE Conference on Intelligent Transportation Systems*, number October, pages 432–437.
- Juvonen, A. and Sipola, T. (2013). Combining conjunctive rule extraction with diffusion maps for network intrusion detection. *2013 IEEE Symposium on Computers and Communications (ISCC)*, pages 000411–000416.
- Kastrinaki, V., Zervakis, M., and Kalaitzakis, K. (2003). A survey of video processing techniques for traffic applications. *Image and Vision Computing*, 21:359–381.

- Ke, Y., Sukthankar, R., and Hebert, M. (2005). Efficient visual event detection using volumetric features. In *Tenth IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 166–173. Ieee.
- Kim, I. S., Choi, H. S., Yi, K. M., Choi, J. Y., and Kong, S. G. (2010). Intelligent visual surveillance A survey. *International Journal of Control, Automation and Systems*, 8(5):926–939.
- Kim, J. and Grauman, K. (2009). Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number June, pages 2921–2928. Ieee.
- Kitani, K. M., Sato, Y., and Sugimoto, A. (2005). Deleted Interpolation Using a Hierarchical Bayesian Grammar Network for Recognizing Human Activity. In *2nd Joint IEEE International Workshop on VS-PETS*, number October, pages 239–246.
- Klaser, A., Marszalek, M., and Schmid, C. (2008). A spatio-temporal descriptor based on 3D-gradients. *British Machine Vision Conference (BMVC)*.
- Koch, R. (2011). Towards Next-Generation Intrusion Detection. *Cyber Conflict (ICCC)*, pages 1–18.
- Kolekar, P. (2013). Global and China surveillance cameras industry market research report. Technical report, Market Research.
- Kovashka, A. and Grauman, K. (2010). Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2046–2053. Ieee.

- Kumar, P., Mittal, A., and Kumar, P. (2005). A Multimodal Audio Visible and Infrared Surveillance System (MAVISS). In *Intelligent Sensing and Information Processing (ICISIP)*, pages 151–156.
- Lagrange, P. (2003). [http://www.transport-research.info/web/projects/project\\_details.cfm?ID=13699](http://www.transport-research.info/web/projects/project_details.cfm?ID=13699). [Accessed 29/06/2014].
- Laptev, I. (2004). *Local Spatio-Temporal Image Features for Motion Interpretation*. PhD thesis, KTH Numerical Analysis and Computer Science.
- Laptev, I. and Lindeberg, T. (2003). Space-time interest points. In *Ninth IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 432–439. Ieee.
- Laptev, I. and Lindeberg, T. (2006). Local Descriptors for Spatio-temporal Recognition. *Lecture Notes in Computer Science*, 3667:91–103.
- Laptev, I., Marszalek, M., Schmid, C., and Rozenfeld, B. (2008). Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. Ieee.
- Laptev, I. and Perez, P. (2007). Retrieving actions in movies. In *11th IEEE International Conference on Computer Vision (ICCV)*, pages 1–8. Ieee.
- Lavee, G., Rivlin, E., and Rudzsky, M. (2009). 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*, 39(5):489–504.
- Lazarevic, A., Kumar, V., and Srivastava, J. (2005). Intrusion detection: a survey. In *Managing Cyber Threats*, pages 19–78.



- Lecomte, S., Lengelle, R., Richard, C., Capman, F., and Ravera, B. (2011). Abnormal Events Detection using Unsupervised One-Class SVM Application to Audio Surveillance and Evaluation . In *8th International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 124–129.
- Liang, N.-Y., Huang, G.-B., Saratchandran, P., and Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, 17(6):1411–23.
- Lindeberg, T. (1998). Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, 30(2):79–116.
- Liu, H., Chen, S., and Kubota, N. (2013). Intelligent Video Systems and Analytics: A Survey. *IEEE transactions on industrial informatics*, 9(3):1222–1233.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.
- Loy, C. C., Xiang, T., and Gong, S. (2010). Stream-based active unusual event detection. In *ACCV 2010*, pages 1–14.
- Lu, Y., Boukharouba, K., Boonæ rt, J., Fleury, A., and Lecœ uche, S. (2014). Application of an incremental SVM algorithm for on-line human recognition from video surveillance using texture and color features. *Neurocomputing*, 126:132–140.
- Lucas, B. D. and Kanade, T. (1981a). An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 1981 DARPA Imaging Understanding Workshop*, pages 121–130.
- Lucas, B. D. and Kanade, T. (1981b). An Iterative Image Registration Technique with an Application to Stereo Vision. *Proceedings of the 1981 DARPA Imaging Understanding Workshop*, pages 121–130.

- Luxburg, U. V. (2010). Clustering stability: an overview. *Foundations and Trends in Machine Learning*, 2(3):235–274.
- Lv, F., Kang, J., Nevatia, R., Cohen, I., and Medioni, G. (2004). Automatic tracking and labeling of human activities in a video sequence. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, ECCV*.
- Maciejewski, R., Hafen, R., Rudolph, S., Tebbetts, G., Cleveland, W. S., Grannis, S. J., and D. S., E. (2009). Generating Synthetic Syndromic- Surveillance Data for Evaluating. *IEEE Computer Graphics and Applications*, 29(3).
- Mahadevan, V., Li, W., Bhalodia, V., and Vasconcelos, N. (2010). Anomaly detection in crowded scenes. *Computer Vision and Pattern Recognition (CVPR)*, pages 1975–1981.
- Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2009). Online dictionary learning for sparse coding. *Conference on Machine Learning*, pages 689–696.
- Mango (2014). <http://www.mangodsp.com/Home.aspx>. [Accessed 08/06/2014].
- MarchNetworks (2014). <http://www.marchnetworks.com/products/video-analytics/default.aspx>. [Accessed 08/06/2014].
- Marín-Jiménez, M. J., Yeguas, E., and Pérez de la Blanca, N. (2013). Exploring STIP-based models for recognizing human interactions in TV videos. *Pattern Recognition Letters*, 34(15):1819–1828.
- Markou, M. and Singh, S. (2003a). Novelty detection: a reviewpart 1: statistical approaches. *Signal Processing*, 83(12):2481–2497.
- Markou, M. and Singh, S. (2003b). Novelty detection: a reviewpart 2: neural network based approaches. *Signal Processing*, 83(12):2499–2521.

- Marszalek, M., Laptev, I., and Schmid, C. (2009). Actions in context. In *Computer Vision and Pattern Recognition, CVPR*, number June, pages 2929 – 2936.
- Matsugu, M., Yamanaka, M., and Sugiyama, M. (2011). Detection of activities and events without explicit categorization. *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1532–1539.
- McCahill, M. and Norris, C. (2002). CCTV in London. Technical report.
- Mehran, R., Oyama, A., and Shah, M. (2009). Abnormal crowd behavior detection using social force model. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (2):935–942.
- Mikolajczyk, K. and Schmid, C. (2005). Performance evaluation of local descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 27(10):1615–30.
- Ming Du, Sankaranarayanan, A. C., and Chellappa, R. (2014). Robust face recognition from multi-view videos. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 23(3):1105–17.
- Mosabbeh, E. A., Raahemifar, K., and Fathy, M. (2013). Distributed activity recognition in camera networks via low-rank matrix recovery. *2013 Seventh International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–2.
- Mubashir, M., Shao, L., and Seed, L. (2013). A survey on fall detection: Principles and approaches. *Neurocomputing*, 100:144–152.
- Mudjirahardjo, P., Tan, J. K., Kim, H., and Ishikawa, S. (2013). Abnormal Motion Detection in an Occlusive Environment. In *Conference of the Society of Instrument and Control Engineers of Japan, SICE*, pages 1398–1402.
- Nam, Y. and Hong, S. (2014). Real-time abnormal situation detection based on particle advection in crowded scenes. *Journal of Real-Time Image Processing*.

- Naylor, M. (2002). <http://www-sop.inria.fr/orion/ADVISOR/>. [Accessed 29/06/2014].
- Naylor, M. and Bastin, D. (2003). Final Report IST1999-11287 : ADVISOR. (1).
- Nice (2014). <http://www.nice.com/analytics>. [Accessed 08/06/2014].
- Niebles, J. C., Wang, H., and Fei-Fei, L. (2006). Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words. *Proceedings of the British Machine Vision Conference 2006*, pages 127.1–127.10.
- ObjectVideo (2014). <http://www.objectvideo.com/>. [Accessed 08/06/2014].
- OTCBVS (2007). <http://www.cse.ohio-state.edu/otcbvs-bench/>. [Accessed 08/06/2014].
- Ouivirach, K., Gharti, S., and Dailey, M. N. (2013). Incremental behavior modeling and suspicious activity detection. *Pattern Recognition*, 46(3):671–680.
- Parker, J. (2010). *Algorithms for image processing and computer vision*.
- Patcha, A. and Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448–3470.
- Patrick, R. and Bourbakis, N. (2009). Surveillance Systems for Smart Homes: A Comparative Survey. In *IEEE International Conference on Tools with Artificial Intelligence*, number Nov., pages 248–252. Ieee.
- Pentland, A. P. (1996). Smart Rooms. *Scientific American*, (April):54–62.
- PETS (2006). <http://www.cvg.rdg.ac.uk/slides/pets.html>. [Accessed 08/06/2014].
- Pontil, M. and Verri, A. (1998). Properties of support vector machines. *Neural Computation*, (I):1–18.

- Popoola, O. P. and Wang, K. (2012). Video-Based Abnormal Human Behavior Recognition A Review. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 42(6):865–878.
- Porikli, F. and Haga, T. (2004). Event Detection by Eigenvector Decomposition Using Object and Frame Features. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 114–114. Ieee.
- Puretechsystems (2014). <http://www.puretechsystems.com/video-analytics.html>. [Accessed 08/06/2014].
- Qi, W., Qiao, F., and Yang, H. (2012). Distributed Smart Camera Network. *Recent Patents on Electrical Engineering*, 5(1):20–29.
- Rababaah, A. R. (2012). A Survey of Intelligent Visual Surveillance Systems. In *International Conference on Image Processing, Computer Vision, and Pattern Recognition*, volume 2, pages 1–7.
- Raty, T. D. (2010). Survey on Contemporary Remote Surveillance Systems for Public Safety. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 40(5):493–515.
- Ravera, B. (2008). [http://cordis.europa.eu/ist/kct/caretaker\\_synopsis.htm](http://cordis.europa.eu/ist/kct/caretaker_synopsis.htm). [Accessed 29/06/2014].
- Rinner, B. and Wolf, W. (2008). An Introduction to Distributed Smart Cameras. *Proceedings of the IEEE*, 96(10):1565–1575.
- Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. (2007). Incremental Learning for Robust Visual Tracking. *International Journal of Computer Vision*, 77(1-3):125–141.
- Rui, Y., Huang, T., Ortega, M., and Mehrotra, S. (1998). Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655.

- Sabahi, F. and Movaghar, A. (2008). Intrusion Detection: A Survey. In *Third International Conference on Systems and Networks Communications*, pages 23–26. Ieee.
- Saldatos, J. (2009). THIS - Transport Hub Intelligent video System. *Center of Excellence for Research and Education*.
- Schuldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local SVM approach. In *17th International Conference on Pattern Recognition (ICPR)*, pages 3–7.
- Senst, T., P, M., Evangelio, H., Eiselein, V., Keller, I., and Sikora, T. (2011). On Building Decentralized Wide-Area Surveillance Networks based on ONVIF. *Advanced Video and Signal Based Surveillance*, pages 420–423.
- Settles, B. (2010). Active Learning Literature Survey. In *University of Wisconsin, Madison*. University of Wisconsin–Madison, Citeseer.
- Shah, M., Javed, O., and Shafique, K. (2007). Automated Visual Surveillance in Realistic Scenarios. *IEEE Multimedia*, 14(1):30–39.
- Shi, Y. and Lichman, S. (2005). Smart Cameras: A Review. Technical report, Australian Technology Park, Bay 15 Locomotive Workshop.
- Sightlogix (2014). <http://www.sightlogix.com/>. [Accessed 08/06/2014].
- Sillito, R. and Fisher, R. (2008). Semi-supervised learning for anomalous trajectory detection. In *British Machine Vision Conference (BMVC)*, pages 1035–1044. Citeseer.
- Sinha, A., Chen, H., Danu, D. G., Kirubarajan, T., and Farooq, M. (2008). Estimation and decision fusion: A survey. *Neurocomputing*, 71(13):2650–2656.

- Smits, P. (2002). Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection. *IEEE Transactions on Geoscience and Remote Sensing*, 40(4):801–813.
- Sodemann, A. A., Ross, M. P., and Borghetti, B. J. (2012). A Review of Anomaly Detection in Automated Surveillance. *Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 42(6):1257–1272.
- Stauffer, C. and Grimson, W. E. L. (1999). Adaptive background mixture models for real-time tracking. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:1–7.
- Stöttinger, J., Goras, B., Pöntiz, T., Hanbury, A., Sebe, N., and Gevers, T. (2011). Systematic evaluation of spatio-temporal features on comparative video challenges. *ACCV Workshop*, pages 349–358.
- Sudo, K., Osawa, T., Tanaka, H., Koike, H., and Arakawa, K. (2008). Online anomal movement detection based on unsupervised incremental learning. In *9th International Conference on Pattern Recognition (ICPR)*, pages 1–4. Ieee.
- Sun, Z., Bebis, G., and Miller, R. (2006). On-road vehicle detection: a review. *IEEE transactions on pattern analysis and machine intelligence*, 28(5):694–711.
- Syed, N., Huan, S., Kah, L., and Sung, K. (1999a). Incremental learning with support vector machines. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1–6.
- Syed, N. A., Liu, H., and Sung, K. K. (1999b). Handling concept drifts in incremental learning with support vector machines. In *5th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 317–321, New York, New York, USA. ACM Press.

- SYNTAXIS (2014). <http://www.synaxissecurity.com/products/video-management-surveillance-systems/truesentry-intelligent-ip-video-surveillance-system/>. [Accessed 08/06/2014].
- TRECVID (2007). <http://trecvid.nist.gov/trecvid.data.html>. [Accessed 08/06/2014].
- Turaga, P., Chellappa, R., Subrahmanian, V. S., and Udrea, O. (2008). Machine Recognition of Human Activities: A Survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488.
- UCSD (2010). <http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm>. [Accessed 08/06/2014].
- U.K. Home Office (2011). <https://www.gov.uk/imagery-library-for-intelligent-detection-systems>. [Accessed 28/06/2014].
- Umakanthan, S., Denman, S., Sridharan, S., Fookes, C., and Wark, T. (2012). Spatio Temporal Feature Evaluation for Action Recognition. In *International Conference on Digital Image Computing Techniques and Applications (DICTA)*, pages 1–8. Ieee.
- Valera, M. and Velastin, S. (2005). Intelligent distributed surveillance systems: a review. *Vision, Image and Signal Processing*, 152(2):192–204.
- Velastin, S. A., Boghossian, B. A., Ping, B., Lo, L., Sun, J., and Vicencio-Silva, M. A. (2005). PRISMATICA : Toward Ambient Intelligence in Public Transport Environments. *Systems, Man, and Cybernetics - Part A: Systems and Humans*, 35(1):164–182.
- Verint (2014). <http://www.verint.com/solutions/video-situation-intelligence/products/surveillance-analytics/index>. [Accessed 08/06/2014].
- VideoIQ (2014). <http://www.videoiq.com/>. [Accessed 15/06/2014].



- Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 511–518.
- Wang, H., Klaser, A., Schmid, C., and Liu, C.-l. (2011). Action recognition by dense trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition (CVPR)*, number June, pages 3169–3176.
- Wang, H., Kläser, A., Schmid, C., and Liu, C.-l. (2012). Dense trajectories and motion boundary descriptors for action recognition. Technical report, Project-Teams LEAR and NLPR CASIA.
- Wang, H., Ullah, M. M., Klaser, A., Laptev, I., and Schmid, C. (2009a). Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference (BMVC)*, pages 124.1–124.11. British Machine Vision Association.
- Wang, W., Luo, J., and Qi, H. (2013). Action recognition across cameras via reconstructable paths. ... *Smart Cameras (ICDSC), 2013 Seventh ...*, pages 1–6.
- Wang, X., Ma, X., and Grimson, W. E. L. (2009b). Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE transactions on pattern analysis and machine intelligence*, 31(3):539–55.
- Wang, Y., Jiang, H., Drew, M., Li, Z.-N., and Mori, G. (2006). Unsupervised discovery of action classes. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1654–1661. Ieee.
- Webb, A. R. (2002). *Statistical Pattern Recognition*, volume 9. John Wiley & Sons, Ltd.
- Weber, M. and Stone, M. (1994). Low altitude wind shear detection using airport surveillance radars. In *IEEE National Radar Conference*, number March, pages 52–57.

- Weinland, D., Ronfard, R., and Boyer, E. (2011). A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241.
- Willems, G., Tuytelaars, T., and Gool, L. V. (2008). An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. In *European Conference on Computer Vision*, pages 650–663.
- Wilson, D. and Sutton, A. (2003). Open-Street CCTV in Australia : A comparative study of establishment and operation. Technical Report April, University of Melbourne.
- Wren, C., Azarbayejani, A., Darrell, T., and Pentland, A. P. (1997). Pfunder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785.
- Xiang, T. and Gong, S. (2005). Video behaviour profiling and abnormality detection without manual labelling. In *Tenth IEEE International Conference on Computer Vision (ICCV)*, pages 1238–1245. Ieee.
- Xiang, T. and Gong, S. (2006). Beyond tracking: Modelling activity and understanding behaviour. *International Journal of Computer Vision*, 67(1):21–51.
- Xiang, T. and Gong, S. (2008). Incremental and adaptive abnormal behaviour detection. *Computer Vision and Image Understanding*, 111(1):59–73.
- Xu, Y., Yan, Y., Huang, H., Yang, H., and Zhou, R. (2014). Voice biometrics using linear Gaussian model. *IET Biometrics*, 3(1):9–15.
- Xue, H. and Liu, Z. (2013). Pedestrian Classification Based on Improved Support Vector Machines. In *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, pages 726–730. Ieee.

- Zhan, B., Monekosso, D., Remagnino, P., Velastin, S., and Xu, L. (2008). Crowd analysis: a survey. *Machine Vision and Applications*, 19(5):345–357.
- Zhong, H., Shi, J., and Visontai, M. (2004). Detecting unusual activity in video. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:819–826.
- Zia, M., Kim, T.-S., and Kim, J.-T. (2013). A Spatiotemporal Robust Approach for Human Activity Recognition. *International Journal of Advanced Robotic Systems*, 10:Article number 391.
- Ziliani, F., Velastin, S., Porikli, F., Marcenaro, L., Kelliher, T., Cavallaro, A., and Bruneaut, P. (2005). Performance evaluation of event detection solutions: the creds experience. *IEEE Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, pages 201–206.