Unsupervised Detection and Localization of Anomalous Motion Patterns in Surveillance Video

ABDULLAH A. ABUOLAIM

(BSCE. (Hons.), Jordan University of Science and Technology, Jordan, 2015)

A THESIS SUBMITTED
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SINGAPORE

2017

Supervisors:
Professor Narendra Ahuja, Main Supervisor
Associate Professor Leow Wee Kheng, Co-Supervisor
Examiners:
Associate Professor Ng Teck Khim
Associate Professor Terence Sim Mong Cheng

DECLARATION

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis. This thesis has also not been submitted for any degree in any university previously.

Signature:

Date: 29 August 2017

ABDULLAH A. ABUOLAIM
2017

ACKNOWLEDGMENTS

In the Name of Allah, the Most Gracious, the Most Merciful, the Most Compassionate. Praise belongs to Allah alone; prayers, peace and blessing upon the last Prophet Mohamed His servant and messenger.

First and foremost, I must acknowledge Allah with limitless thanks; the Ever-Thankful, for His bless. I strongly believe that this piece of work would have never been done, without His guidance.

I would like to express my gratitude to my father, Allah rest his soul. None of this could have been possible without his love. I love you and miss you every day, for my beloved mother, who taught me that even the largest tasks can be accomplished if it is done one step at a time and who inspired my scientific thinking, for my brothers and sisters for their endless love, support and encouragement.

I would like to thank my supervisors, Assoc. Prof. Leow Wee Kheng and Prof. Narendra Ahuja for their continuous support and guidance; they have made my work easier and more interesting. A special sincere gratitude towards Assoc. Prof. Wee Kheng, he trained me how to strengthen my thinking abilities, he taught me how to formulate a research problem in the right way, solve it, and write a good scientific reports. He was always responsible, and he kept providing insightful ideas. Many thanks to my thesis examiners Assoc. Prof. Ng Teck Khim and Assoc. Prof. Terence Sim Mong Cheng for their valuable comments and guidance.

I owe a deep debt of gratitude to A*STAR, ADSC research institute for offering me scholarship and for giving me an opportunity to complete this work. Also, I would like to offer my sincere appreciation to all of the faculty members in NUS for showing me by example and through challenging coursework, how to think. Finally, I want to thank my colleagues for criticizing my work and exchanging constructive discussions.

ABSTRACT

Surveillance cameras are widely installed in public places to monitor pedestrian activities for security purposes. An important surveillance application is to detect anomalous motion automatically, and notify the human observer using computerized methods. Many methods have been proposed for detecting anomalous motion patterns in surveillance videos. They can be characterized according to the approach adopted, which is supervised or unsupervised, and the features used. Supervised methods group features into normal and abnormal classes using trained classifier or probabilistic model. They train a classifier or probabilistic model using features of training data with normal class labels in the training phase. Then, trained classifier or probabilistic model is used to classify features as normal or abnormal. Unsupervised methods group features into clusters without using trained model, and they do not need labeled data. Unfortunately, existing literature has not elucidated the essential ingredients that make the methods work as they do, despite the fact that tests have been conducted to compare the performance of various methods. This thesis attempts to fill this knowledge gap by studying the videos tested by existing methods and identifying key components required by an effective unsupervised anomaly detection algorithm. Existing methods also tend to be very complex. Investigation into the test videos used by most of these methods suggests that they are overly complex, because speed or direction of moving objects seems possible for unsupervised anomaly detection. This thesis investigates the problem of unsupervised anomaly detection from first principle: analysis of the test videos to identify prominent characteristics. The investigation leads to a two-stage algorithm for unsupervised detection of anomaly based on speed or direction, and the dominant motion. Our comprehensive test results show that an unsupervised algorithm that captures the key components can be relatively simple and yet perform equally well or better compared to existing methods.

Contents

List of Publications

A. A. Abuolaim, W. K. Leow, J. Varadarajan, and N. Ahuja. On the Essence of Unsupervised Detection of Anomalous Motion in Surveillance Videos. In *Proc. Int. Conf. on Computer Analysis of Images and Patterns,* Aug 2017.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

In recent decades, surveillance cameras are widely used in public places. These cameras help the human observer to monitor public places and ensure the public safety. Human observers have the ability to detect anomalous motion from single surveillance scene with low density crowd [1]. However, they can get tired after monitoring for long hours and miss even the easy cases. Moreover, psycho-physical scientists indicate that monitoring multiple surveillance scenes is tedious, because most of the time nothing strange occurs in the scene [2]. Therefore, with the improvement of computer vision techniques, the research of anomaly detection in surveillance videos has caught further attention in the last few years. This research attracted many researchers to help the human observer in monitoring multiple surveillance scenes by detecting anomalous motion automatically using computerized methods. These methods can also be used for criminal investigation to sieve through video archives to detect anomalous activities that have happened in the past. This research also has many promising applications, such as intelligent surveillance [3], and safety evaluation [4]. Figure 1.1 shows a human observer monitoring multiple surveillance screens.

The meaning of anomalous motion pattern depends on the application context. This thesis focuses on surveillance videos of walking pedestrians (Fig. 1.2). In these videos, the normal human motion is the dominant motion (Fig. 1.2, 1st column). This normal motion happens when pedestrians are walking on the

Figure 1.1: Human observer monitoring multiple surveillance screens.

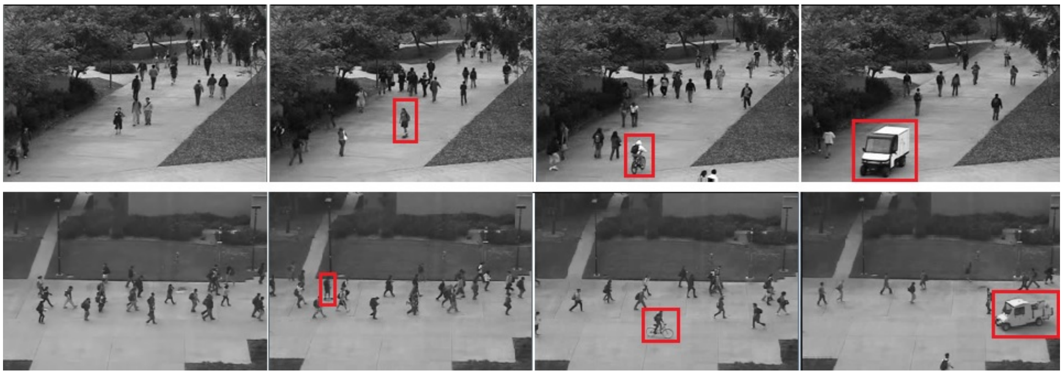

Figure 1.2: The first column shows normal motion when pedestrians are walking on the pedestrian walkway. The other columns show the abnormal motion (in red box) when skater, cyclist or cart cross pedestrian walkway respectively.

pedestrian walkway. The abnormal motion is a motion that does not conform to the dominant motion in a given surveillance scene, Figure 1.2 also shows anomalies (bounded by red boxes) in the 2$^{\text{nd}}$, 3$^{\text{rd}}$, and 4$^{\text{th}}$ columns, when skater, cyclist, or cart cross pedestrian walkway respectively.

Many methods have been proposed for anomaly detection with varying degree of accuracy. They can be characterized according to the approach adopted, which is supervised [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20] or unsupervised [21, 22, 23, 24, 25], and the features used, which range from low-level optical flow to high-level multiple object trajectories. In the first category, supervised methods group features into normal and abnormal classes using classifier or probabilistic models. They train a classifier or probabilistic model using features of training data with normal class labels in the training phase. Then, they classify new observation as normal or anomalous based on the trained model. These methods are usually accurate. However, the models of supervised methods need to be re-trained to include new normal class with new labeled data. In the second category, unsupervised methods group features into clusters without using trained models. They are different from supervised method because they do not train models and they do not need labeled data. They are also easy to expand to include new normal motion, but their accuracy is relatively lower than supervised methods.

Since the definition of an anomaly varies in different applications, designing a general framework for detecting anomalous motion patterns is still quit challenging. According to the nature of the problem definition of anomaly, anomalous motion will rarely happen. In reality, it can be any motion that does not conform to the dominant motion. Therefore, this definition will naturally lead to unbalanced normal/anomalous groups in which the normal group is much larger than anomalous group, hence, the most suitable and typical approaches for deriving a good solution are:

- Derive a supervised method that uses training data with only normal class labels for training. In testing, what else diverges from normal patterns will be anomalous.
- Derive an unsupervised method that groups the input data into two main groups: dominant (normal) and non-dominant (anomalous).

The above two approaches are commonly followed by existing methods; which is make sense; and in this thesis we will focus on them. In addition, anomalous motion is unpredictable, and there is a lack of specific real anomalous test videos, which introduces extra challenges to the methods who use anomalous labeled data for training.

Unfortunately, existing literature of the most common approaches has not elucidated the essential ingredients that make the methods work as they do, despite the fact that tests have been conducted to compare the performance of various methods. For example, test results (Chapter 4) seem to suggest that there is no significant advantage in offline training performed by supervised methods compared to well-crafted unsupervised methods. It is also uncertain whether the time taken to process high-level features necessarily leads to better detection accuracy. This situation makes it difficult to optimize the methods for real-time online detection and efficient video archive analysis.

1.2 Thesis Objective

This thesis attempts to fill this knowledge gap by studying the videos tested by existing methods and identifying key components (effective features) required by an effective unsupervised anomaly detection algorithm by proposing a two-stage algorithm based on speed or direction, and dominant motion. We have chosen to investigate unsupervised method instead of supervised method for the following reasons: (1) Unsupervised method does not require tedious and time-consuming manual labeling of training data. (2) It does not require an offline training phase. Therefore, it can be more easily extended to handle new normal and abnormal motion patterns that have not happened in the past. (3) Without the need of offline training, it can be more easily adapted to real-time online applications by implementing incremental algorithms. We focus on surveillance videos of pedestrians captured by stationary cameras because they are widely tested in the literature. Our comprehensive test results on these videos show that an unsupervised algorithm that captures the key components can be relatively simple and yet perform equally well or better compared to existing methods.

1.3 Thesis Organization

The remainder of this thesis is organized as follows: Chapter 2 discusses existing methods of anomaly detection and localization in surveillance video, including feature extraction and representation (Section 2.1), supervised methods (Section 2.2), and unsupervised methods (Section 2.3). A review of the test videos used in existing work (Section 2.4) and a summary of existing methods (Section 2.5) are also presented in Chapter 2. Then, Chapter 3 formulates the problem of anomaly detection (Section 3.1), develops the algorithm and presents the implementation details (Section 3.2). In Chapter 4, the experimental results, comparisons with state-of-the-art methods, and discussions are given. Finally, the conclusions are summarized in Chapter 5.

Chapter 2

Related Work

Anomaly detection and localization in surveillance video has been widely studied in the last decade. Existing methods can be organized into two main categories: supervised methods (Section 2.2) and unsupervised methods (Section 2.3). Regardless of the category, all methods have a feature extraction and representation stage. So, this common stage is reviewed first in Section 2.1. A review of test videos and a summary of all these methods will be discussed in Section 2.4 and Section 2.5 respectively.

2.1 Feature Extraction and Representation

The existing methods begin by extracting features from the input videos and then representing them to make detection decisions based on the feature representation. The feature representation can be subdivided into two main groups: hand-crafted and learned, where the hand-crafted feature representation include: trajectory of feature point and region-based representation.

Trajectory is a sequence of spatial locations (x, y) of a moving feature point, this feature point can be tracked over time. The trajectories of feature points can be obtained through tracked interest points [7, 11] or targets [23, 24]. Shandong Wu *et al.* [7] calculate optical flow, and then employ particle advection [26] to estimate the positions of moving particles using sub-pixel-level optical flow interpolation. Cui *et al.* [11] use the method of [27] to detect spatio-temporal interest points (STIP), and then track the STIP using KLT tracker [28, 29]. The tracked interest points can be grouped to represent the moving object, which helps in obtaining high-level information (e.g., speed, direction, etc.). However, it is difficult to track trajectory's point in extremely crowded scenes, due to dynamic background and scene clutter. Moreover, the trajectories tend to drift due to frequent occlusions.

Regarding pedestrian detection and multi-target tracking [23, 24], the objects of interest are detected first and marked as regions. These regions are tracked over time. The tracked region results in trajectory of region, and each region is tagged with a frame index and position for localization after anomaly detection. Yuan *et al.* [23] use the 3-D DCT model proposed in [30] to detect and track pedestrians. Lin *et al.* [24] employ the multiple hypothesis tracking algorithm proposed in [31] to track multiple pedestrians. This trajectory of pedestrian region facilitates the detection of abnormality at high-level semantics such as irregular long-term trajectory, speed and direction of object. However, the trajectory of region suffers from detection, segmentation, and tracking errors, and these errors dramatically increase in crowded or cluttered scenes. In addition, the trajectory of region is computationally expensive in terms of detection and tracking.

The region-based representation begins by dividing the video frames into regions, and then extract the features for each region to model or learn spatio and/or temporal motion patterns from image pixels [17, 21], 2D spatial regions [5, 6, 10, 12, 14, 16, 19, 20, 22, 23, 24] or 3D spatio-temporal regions [8, 9, 10, 13, 15, 18, 19, 25] of the video. The extracted features include optical flow [5, 12, 21, 22], histogram of optical flow (HOF) [6, 8, 18, 19, 20, 23, 24], histogram of oriented gradient (HOG) [8, 18], 3D SIFT [8, 25], histogram of edge orientation [20], descriptors of intensity, gradient, object persistence, motion direction, optical flow orientation, speed, etc. [10, 13, 16], structural descriptors based on HOF [23], particle advection based on optical flow [17], and dynamic textures [32] used in [9, 15]. Compared to the trajectory of feature point, region based representation was proposed to avoid tracking individual objects and to overcome tracking limitations. Moreover, simple features, such as optical flow and intensity gradient, take much less time to extract compared to features extracted by complex algorithms, such as pedestrian detection and multiple target tracking [23, 24]. However, these features need further processing to give high-level information. They are also not reliable in terms of object detection. Furthermore, this feature representation emphasizes dynamics in regions, ignoring anomalous object appearance. On the other hand, some methods have more complete representation that considers both appearance and motion. For example, [9, 15] introduce a mixtures of dynamic textures model to jointly utilize appearance and motion features, and [13] extracts both appearance and motion features over spatial neighborhoods.

The methods of [11, 17] characterize motion flow with interaction between crowd elements and introduce social models such as social force model. These methods have been inspired by the classical sociological study of crowd behavior in [33]. As for modeling crowd elements interactions, Cui *et al.* [11] propose an interaction energy potential to model the pedestrian interactions. After they extract the trajectories, each trajectory will be represented by interaction energy potentials. Then, standard bag-of-words method is used to represent each video clip. Mehran *et al.* [17] introduce social force model to analyze pedestrian dynamics. In their method, the extracted particles will be considered as individuals. The interaction forces of the particles are estimated using social force model. Then, the estimated interaction forces will be mapped into the frame to characterize Force Flow for each pixel. Even though the methods of [11, 17] model the interactions between crowd elements, their models mainly focused on motion information and need a prior knowledge for specific scenarios.

Si Wu *et al.* [6] represent the position, speed, and direction of the spatial foreground 2D regions with a three probability density function (pdf). Shandong Wu *et al.* [7] draw inspiration from the mathematical theory of chaotic systems to model and analyze nonlinear dynamics of trajectories used in [34]. To characterize a chaotic system, they calculate two chaotic invariants, namely the largest Lyapunov exponent and the correlation dimension. Cheng *et al.* [8] construct code-book using bottom-up greedy clustering of the extracted descriptors. Cong *et al.* [19, 20] extract Histogram of Optical Flow (HOF), then employ sparsity consistency to obtain an optimal subset of non-redundant features free of noise. This optimal subset is considered as training dictionary. By extracting Multi-scale Histogram of Optical Flow (MHOF), the method of [20] improves the detection performance compared to [19]. Duan-Yu Chen *et al.* [22] quantize optical flow

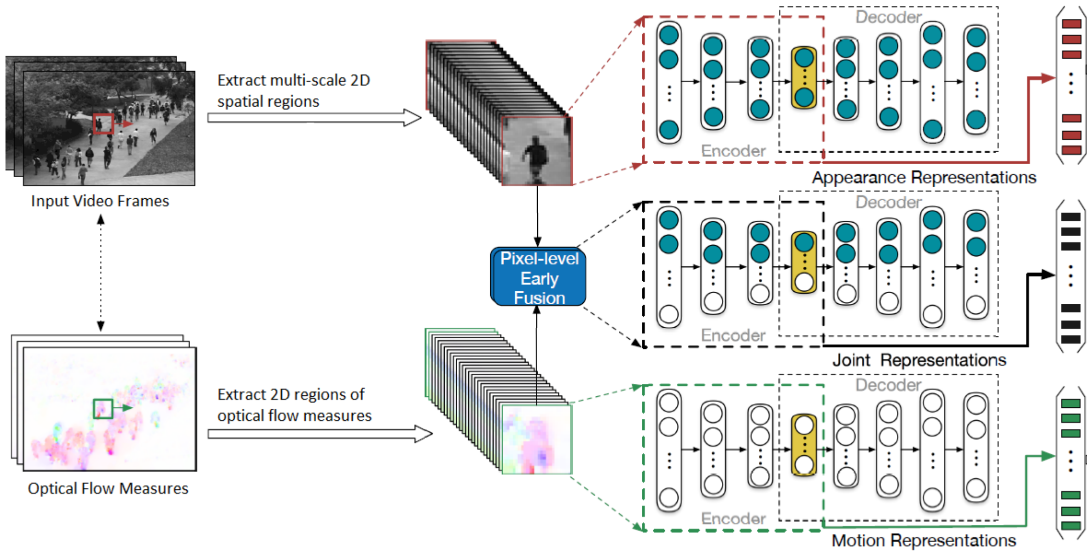

Figure 2.1: The structure of the stacked denoising auto-encoders (SDAE) proposed by Xu *et al.* [5]. The multi-scale 2D spatial regions will be warped into equal size.

orientations in 2D regions. Lin *et al.* [24] detect the foreground at multiple scales using the method of [35] based on the GMM modeling. They next detect Region Of Interest (ROI) of the foreground by scanning the input video using a 2D sliding window. In this way, most of the background will be filtered out, which reduces the computational cost and suppresses the effect of noise. For each ROI in the foreground the optical flow of each pixel is calculated using Horn and Schunck's method [36]. Javan *et al.* [25] adopt a probability density function (pdf) to model the densely sampled 3D saptio-temporal regions of HOG features. In their method, the high-dimensional pdf need to be approximated, and it suffers from the problem of curse of dimensionality. Yuan *et al.* [23] use the 3-D DCT model proposed in [30] to detect and track pedestrians. Each detected pedestrian will be represented in a 2D bounding box. They also propose a structural descriptor based on HOF. The proposed descriptor will be extracted for each bounding box.

The common shortcoming of the previously mentioned hand-crafted features is that they need a prior knowledge to design an effective representation, which is time-consuming and difficult. Nowadays, deep learning has become a hot topic, in which the researchers employ deep learning methods to learn features automatically from input raw data. Deep learning methods have been successfully used in many computer vision tasks, such as object detection [37], image classification [38] and activity recognition [39]. The key reason behind using deep learning methods is that discriminative and meaningful features can be adaptively learned through multi-layer nonlinear transformations. Thus, it makes sense that detecting anomalous motion patterns in videos can also benefit from deep learning features.

An autoencoder is a feedforward neural network used for unsupervised feature learning. It has an input layer, an output layer and at least one hidden layer in between. The output layer of an autoencoder must has the same number of nodes as the input layer. The aim of an autoencoder is to reconstruct its own inputs. Thus, auto-encoders are unsupervised learning models. Auto-encoders have been used to extract features from video in [5, 14, 18]. Xu *et al.* [5] propose a novel approach based on stacked denoising auto-encoder (SDAE) [40] to learn features of both appearance and motion patterns in an unsupervised way. They introduce a three pipelines of SDAE as shown in Figure 2.1. The inputs are: multi-scale 2D spatial regions of original frame (to capture appearance), 2D regions of optical flow measures (to capture motion), and a joint vector that combines 2D spatial regions with their corresponding optical flow measures. The number of nodes of the first layer of the 2D spatial regions and 2D optical flow regions is both set to 1024 (dimension of the input vector), while the first layer of the joint vector pipeline is set to 2048. Therefore, the encoder structure can be defined as: 1024(2048)⇒ 512(1024)⇒ 256(512)⇒ 128(256), and the decoder is a symmetric structure. Based on [40], the output of any layer in a SDAE can be used as learned feature representation. In their structure, they choose the output vector of the last hidden layer in the encoder part because it is the smallest feature vector that gives a more compact feature representation. Similarly, Mohammad *et al.* [18] use the architecture of denoising auto-encoders proposed in [41] to learn input features, but the input of their method is the 3D spatio-temporal regions. Despite the fact that the proposed novel unsupervised feature learning methods [5, 18] extract effective and compact feature representation, their methods only consider short-term temporal motion, i.e., optical flow measures (only two consecutive frames). Feng *et al.* [14] use also the SDAE proposed in [5], and their method learns long-term temporal motion using recurrent neural network, which performs non-linear transformation and considers both the current input state and the previous hidden state. More specifically, they adopt long short-term memory (LSTM) framework [42] because it is capable of learning long-term temporal motion dependencies. For all methods that use auto-encoder, the stochastic gradient descent (SGD) is used to optimize and learn them. Auto-encoders can extract and learn features efficiently without need a prior knowledge, where they can be easily generalized for different scenarios. However, auto-encoders take a large amount of time to train. They also have fuzzy design decisions (e.g., number of nodes, layers, learning parameters, etc.) with a lack of theoretical-based justification.

2.2 Supervised Methods

Supervised methods classify features into normal and abnormal classes using trained models. They typically work in two phases: training phase and testing phase. In the training phase, these methods use class classifiers to learn a model of the labeled normal training data. In the testing phase, they determine whether new testing data belong to the normal class. The training models include probabilistic model [6, 7, 8, 9, 12, 15, 16, 17, 18], dictionary [10, 13, 19, 20], and classifier [5, 11, 14]. Unlike supervised methods, Saligrama *et al.* [13] use k-nn to find anomalies, where their method does not need the training phase.

Methods of [6, 7, 8, 9, 12, 15, 16, 17, 18] use the samples with only normal class label to train a probabilistic model in the training phase. They infer the likelihood of a test sample with respect to a trained probabilistic model, where the test sample with low-probability will be considered as anomaly in the testing phase. Among the methods that train a probabilistic model, Si Wu *et al.* [6] train probability density functions (pdfs) using a probabilistic conjugate Bayesian analysis. Shandong Wu *et al.* [7] train the Gaussian mixture models (GMMs) using the chaotic feature set to describe the probability density function of the normal motion patterns. The GMMs are trained using expectation maximization (EM). Cheng *et al.* [8] adopt a Gaussian process regression (GPR) model. Li and Mahadevan *et al.* [9, 15] detect temporal anomaly using the popular background subtraction method in [43]. The method of [43] employs GMM at each 2D region for modeling the local distribution of region intensities. In [9, 15] the GMM is replaced by Mixture of Dynamic Textures (MDT) [32]. The MDT of spatio-temporal 3D regions is learned using EM in the training phase.

To detect spatial anomaly, Li and Mahadevan *et al.* [9, 15] also use the discriminant saliency criteria of [44], where the anomalous spatial 2D regions are those whose saliency measures above a pre-defined threshold. The method of [9] apply a Conditional Random Field (CRF) filter on multi-scale image regions, which significantly improves the detection performance compared to [15]. Kim *et al.* [16] model the descriptors of 2D spatial regions with a mixture of probabilistic principal component analysis (MPPCA) models. Then, they adopt Markov Random Field (MRF) model, where the nodes in the MRF graph correspond to a 2D spatial regions in the video frames, and neighboring nodes are associated with links. Finally, the trained MPPCA model and MRF graph are used to compute maximum a posterior estimate of new observation. Adam *et al.* [12] and Sabokrou *et al.* [18] model the extracted feature representation with simple probability distributions such as Gaussian distributions, where they have a simple training phase that estimates only the distribution parameters (i.e., mean and variance). Mehran *et al.* [17] select randomly a spatio-temporal volumes of Force Flow, they model normal motion patterns in the video using a probabilistic graphical model called Latent Dirichlet Allocation (LDA) [45] and they use EM to train LDA model.

There is another group of methods [10, 13, 19, 20] that construct a dictionary using only the normal training samples in the training phase. In the testing phase, these methods use the reconstruction error of a new observation as a metric for anomaly detection. Boiman *et al.* [10] introduce an inference by composition method to compute the joint probability between a training dictionary and a new testing sample. Bayesian network propagation is used to compute the joint probability. They consider new testing sample as anomalous if it cannot be reconstructed from training dictionary. Cong *et al.* [19, 20] consider an optimal subset of feature representation as training dictionary. In their method, each testing sample could be a sparse linear combination from the training dictionary using weighted l_1 minimization. They determine whether testing sample is normal or not based on its linear reconstruction cost.

Saligrama *et al.* [13] propose a supervised method that does not need the training phase. They first compute the K-nearest neighbor (K-NN) for each 3D spatio-temporal region based on Euclidean distance. Then, they aggregate weighted K-NN distances from all regions to compute normalized composite score. This composite score will be ranked with respect to other such composite scores associated with training normal samples. They finally declare anomalies as low scores against offline templates.

Regarding the methods that train a classifier [5, 11, 14], Cui *et al.* [11] and Xu *et al.* [5] adopt support vector machine (SVM) classifier to find the abnormal motion patterns. The method of [5] is a bit different, it uses one-class SVM [46] for each of three types of learned feature representations (three pipelines). They train one-class SVM of radial basis function (RBF) kernel using only normal samples. Then, the three anomaly scores of the three one-class SVMs are computed and combined using unsupervised late fusion scheme. Finally, they consider a test sample as anomalous if its score below a pre-defined threshold. As mentioned previously, even though Xu *et al.* [5] extract effective and compact feature representation, their methods only consider short-term temporal motion. Therefore, Feng *et al.* [14] adopt LSTM model to learn long-term temporal motion dependencies by taking both the current input state and the previous hidden state as inputs to predict time dependencies. The test sample that disobeys the predicted dependency is considered as anomaly.

The methods that train a probabilistic model are based on a firm theoretical foundation, they are also theoretically justified to get optimal solution. However, some models introduce more parameters, which increases the complexity. They also suffer from the curse of dimensionality. Regarding the methods that construct a dictionary, even though they are based on a firm theoretical foundation, these methods need to exhaustively sample 2D/3D regions from the video, which leads to higher computational cost. Moreover, these methods cannot be adopted or applied directly in an online manner, because they require the entire dictionary to be constructed beforehand so that these methods can proceed. As for the methods that train classifiers, the training of SVM is relatively easy because SVM is defined by a convex optimisation problem (no local minima). SVM also can scale relatively well to high-dimensional data. However, it is reasoning to manually choose the kernel of SVM. On the other hand, deep learning methods [5, 14, 18] can learn discriminative features automatically. However, deep learning methods are hard to train and there is no significant advantage of deep learning methods compared to other supervised methods based on our test results (Chapter 4). In General, well-trained supervised methods can be accurate. Moreover, their testing phases are typically efficient enough for real-time applications, provided the features can be extracted efficiently. However, this approach may suffer from a high false positive rate, since any normal example not included in the training data will be detected as anomaly. Therefore, it is difficult to extend these methods to include new normal scenario.

2.3 Unsupervised Methods

Unsupervised methods [21, 22, 23, 24, 25] typically group extracted features into clusters without training a model and without relying on labeled data. The clustering algorithms that have been used include hierarchical cluster merging [22], k-means [24], online weighted clustering [24], and fuzzy probabilistic clustering [25]. After clustering, these methods label dominant clusters (i.e., clusters with the most members) as normal and the other clusters as abnormal. The threshold for deciding which clusters are dominant is empirically set. The methods of [21] and [23], on the other hand, do not perform clustering. Instead, the method of [21] detects high speed motion and performs line intersection to detect the center of crowd dispersion, and the method of [23] measures dissimilarity between features to detect anomalies.

Duan-Yu Chen *et al.* [22] apply hierarchical cluster merging. The similarity measure is used to calculate distance between feature points. Hierarchical clustering is easy to implement and does not need a prior information about the number of clusters required. However, in hierarchical clustering algorithm, there is no objective function need to be minimized and the time complexity is $O(n^2 \log n)$, which makes the clustering of feature points of large video computationally expensive. Lin *et al.* [24] quantize the flow vectors within Region Of Interest (ROI) using k-means clustering to obtain the Adaptive Multi-scale Histogram Optical

Flow (AMHOF) features. Compared to hierarchical clustering, k-means has a linear time complexity, but k-means requires the number of clusters to be initialized (empirically set). Each ROI is characterized by AMHOF and spatial location. An online weighted clustering of ROIs is performed to obtain abnormal clusters. They adopt weighted clustering to overcome the perspective distortion. To improve the detection performance, they also apply a simplified Multi-Target Tracker (MTT) algorithm [31]. The method of Lin *et al.* [24] is able to catch slow changes of normal motion patterns in an online adaptive manner. However, their method cannot be optimized to be real-time, because they use MTT algorithm with two clustering methods. Javan *et al.* [25] perform a fuzzy probabilistic clustering to obtain the abnormal clusters. The resultant clusters of the fuzzy probabilistic clustering can be characterized by a small number of parameters. However, fuzzy probabilistic clustering usually converges to local minimum.

Chun-Yu Chen and Yu Shao [21] compute the weighted speed of pedestrians based on optical flow to detect pedestrian escape motion pattern using an empirically set threshold. They also introduce the divergent centers analysis to detect the center of crowd dispersion by intersecting the paths of escaping pedestrians. Their method is very simple and fast. However, their method is only able to detect high speed motion and will fail in detecting other anomalies (e.g., direction, appearance, and interaction). Yuan *et al.* [23] propose a measure of dissimilarity between descriptors to detect anomalies. The method of [23] can perform in an online manner. However, it cannot be optimized to be real-time. Moreover, The method of [23] is mainly based on the 3-D DCT tracking method of [30], which means the detection of anomalies will be mainly related to robust detection and tracking of pedestrians (tracking failure leads to detection failure).

Compared to supervised methods, unsupervised methods do not require manually labeled training data, do not have separate training phase and testing phase, and do not perform offline training. Unsupervised methods can be easily extended to handle new normal/abnormal motion patterns, because they do not need to retrain a model and do not need new labeled data. Moreover, unsupervised methods that use incremental algorithms are very suitable for real-time online applications.

Table 2.1: The test videos used by various papers.

Reference	Datasets									
	Surveillance								Non-Surveillance	
	Pedestrian						Traffic		Human	
	UCSDped1	UCSDped2	UMN	Subway	PETS2009	BEHAVE	U-turn	QMUL	Web	[25, 47]
[5, 14]	✓									
[6, 7, 22]			✓		✓					
[8]	✓			✓				✓		
[9, 19]	✓	✓	✓	✓			✓			
[10]	✓			✓						
[11]			✓			✓				
[12, 16]	✓	✓		✓						
[13]	✓		✓	✓			✓			
[15, 20, 24]	✓	✓								
[17]	✓	✓	✓		✓				✓	
[18]		✓	✓							
[21]			✓							
[23]	✓	✓	✓							
[25]	✓	✓		✓						✓

2.4 Review of Test Videos

This section reviews test videos used by existing methods of common approaches (Table 2.1). These test videos have surveillance and non-surveillance videos. The surveillance videos are divided into: pedestrian videos and traffic videos. On the other hand, the non-surveillance videos are only human videos.

As shown in Table 2.1, there are many datasets tested by existing work. Some of them are commonly used including UCSDped1, UCSDped2, UMN, Subway, and PETS2009. The rest are not commonly used including BEHAVE, U-turn, QMUL, Web, and the dataset in [47, 25].

This thesis focuses on surveillance videos of pedestrians. Therefore, this section analyzes in details the surveillance videos with pedestrian activities. These datasets include UCSDped1, UCSDped2, UMN, PETS2009, Subway, and BEHAVE. Each surveillance video in these datasets is recorded using stationary camera.

The UCSD dataset [48] is divided into 50 training videos and 48 testing videos. Each video has a length ranging from 120 to 200 frames. For the training video of UCSD, it contains only pedestrians walking in different directions at pedestrian walkway, and this is considered as normal motion. For the testing video of UCSD,

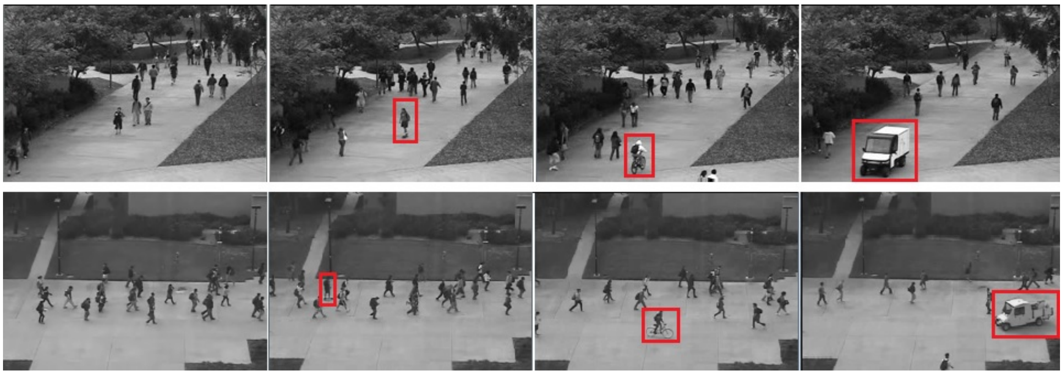

Figure 2.2: UCSD dataset. the first row is ped1 scene and the second row is ped2 scene. The first column shows normal motion when pedestrians are walking on the pedestrian walkway. The other columns show the abnormal motion (in red box) when skater, cyclist or cart cross pedestrian walkway respectively.

it contains mostly pedestrian walking in different directions at pedestrian walkway, but occasionally some carts, cyclists, or skaters cross the pedestrian walkway at higher speed compare to pedestrians, and this is considered as abnormal motion (because they pose hazard to the pedestrians). The training and testing videos are not staged or synthesized and all of the anomalous motions naturally occur. Two of test videos contain a wheelchair, and the wheelchair moves at the same speed as pedestrian walking. In these two test videos, the wheelchair is labeled as abnormal, but this is actually ambiguous whether should be normal or abnormal, because these are also pedestrian but on wheelchair; unlike other anomalies they do not pose hazard to the pedestrians.

The videos of UCSD are captured in two different scenes called “ped1” and “ped2”, as shown in Figure 2.2. The first scene, denoted “ped1”, contains 34 training videos and 36 testing videos. It has videos with 158x238 resolution and the pedestrians walking vertically with respect to the camera. The second scene, denoted “ped2” contains 16 training videos and 12 testing videos. It has videos with 240x360 resolution and the pedestrians walking horizontally with respect to the camera.

The testing videos of UCSD show that all anomalous entities such as skaters, cyclists and carts are moving at higher speed than pedestrians. For UCSDped2 “Test002” and “Test004” videos, the speeds of moving objects are computed by computing the speeds of distinctive feature points associated to the moving objects. To provide more detailed analysis, the speeds of distinctive feature points over the whole video are plotted in two graphs for UCSDped2 “Test002” and “Test004” videos as shown in Figure 2.3. In each graph, there is a dominant group (green box) of feature points with similar speed. This is regarded as the normal group. There are also smaller dominant groups (red boxes) of feature points with similar speed, and they correspond to abnormal motion (e.g., skaters, cyclists and carts). These are the abnormal groups with higher speed. The rest of the feature points (do not belong to any of dominant groups) are considered as ambiguous points. Obviously, the speeds of normal and abnormal groups are very different. In the same Figure 2.3, the point (a) is more likely to be normal, the point (c) is more likely to be abnormal, and the point (b) in the middle is very ambiguous. Additionally, the feature points correspond to abnormal groups are visualized in Figure 2.4 for both test videos “Test002” and “Test004”. The “Test002” has the second dominant group (red box), the feature points of this group are visualized and cover cyclist crosses the pedestrian walkway as illustrated in Figure 2.4 (column 1). Regarding to the “Test004”, it has the second and third dominant groups (red boxes), the feature points of these groups are visualized, where they cover cart (second dominant group) and cyclist (third dominant group) cross the pedestrian walkway as illustrated in Figure 2.4 (column 2). On the other hand, UCSDped1 testing videos contains only 2 videos (Test021 and Test023) where the anomalous entities move in a speed similar to walking pedestrians speed. For example, in UCSDped1 “Test021” video, wheelchair crosses the pedestrian walkway in a speed similar to pedestrian walking speed. Figure 2.5 plot of the speeds of distinctive feature points over video frames for UCSDped1 “Test021”. After looking at Figure 2.5, it has only the first dominant group.

The UMN dataset [49] contains one video. This video has 7710 frames. It is captured in three different scenes as shown in Figure 2.6. Each scene has a different length. The video of UMN contains pedestrians walking in different directions, and this is considered as normal motion (6280 frames). Suddenly, the pedestrians start escaping in panic at higher speed, and this is considered as abnormal motion (1430 frames). However, this video is staged, synthesized, and so artificial. It also produces a very large difference in speed when pedestrians start running in the

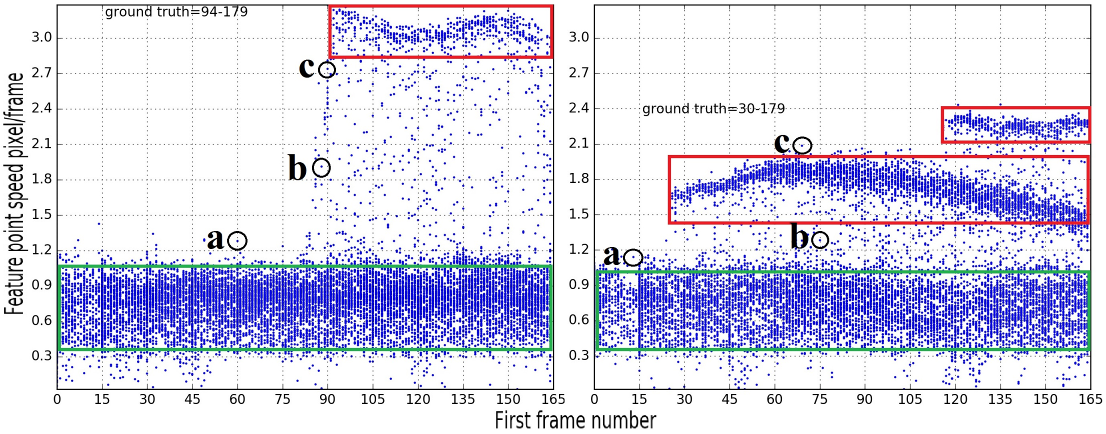

Figure 2.3: The speeds of distinctive feature points over video frames for UCSDped2 "Test002" and "Test004" videos. The x-axis represents the first frame where tracking of the feature point strats.

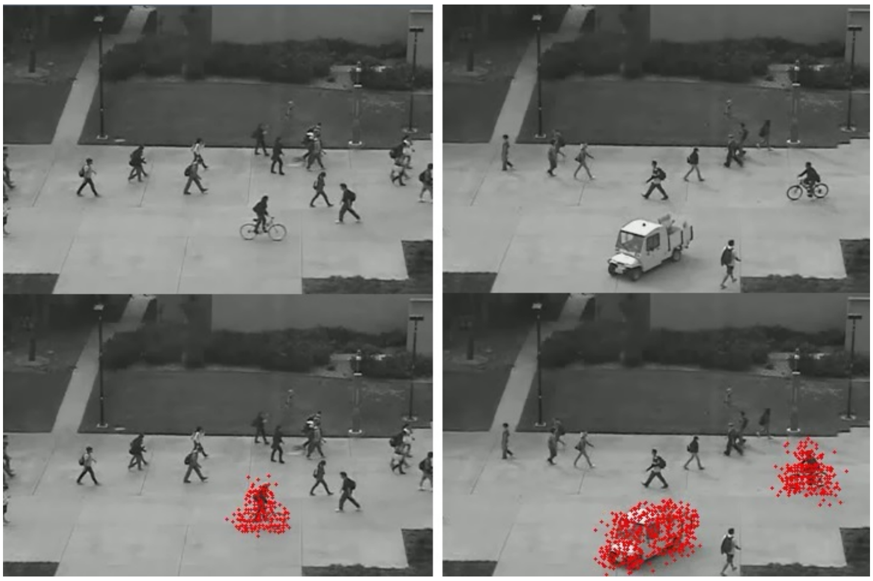

Figure 2.4: The first column is the visualization of feature points (red points) correspond to abnormal group in UCSDped2 "Test002" (at frame 160). The second column is the visualization of feature points (red points) correspond to two abnormal groups in UCSDped2 "Test004" (at frame 170).

abnormal case.

For further analysis; Figure 2.7 is plot of the speeds of distinctive feature points over video frames for UMN 1^{st} scene and 3^{rd} scene. It is clear in Figure 2.7 there

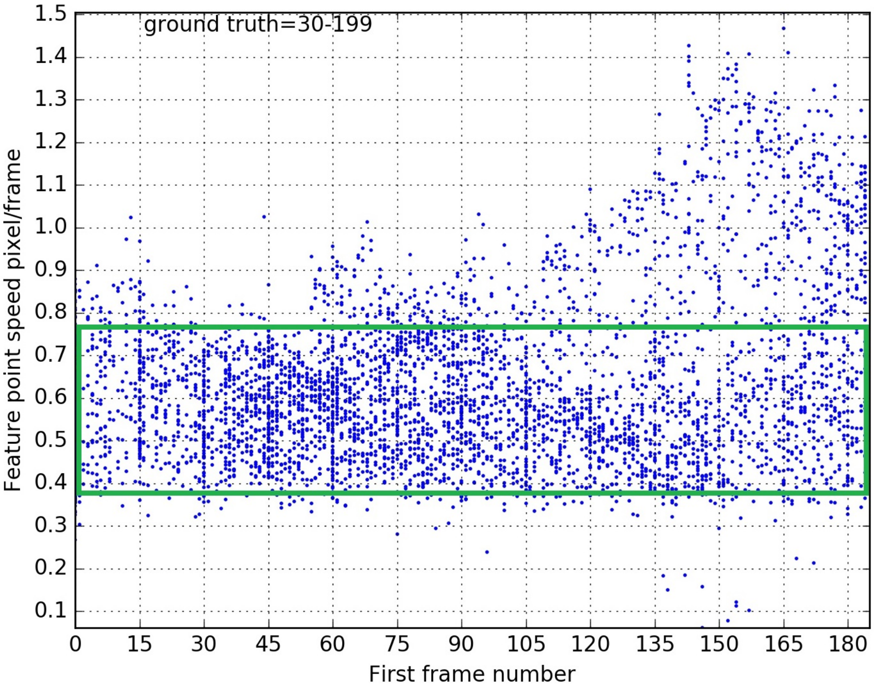

Figure 2.5: The speeds of distinctive feature points over video frames for UCSDped1 “Test021” video.

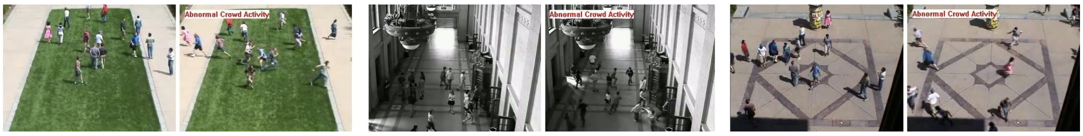

Figure 2.6: UMN dataset. Three different scenes with normal/abnormal scenarios.

is a large dominant group (green box) of feature points with similar speed. This is regarded as the normal group. There are also smaller dominant groups (red boxes) of feature points with similar speed, and they correspond to abnormal motion. The speed is very different between normal and abnormal motion as illustrated in Figure 2.7.

The PETS2009 dataset [50] contains two anomaly scenarios. For each scenario, there are 4 videos captured in 4 different views by different cameras as shown in Figure 2.8. Each view has 576x768 frame resolution. The first scenario has 223 frames, and the second scenario has 378 frames.

For the first scenario of PETS2009, it contains a group of pedestrians walking at pedestrian walkway, and this is considered as normal motion; abnormal mo-

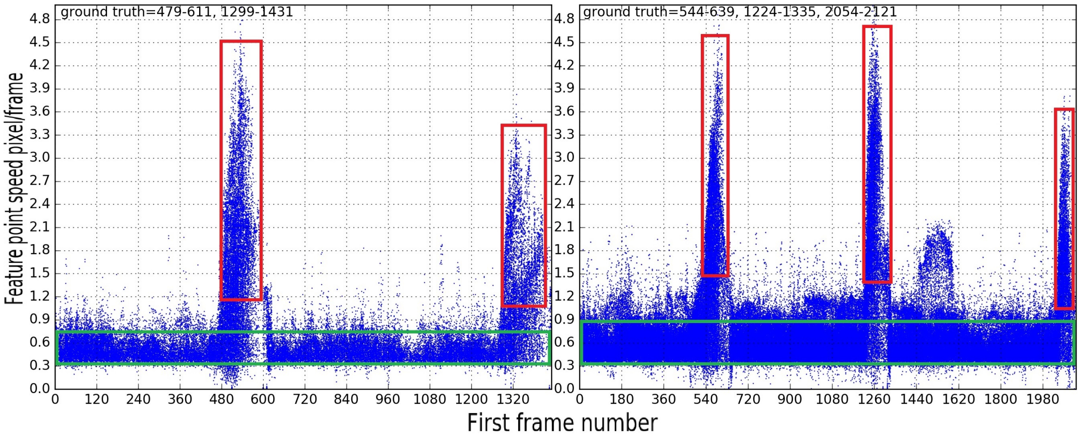

Figure 2.7: The speeds of distinctive feature points over video frames for UMN 1st and 3rd scenes.

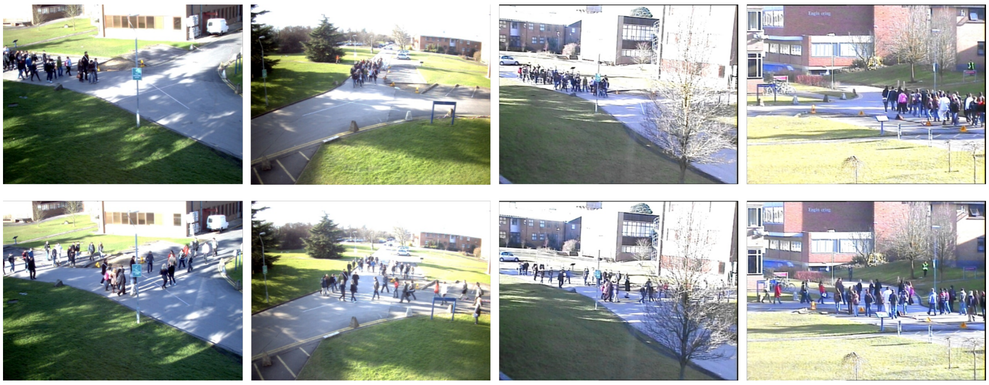

Figure 2.8: PETS2009 dataset. Each row is a scenario and each column is a view.

tion happens when pedestrians suddenly start running in one direction. For the second scenario of PETS2009, it contains three groups of pedestrians walking to the center of pedestrian walkway, then they merge at the center, and this is considered as normal motion; abnormal motion happens when pedestrians suddenly start running in different directions. However, the two scenarios have significant differences in the field of view and illumination for the four views. Moreover, the test videos of PETS2009 are staged, synthesized, and so artificial. It also produces a very large difference in speed when pedestrians start running in the abnormal case.

For more detailed analysis; Figure 2.9 is plot of the speeds of distinctive feature points over video frames for the PETS2009 1st and 2nd scenarios (first view). It is clear in Figure 2.9, there is a large dominant group (green box) of feature points

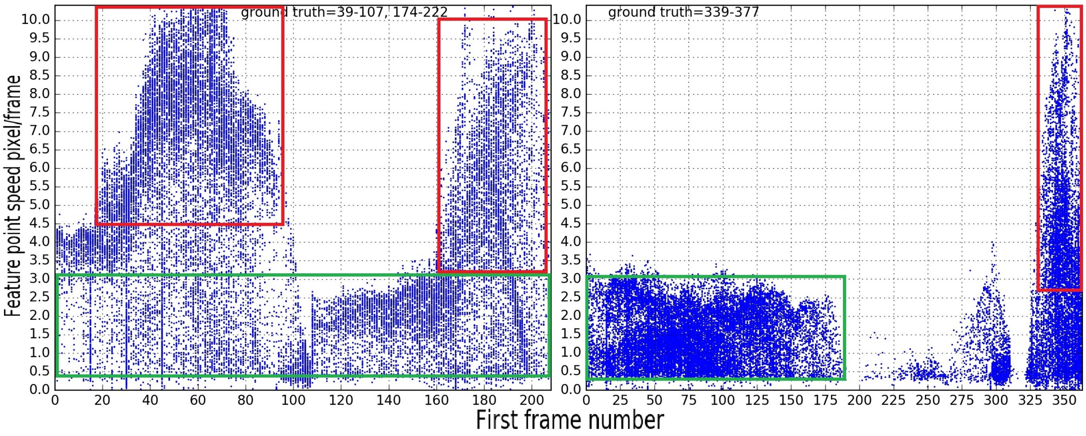

Figure 2.9: The speeds of distinctive feature points over video frames for PETS2009 1st and 2nd scenarios. These scenarios are taken from the first view.

with similar speed. This is regarded as the normal group. There are also groups (red boxes) with higher speed, and they correspond to abnormal motion.

The Subway dataset [12] contains two videos. These videos are recorded from the entrance (96 min, around 145k frames) and exit (43 min, around 65k frames) of a subway station with 384x512 frame resolution.

The videos of Subway contain passengers entering and exiting the station from the entrance and exit respectively, and this is considered as normal motion; abnormal motion happens when passengers are exiting from the entrance or entering from the exit as shown in Figure 2.10. These test videos have only two anomalous motion patterns, and predictable spatial localization of anomaly (at entrance and exit regions). Speed alone is not enough to detect anomalies for these test videos. However, direction alone would further improve the detection of anomalies.

The BEHAVE dataset [51] contains 4 videos with 480x640 frame resolution. These videos contain pedestrian group activities, including meeting, splitting up, standing, walking, and ignoring each other; these activities are considered as normal motion. Abnormal motion starts when fighting or escaping happens as shown in Figure 2.11. However, these videos are staged, synthesized, and so artificial. They are also speed dependent and produce a very large difference in speed when pedestrians start fighting or running in the abnormal case.

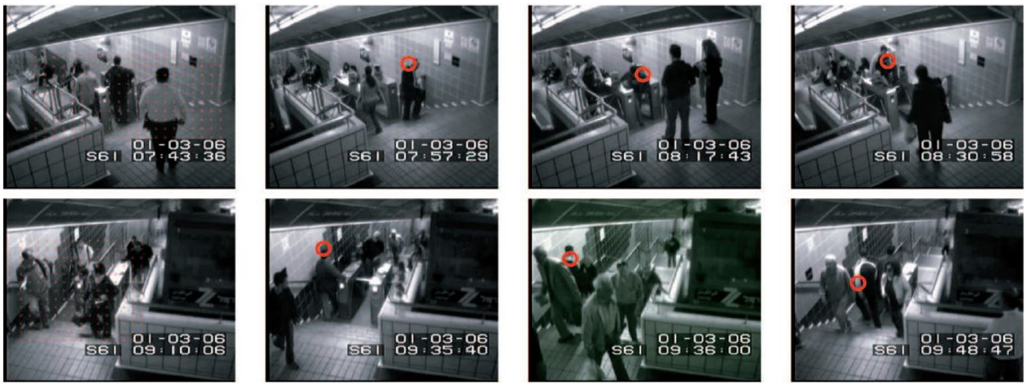

Figure 2.10: Subway dataset. The first row is the entrance scene and the second is the exit scene. The first column is the normal motion pattern and the others are the possible anomalies.

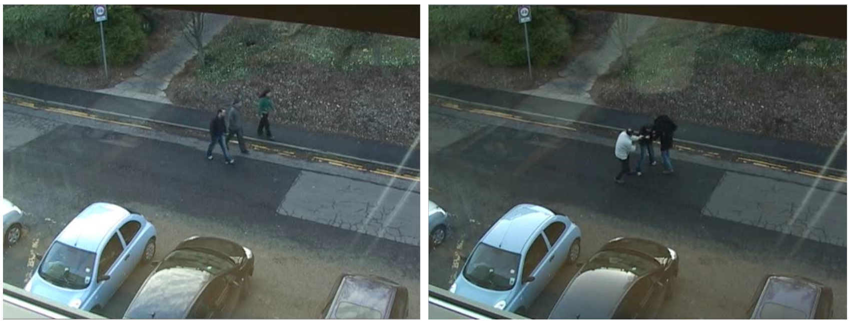

Figure 2.11: BEHAVE dataset. The first picture shows normal motion, where pedstrians walking. The second picture shows abnormal motion, where the fighting happens.

2.5 Summary

Supervised methods in general need to re-train a classifier or probabilistic model to include new normal scenario, and they need new labels. Unsupervised methods do not train a classifier or probabilistic model beforehand, and they do not need labels; which makes them easy to expand. Supervised methods follow the common approach and train models from only normal samples.

By reviewing the surveillance test videos of pedestrians, most of them are speed dependent (i.e., UCSDped1, UCSDped2, UMN, PETS2009, and BEHAVE). The Subway dataset is not speed dependent, but it is direction dependent. The BEHAVE dataset is not commonly used (only used by one paper [11]); anyway the escape and fighting anomalies in BEHAVE are about speed, and their scenarios are already included in other test videos. Therefore, this thesis proposes a two-stage algorithm for unsupervised detection of anomaly based on speed or direction, and the dominant motion.

Chapter 3

Unsupervised Anomaly Detection

3.1 Problem Formulation

In the common pedestrian test videos analyzed in Section 2.4, the pedestrians are the normal moving entities. Moreover, they are the dominant moving entities and they move at roughly the same speed or direction. Therefore, the dominant motion can be regarded as normal motion. On the other hand, abnormal moving entities such as carts, cyclists, skaters, escaping humans move at a higher speed or opposite direction. They are not the dominant motion in the whole video.

The essence of unsupervised method is to group feature points into non-overlapping clusters that each contains consistent members. Therefore, unsupervised detection and localization of anomalous motion can be decomposed into two sub-problems: (1) grouping of feature points into clusters, and (2) labeling of clusters.

The first sub-problem is formulated as follows: Given either motion speed or motion direction of n feature points f_i, $i = 1, \ldots, n$, in video, group f_i into m non-overlapping clusters $\mathcal{C}_j$, $j = 1, \ldots, m$. Each cluster $\mathcal{C}_j$ is characterized by the cluster size $\mid \mathcal{C}_j \mid$ and the cluster center, which is the average feature value $\bar{f}_j$ of the features in $\mathcal{C}_j$. The grouping should satisfy the following constraints:

1. Small intra-cluster difference $d_v(\mathcal{C}_j)$.

$$d_v(\mathcal{C}_j) = \frac{1}{\mid \mathcal{C}_j \mid} \sum_{f \in \mathcal{C}_j} (f - \bar{f}_j)^2. \tag{3.1}$$

2. Large inter-cluster difference $d_x(\mathcal{C}_j, \mathcal{C}_k)$.

$$d_x(\mathcal{C}_j, \mathcal{C}_k) = (\bar{f}_j - \bar{f}_k)^2. \tag{3.2}$$

3. Well-separated clusters.

$$d_v(\mathcal{C}_j) < d_x(\mathcal{C}_j, \mathcal{C}_k),\ \forall\, \mathcal{C}_j,\ \textit{and all}\ \mathcal{C}_k \neq \mathcal{C}_j. \tag{3.3}$$

The second sub-problem is formulated as follows: Given clusters $\mathcal{C}_j$, $j = 1, \ldots, m$, label each cluster as either normal, abnormal or ambiguous according to the following conditions:

1. Clusters with the largest sizes are normal.

2. Clusters with the highest speeds are abnormal.

3. Clusters not labeled as normal or abnormal are ambiguous.

A video frame that contains any of anomalous feature points is regarded as abnormal; otherwise, it is normal. In the abnormal frame, abnormal moving entities are localized based on positions and frame indices of the anomalous feature points.

3.2 Proposed Algorithm

The goal of this thesis is to identify the essential ingredients for effective unsupervised detection of anomalies in pedestrian surveillance videos. To achieve this goal, we apply the principle of Occam's razor: given several equally effective alternatives, we choose the simplest alternative. Therefore, we call our method OCCAM. Similar to unsupervised methods based on clustering, OCCAM consists of four main stages:

1. Extract and track distinctive feature points.

2. Group feature points into clusters based on speed.

3. Label clusters based on speed and size.

4. Finally, anomalous motions are detected and localized in the videos.

Stage 1: Feature Extraction and Tracking

Analysis of common test videos used in existing work (Section 2.4) shows that normal and abnormal motion may be differentiated by either motion speed or motion direction alone, depending on the test videos. Therefore, OCCAM uses motion speed or motion direction as the feature. OCCAM extracts and tracks distinctive feature points using the dense trajectory features proposed by Wang *et al.* [52, 53]. Their method densely samples feature points at multiple spatial scales. It also tracks feature points using median filtering in a dense optical flow field [54]. Their method tracks the feature points for 15 frames (to avoid drifting) and sample new feature points to replace them. Additionally, their method removes the static feature points, and removes feature points with large displacements between two consecutive frames to reduce errors.

Each feature point p_i, $i = 1, \ldots, n$, has a sequence of spatial locations over time $\{\mathbf{x}_i^t, \ldots, \mathbf{x}_i^{t+l}\}$, where $\mathbf{x}_i^t$ is a position $(x,\ y)$ of point i at frame t, and l is the trajectory length. The speed s_i and direction θ_i of feature point p_i are calculated by:

$$s_i = \frac{\|\mathbf{x}_i^{t+l} - \mathbf{x}_i^t\|}{l}. \tag{3.4}$$

$$\theta_i = \left(\arctan\left(\frac{y_i^{t+l} - y_i^t}{x_i^{t+l} - x_i^t}\right) \times \frac{180^\circ}{\pi} \right) \bmod 360^\circ \tag{3.5}$$

In UCSDped1 test videos, objects and humans move toward and away from the camera, and there is a noticeable amount of perspective distortion. This distortion results in motion parallax, where objects that are closer to camera move faster than objects that are further away from camera. This perspective distortion does not give the actual speed of feature points, which directly affects the accuracy of the proposed method on UCSDped1. Therefore, to overcome this distortion, the feature points of UCSDped1 are projected into the ground plane using an estimated Homography. The speed of feature point is calculated after the projection, to give the actual speed for this special case. Section 4.6 discusses the direct effect of projected points on detection accuracy.

Stage 2: Feature Clustering

Feature clustering is performed on either motion speed or motion direction. Let us denote the extracted feature values as f_i, $i = 1, \ldots, n$. Since the features are 1-D, the simplest way to cluster f_i is to divide the feature value range (minimum to maximum) into m equal intervals, and regard each interval as a cluster C_j, $j = 1, \ldots, m$. Then, features f_i can be clustered efficiently into their respective clusters in a fixed $O(n)$ time. Each cluster C_j is characterized by the cluster size $|C_j|$ and the cluster center, which is the average feature value $\bar{f}_j$ of the features in C_j. This simple and efficient clustering method ensures that the intra-cluster differences are much smaller than the inter-cluster differences, which satisfies the constraint of well-separated clusters (will be explained in an example later Fig. 3.2).

After clustering, normalized cluster size S_j and normalized cluster center F_j are computed for each cluster C_j. Let us denote the dominant cluster, the cluster with the largest size, as C^+ and the largest feature value as f^*. Then, S_j and F_j are computed as follows:

$$S_j = |C_j|/|C^+|, \quad F_j = \bar{f}_j/f^*. \tag{3.6}$$

Therefore, these normalized values range between 0 and 1. Each cluster C_j is now characterized by a characteristic vector of two components, namely normalized cluster size S_j and normalized cluster center F_j.

Stage 3: Cluster Labeling

Unlike existing methods, OCCAM labels the clusters into three types: normal, abnormal, and ambiguous. The ambiguous clusters allow the normal and abnormal clusters to be separated as widely as possible. Since the characteristic vectors of the clusters are 2-D, 2-D k-means clustering is used to group the clusters C_j into three groups G_h, $h = 1, 2, 3$.

First, k-means clustering is initialized as follows: The center of group G_1 is initialized by the characteristic vector of the dominant cluster C^+. Similarly, the abnormal group G_2 is initialized with the cluster C^- whose cluster center is the furthest from that of C^+ because C^- is most likely to be abnormal. The ambiguous group G_3 is initialized with the cluster that is approximately equidistant to C^+ and C^-.

Next, k-means clustering is executed to group the remaining clusters C_j into the three groups G_h. The distance between a cluster and a group is measured

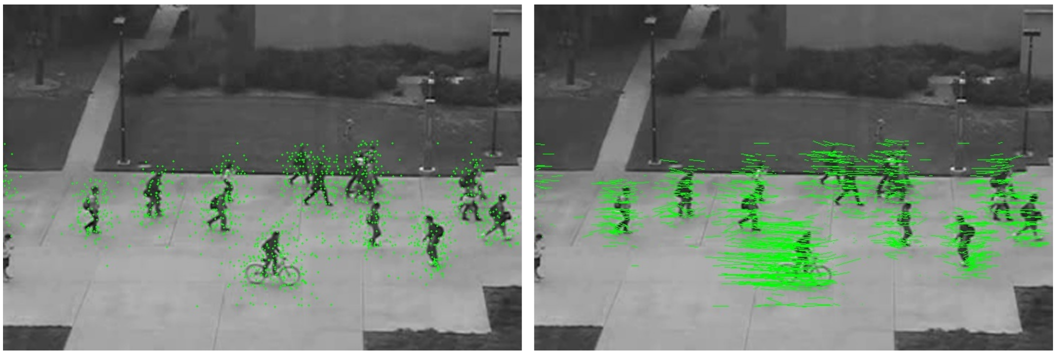

Figure 3.1: The pictures above were taken from UCSDped2, Test002 video at frame 150. The first picture visualizes the feature points (green points). The second picture visualizes the feature point trajectories (green lines).

in terms of the Euclidean distance between their characteristic vectors. After clustering, all the clusters in group G_1 are labeled as normal, those in G_2 abnormal, and those in G_3 ambiguous. In addition, the abnormal cluster that is nearest to G_1 is re-labeled as ambiguous so as to widen the separation between normal and abnormal clusters.

Stage 4: Anomaly Detection and Localization

After cluster labeling, the features f_i in abnormal clusters are labeled as abnormal features. The corresponding trajectory positions $\mathrm{x}_i(t)$ of f_i are labeled as abnormal feature points. Finally, the video frames that contain abnormal feature points are labeled as abnormal frames.

Example

Let us illustrate the proposed algorithm using test video "Test002" from UCSDped2. This video will be processed through the four main stages. In the first stage, the distinctive feature points p_i are extracted and tracked; Figure 3.1 visualizes the feature points and their trajectories.

In the second stage, the speeds of feature points s_i are divided into equal speed intervals $\mathcal{C}_j$, $j = 1, \ldots, m$. Figure 3.2 shows an example of equal speed intervals, where $m = 10$, and the black horizontal lines are the interval boundaries. From

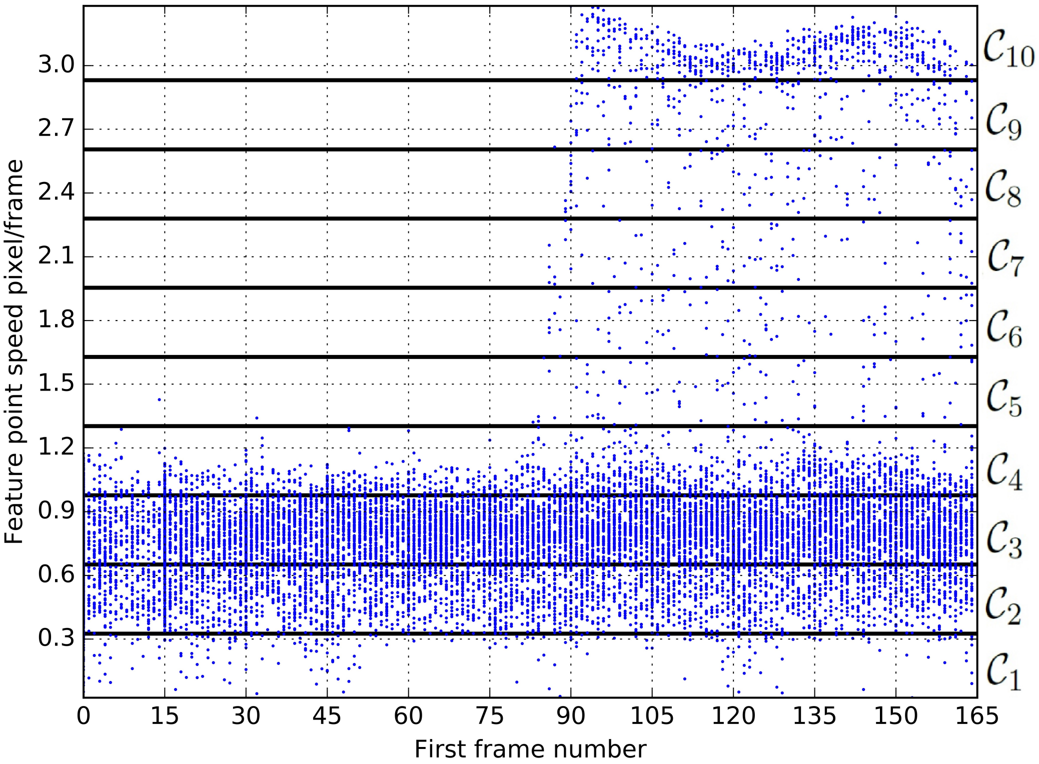

Figure 3.2: Plot of ten intervals of feature point speeds (UCSDped2, Test002 video). The x-axis is the first frame number and the y-axis is the speed in pixel/frame.

Figure 3.2, $\mathcal{C}_3$ (third interval) has a very large size. The last interval $\mathcal{C}_{10}$ has a smaller size, and it is very different in speed compering to $\mathcal{C}_3$. From Fig. 3.2 we can recognize that the points in each interval are approximately identically distributed, which makes the center for each interval (interval points mean) to be roughly in the middle, the intra-cluster difference for each interval equals roughly to the half of the interval length, and the inter-cluster difference between two adjacent intervals (smallest inter-cluster difference) equals roughly to the interval length. Thus, the largest intra-cluster difference is roughly equal to the half of smallest inter-cluster difference, in which that satisfies the constraint of well-separated clusters. Figure 3.3 also sketches normalized cluster size S_j (x-axis) and normalized cluster center F_j (y-axis).

In the third stage, The resultant intervals $\mathcal{C}_j$, are labeled based on normalized cluster size S_j and normalized cluster center F_j using 2-D k-means clustering. Figure 3.4 sketches initial cluster centers in stars (left plot) and the clustering results

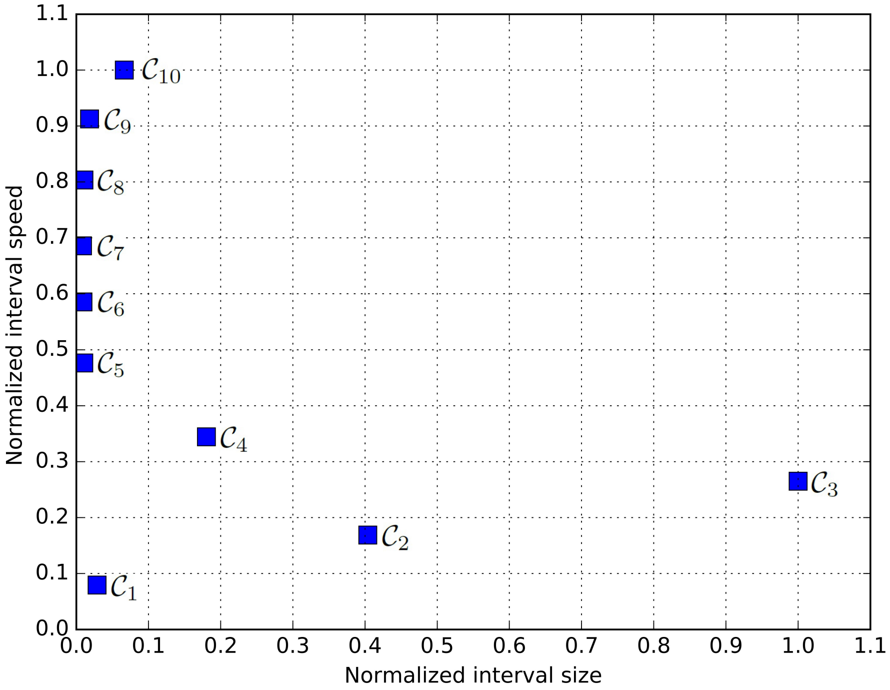

Figure 3.3: Plot of normalized size S_j (x-axis) and speed F_j (y-axis) of ten intervals (UCSDped2, Test002 video).

after few iterations (right plot). Based on resultant clusters, stage 4 localizes the anomalous feature points of abnormal group G_2 (red squares). Each anomalous feature point is visualized as a red transparent circle with radius equal to four pixels as shown in Figure 3.5.

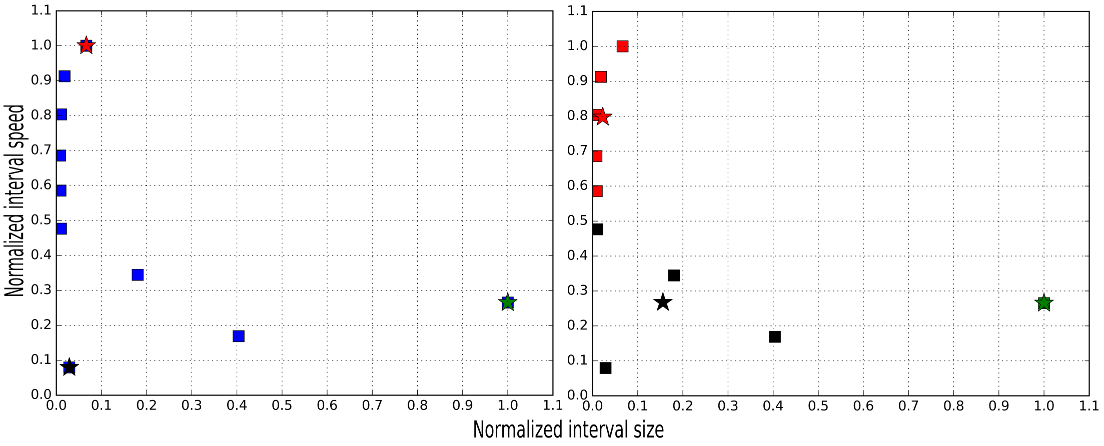

Figure 3.4: The left plot sketches initial cluster centers in stars. The right plot is clustering results after few iterations with the final centers (UCSDped2, Test002 video).

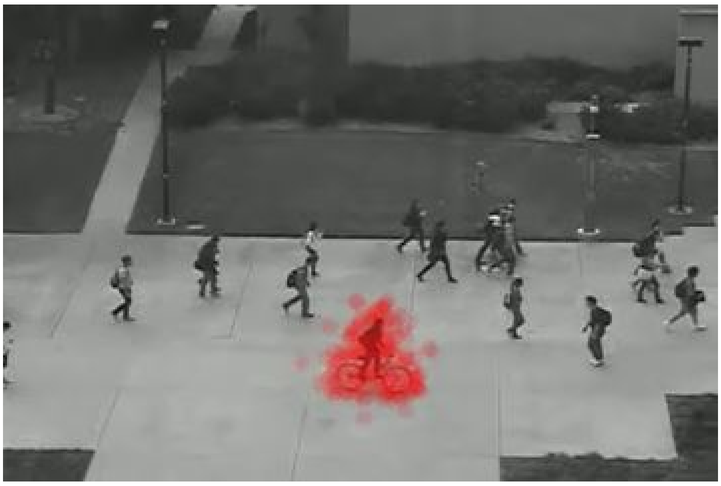

Figure 3.5: Visualization of anomalous feature points of UCSDped2 Test002 video at frame 150.

Chapter 4

Experiments and Discussions

4.1 Datasets

To evaluate the performance of the proposed method, OCCAM will be applied on five publicly available datasets namely UCSDped1, UCSDped2, Subway, UMN, and PETS2009 (Section 2.4). For OCCAM, motion directions were extracted from Subway video whereas motion speeds were extracted from the other videos. Next, feature clustering and cluster labeling were performed to detect abnormal feature points and abnormal frames.

4.2 Evaluation Methodology

The frame-level criterion and the pixel-level criterion are two common criteria, and they are used to evaluate the performance of anomaly detection and localization.

The frame-level criterion evaluates the detection by comparing the detection results (at frame level) to the video's frame-level ground-truth annotations. This criterion determines four primary parameters:

- True positive (TP): a frame is a true positive, if the algorithm detects an anomalous frame, and it matches the ground truth frame's annotation.
- False positive (FP): a frame is a false positive, if the algorithm detects an anomalous frame, and it does not match the ground truth frame's annotation.

- True negative (TN): a frame is a true negative, if the algorithm does not detect an anomalous frame, and it matches the ground truth frame's annotation.
- False negative (FN): a frame is a false negative, if the algorithm does not detect an anomalous frame, and it does not match the ground truth frame's annotation.

The pixel-level criterion evaluates the localization by comparing the localization results (at pixel level) to the video's pixel-level ground-truth annotations. A frame is a true positive if it is positive and at least 40 percent of its anomalous pixels are localized as proposed in many papers such as [9, 15]. A frame is a false positive if it is negative and has any of anomalous pixels are localized.

The two criteria measure false positive rate "FPR", true positive rate "TPR", and accuracy "ACC" using the following equations:

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}}. \tag{4.1}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{4.2}$$

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}. \tag{4.3}$$

4.3 Experimental Setup and Determination of m

There are two parameters need to be set. The first one l is the trajectory length, and the second one m is the number of clusters in stage 2. l is set to the default value as determined by Wang *et al.* [52, 53] and equal to 15. To determine the best value of m, a test was performed on USCDped1, UCSDped2, PETS2009scen1, and PETS2009scen2 datasets with varying the value of m. This test measures the accuracy (ACC) at frame-level criterion using different values of m, and shows the direct effect of m on the accuracy (ACC).

Table 4.1: Effect of different values of m on detection accuracy (ACC) from different datasets.

Dataset	**ACC**					
	$m = 5$	$m = 7$	$m = 10$	$m = 13$	$m = 16$	$m = 20$
USCDped1	0.8937	0.9028	**0.9072**	0.8810	0.8247	0.8196
USCDped2	0.8243	0.8593	0.9530	0.9595	**0.9635**	0.9460
PETS2009scen1	0.8738	0.9005	**0.9100**	0.8924	0.9063	0.9013
PETS2009scen2	0.8773	0.8992	**0.9874**	0.9211	0.9801	0.9727

Table 4.1 shows that with $m = 10$, the dataset's accuracy (ACC) is sufficiently high. For USCDped1, PETS2009scen1, and PETS2009scen2, the best value of m is 10, and it achieves the highest accuracy (ACC). For USCDped2, the best value of m that achieves the highest ACC is 16, but at $m = 10$ the ACC does not significantly change. Therefore, m is fixed at 10 for subsequent tests.

4.4 Benefit of Ambiguous Clusters

This test illustrates the benefit of having ambiguous clusters. A variant of OCCAM, denoted as OCCAM−, was tested such that its cluster labeling stage ran k-means clustering with $k = 2$ for normal and abnormal groups, without ambiguous group. Existing methods also label their clusters as either normal or abnormal, without ambiguous clusters. Both OCCAM and OCCAM− were tested on the common test videos discussed in Section 2.4. True positive rate (TPR) and false positive rate (FPR) were measured for the detected abnormal frames.

Table 4.2 compares the results of OCCAM and OCCAM−. For all test videos at frame-level, OCCAM's TPR is slightly smaller than that of OCCAM−, but OCCAM's FPR is significantly smaller than that of OCCAM−. That is, by regarding some clusters as ambiguous, OCCAM makes significantly fewer false detections than does OCCAM− without significantly sacrificing its true detection rate.

Table 4.2: Benefit of ambiguous clusters. OCCAM (O) has slightly smaller TPR, but significantly smaller FPR compared to OCCAM− (O−).

Test videos	TPR		FPR	
	O	O−	O	O−
UCSDped1	0.887	0.982	0.214	0.741
UCSDped2	0.957	0.994	0.154	0.677
Subway Entrance	0.835	0.942	0.152	0.773
Subway Exit	0.850	0.967	0.136	0.634
UMN	0.910	0.999	0.002	0.818
PETS2009 Scene 1	0.892	0.973	0.079	0.482
PETS2009 Scene 2	0.987	0.999	0.125	0.395

4.5 Performance Comparison

OCCAM's results are compared based on frame-level criterion with all of the existing methods discussed in Section 2.2 and Section2.3. These methods belong to the following categories:

- Supervised (training from only normal examples): AMDN [5], BM [6], CI [7], GPR [8], H-MDT-CRF [9], IBC [10], IEP [11], LMH [12], Local-KNN [13], LSTM [14], MDT [15], MPPCA [16], OF [17], SF [17], Sabokrou [18], SRC [19], and STMC [20]. [17] tested both OF and SF methods.
- Unsupervised: DC [21], FF [22], OADC-SA [23], OWC-MTT [24], and STC [25].

In this section, most of existing methods use the receiver operating characteristic (ROC) curve to present their results. This curve combines the two measurements, FPR (x-axis) and TPR (y-axis), and plots multiple points by varying threshold. The results of others' ROC curves are collected either by directly contacting the authors or by using software to trace the curve points from different papers. OCCAM has no parameter to tune. So, we plot only a point instead of a ROC curve for OCCAM.

Most of these methods were tested only on some of the test videos. The test results on UCSDped1, UCSDped2, and Subway were reported as ROC curves.

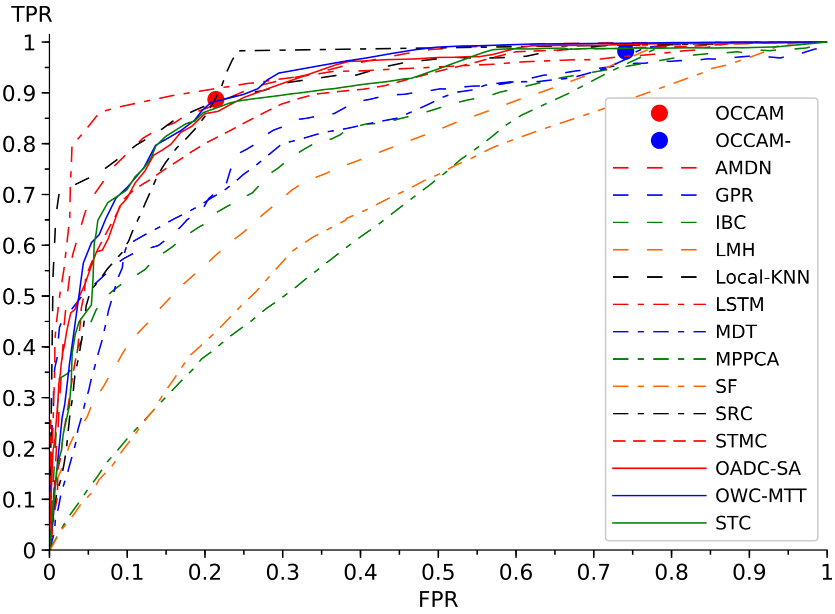

Figure 4.1: Performance comparison at frame-level. 14 methods are available for comparison on UCSDped1 videos. Supervised methods (dashed lines), unsupervised methods (solid lines).

For the test results on UMN, some papers reported ROC curves whereas others reported only accuracy. For PETS2009, only accuracy was reported. ROC curves are not reported for H-MDT-CRF [9] on UCSDped1 and UCSDped2, LMH [12] and MPPCA [16] on Subway, and Sabokrou [18] on UMN. Therefore, they are not included in our ROC graphs.

For UCSDped1 and UCSDped2 (Fig. 4.1 and Fig. 4.2) and UMN videos (Fig. 4.5), OCCAM is among the best performers compared to existing methods. For the Subway videos (Fig. 4.3 and Fig. 4.4), OCCAM's performance is comparable to those of existing methods that are far more complex than OCCAM. For the same FPR, OCCAM achieves the highest TPR compared to existing methods for UCSDped2 (Fig. 4.2), Subway exit (Fig. 4.4), and UMN (Fig. 4.5), the 3rd highest TPR for UCSDped1 (Fig. 4.2), and the 4th highest TPR for Subway entrance (Fig. 4.3). In applications where high FPR is tolerable, OCCAM can run as OCCAM− without ambiguous clusters. Then, OCCAM− achieves TPR of close to 1.0 for all test cases. Fig. 4.1, 4.2, 4.3, 4.4 and 4.5 also show that existing unsupervised methods can perform as well as or better than supervised methods.

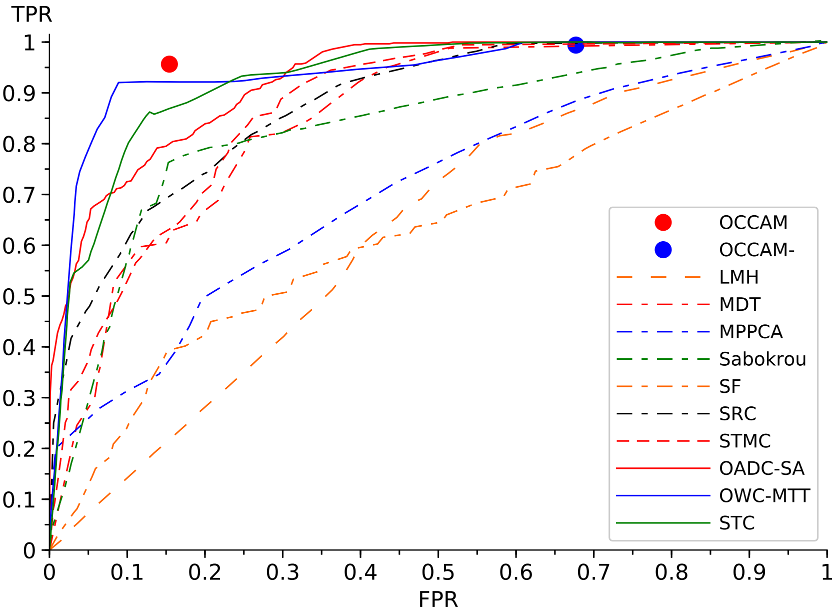

Figure 4.2: Performance comparison at frame-level. 10 methods are available for comparison on UCSDped2 videos. Supervised methods (dashed lines), unsupervised methods (solid lines).

Table 4.3: Frame-level performance comparison on UMN. OCCAM has the highest overall accuracy. (S) Supervised method, (U) unsupervised method.

Method	Type	ACC			
		Scene 1	Scene 2	Scene 3	Overall
OCCAM	U	0.9862	**0.9742**	**0.9934**	**0.9819**
BM [6]	S	**0.9903**	0.9536	0.9663	0.9640
CI [7]	S	0.9062	0.8506	0.9158	0.8791
SF [17]	S	0.8441	0.8235	0.9083	0.8509
SRC [19]	S	0.9052	0.7848	0.9270	0.8470
DC [21]	U	0.9704	0.9534	0.9647	0.9598
FF [22]	U	0.8869	0.8000	0.7792	0.8104

Some existing papers reported only accuracy on UMN and PETS2009 videos. Tables 4.3, 4.4 and 4.5 show that OCCAM is more accurate than these methods for both UMN and PETS2009.

For UCSDped1 and UCSDped2 videos, Li and Mahadevan [9, 15] also proposed a pixel-level criterion to measure the spatial accuracy of detected abnormal

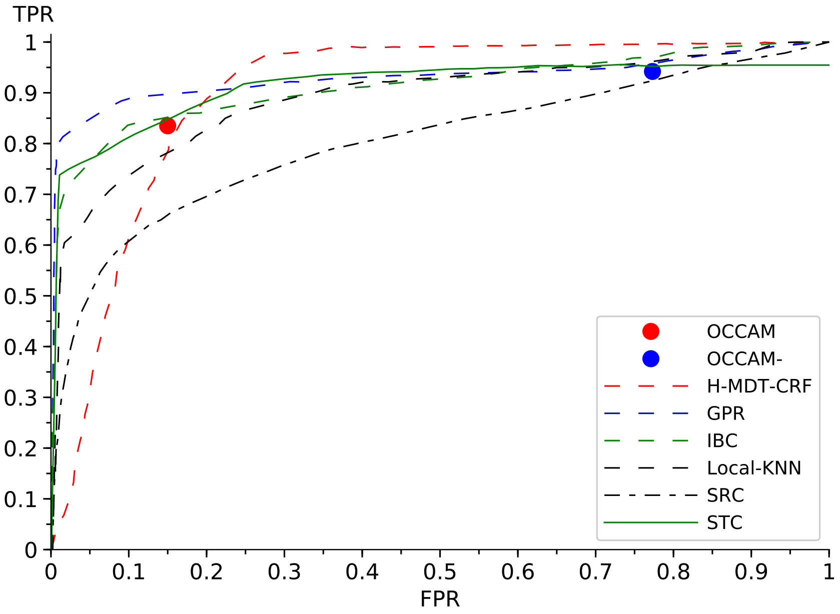

Figure 4.3: Performance comparison at frame-level. 6 methods are available for comparison on Subway entrance video. Supervised methods (dashed lines), unsupervised methods (solid lines).

frames. This error measure depends on the number of detected abnormal pixels in an abnormal region. Since OCCAM detects only selected pixels in these regions instead of the whole regions, pixel-level criterion is not appropriate for OCCAM. Instead, this thesis measures spatial accuracy in terms of precision, which is the percentage of detected abnormal pixels that are true positives. OCCAM achieves abnormal pixel detection precision of 0.72 for UCSDped1 and 0.78 for UCSDped2. Moreover, most of the false positive pixels are located around the abnormal regions. On the other hand, the spatial precision of OCCAM$-$ on UCSDped1 and UCSDped2 is, respectively, 0.37 and 0.40, which is much lower than that of OCCAM. Therefore, ambiguous clusters are important for OCCAM to achieve high spatial accuracy in detecting abnormal pixels. For visualization purposes, OCCAM's localization result is compared directly with pixel-level ground truth by sketching a red transparent circle with radius equal to four pixels for each anomalous feature point. The localization results applied on USCDped1 and USCDped2 are visualized in Figure 4.6 and Figure 4.7 respectively.

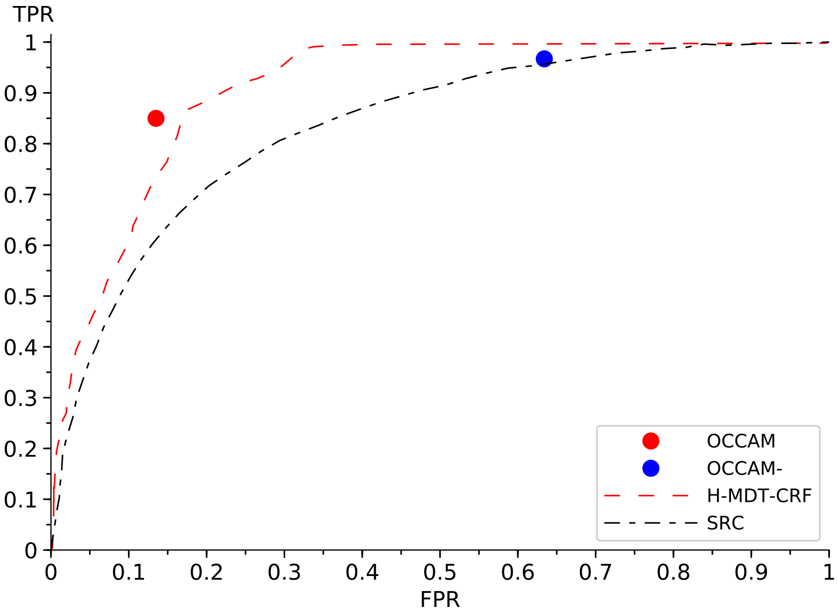

Figure 4.4: Performance comparison at frame-level. 2 methods are available for comparison on Subway exit video. Supervised methods (dashed lines), unsupervised methods (solid lines).

Table 4.4: Frame-level performance comparison on videos of PETS2009, 1st scenario. OCCAM has the highest overall accuracy. (S) Supervised method, (U) unsupervised method.

Method	Type	ACC				
		View 1	View 2	View 3	View 4	Overall
OCCAM	U	0.8964	**0.8514**	**0.9324**	**0.9596**	**0.9100**
BM [6]	S	**0.9245**	0.8302	0.8962	0.9057	0.8892
CI [7]	S	0.5660	0.8302	0.8113	0.5283	0.6040
SF [17]	S	0.6321	0.7076	0.5283	0.4811	0.5873
FF [22]	U	0.3774	0.3774	0.3774	0.3774	0.3774

4.6 Discussions

A. Computational Complexity: The time complexity of OCCAM is mainly related to the two clustering levels: linear one-dimensional clustering (stage 2) and two-dimensional k-means clustering (stage 3). In stage 2, the timing is linear and depends on the number of feature points n; the time complexity of the second stage is $O(n)$. In stage 3, the timing is affected by four

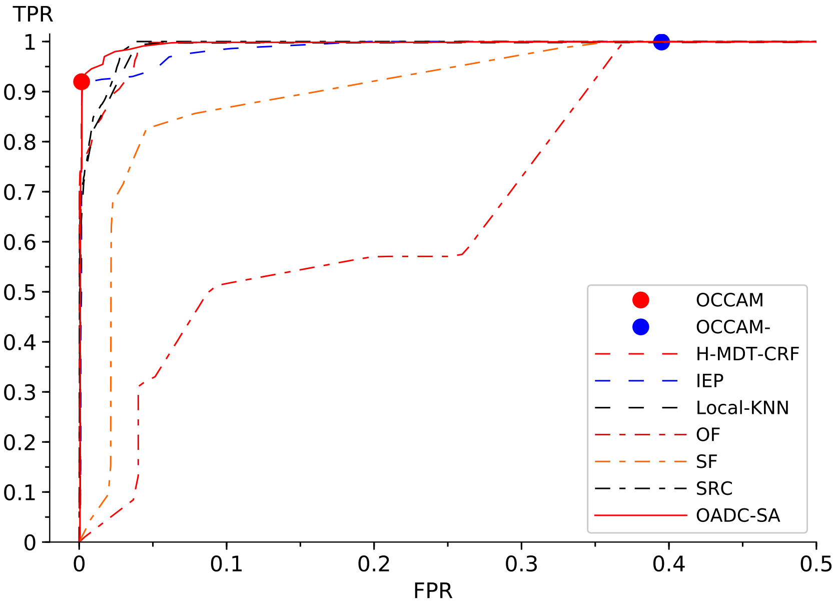

Figure 4.5: Performance comparison on UMN video. 7 methods are available for comparison at frame-level. Supervised methods (dashed lines), unsupervised methods (solid lines).

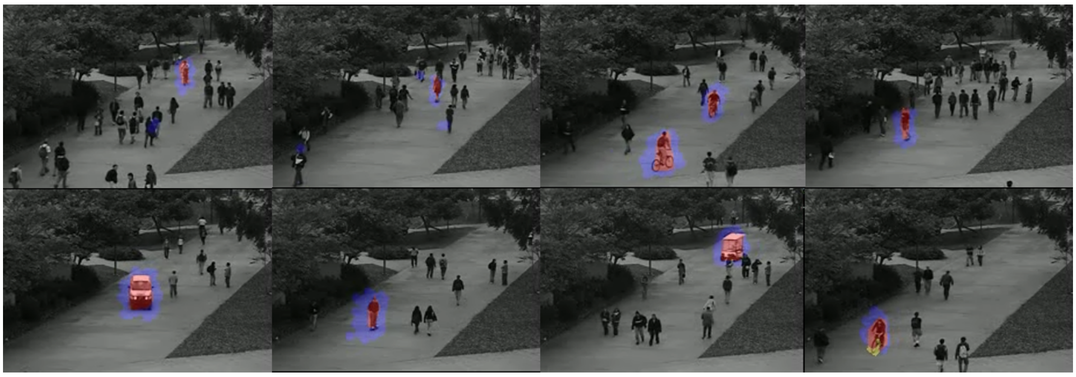

Figure 4.6: The comparison of ground truth and localization results generated by OCCAM on UCSDped1. Red pixels represent the TP pixels, blue pixels represent FP pixels, yellow pixels represent FN pixels, and all other pixels represent TN pixels.

parameters: number of groups G_h, number of iterations N, number of input clusters m and the dimension of their characteristic vector. The number of groups and the cluster dimension are fixed and equal to 3 and 2 respectively. Therefore the time complexity of the third stage is $O(mN)$. Most of the time, the number of iterations needed to converge is very small because the

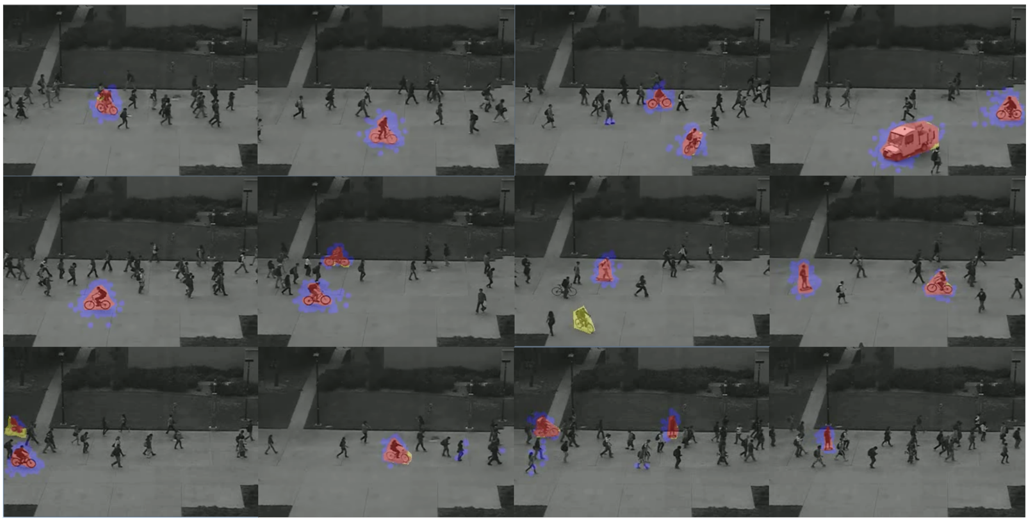

Figure 4.7: The comparison of ground truth and localization results generated by OCCAM on UCSDped2. Red pixels represent the TP pixels, blue pixels represent FP pixels, yellow pixels represent FN pixels, and all other pixels represent TN pixels.

Table 4.5: Frame-level performance comparison on videos of PETS2009, 1st scenario. OCCAM has the highest overall accuracy. (S) Supervised method, (U) unsupervised method.

Method	Type	ACC				
		View 1	View 2	View 3	View 4	Overall
OCCAM	U	**0.9735**	**0.9894**	**0.9920**	**0.9947**	**0.9874**
BM [6]	S	0.9601	0.9415	0.9521	0.9149	0.9422
CI [7]	S	0.9495	0.9202	0.9415	0.8936	0.9262
SF [17]	S	0.9122	0.8936	0.9468	0.6463	0.8497
FF [22]	U	0.945	0.6383	0.9548	0.9681	0.8766

number of groups G_h and m are small too.

B. The effect of projected feature points on UCSDped1: As mentioned in Section 3.2 (stage 1), the amount of perspective distortion in UCSDped1 is very high. This perspective distortion gives wrong speed of feature points, which affects overall accuracy. To overcome this distortion, the feature point positions of UCSDped1 only are projected into the ground plane using an estimated Homography. To show the effect of the point projection, the

Table 4.6: The effect of projected feature points on the accuracy of OCCAM applied on UCSD.

Method	**ACC**	
	UCSDped1	UCSDped2
OCCAM with projection	**0.912**	0.961
OCCAM without projection	0.641	**0.963**

accuracy (ACC) of OCCAM applied on UCSD at frame-level is computed with/without projection in Table 4.6. Form this table, it is clear that projecting the points significantly improves the accuracy of OCCAM applied on UCSDped1 videos. For OCCAM applied on UCSDped2 videos, the accuracy almost the same with/without the projection.

Chapter 5

Conclusions

This thesis investigated the essential components (effective features) required for effective unsupervised detection of anomalies in surveillance videos of pedestrians. It shows that relatively simple but well-designed unsupervised algorithm like OCCAM can perform as well as or better than existing supervised and unsupervised methods. In particular, simple but informative features such as motion direction and motion speed are sufficient for achieving high TPR with low FPR. Moreover, inclusion of ambiguous clusters in the cluster labeling process reduces FPR significantly without sacrificing TPR much. At the same FPR, OCCAM achieves among the highest TPR compared to existing methods. It also has the highest accuracy for UMN and PETS2009 videos compared to existing methods that reported only accuracy. In applications where high FPR is tolerable, OCCAM can run as OCCAM$-$ without ambiguous clusters. Then, OCCAM$-$ achieves TPR of close to 1.0 for all test cases. With ambiguous clusters, OCCAM's spatial precision of detecting abnormal pixels is also very high. In general, OCCAM and existing unsupervised methods can perform as well as or better than supervised methods. Therefore, our research results can serve as a useful benchmark for testing new algorithms and for developing more advanced algorithms that require features other than motion speed and direction.

The lesson learned in this thesis is that the analysis of input data is the key to address the right problem and derive the possible solution. Additionally, the objective of scientific research is not necessarily to come up with the best algorithm, but rather to understand the approach of solving the right problem in more structured way. This way of understating seems more logical and systematic.

References

[1] T. Hospedales, S. Gong, and T. Xiang, "Video behaviour mining using a dynamic topic model," *International journal of computer vision*, vol. 98, no. 3, pp. 303–323, 2012.

[2] N. Sulman, T. Sanocki, D. Goldgof, and R. Kasturi, "How effective is human video surveillance performance?," in *Pattern Recognition, 2008. ICPR. 19th International Conference on.* IEEE, 2008, pp. 1–3.

[3] A. A. Sodemann, M. P. Ross, and B. J. Borghetti, "A review of anomaly detection in automated surveillance," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1257–1272, 2012.

[4] T. Gandhi and M. M. Trivedi, "Pedestrian protection systems: Issues, survey, and challenges," *IEEE Transactions on intelligent Transportation systems*, vol. 8, no. 3, pp. 413–430, 2007.

[5] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," in *Proc. BMVC*, 2015, pp. 1–12.

[6] S. Wu, H.-S. Wong, and Z. Yu, "A bayesian model for crowd escape behavior detection," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 24, no. 1, pp. 85–98, 2014.

[7] S. Wu, B. E. Moore, and M. Shah, "Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes," in *Proc. CVPR*, 2010, pp. 2054–2060.

[8] K. Cheng, Y. Chen, and W. Fang, "Video anomaly detection and localization using hierarchical feature representation and gaussian process regression," in *Proc. CVPR*, 2015, pp. 2909–2917.

[9] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. PAMI*, vol. 36, no. 1, pp. 18–32, 2014.

[10] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *IJCV*, vol. 74, no. 1, pp. 17–31, 2007.

[11] X. Cui, Q. Liu, M. Gao, and D. N. Metaxas, "Abnormal detection using interaction energy potentials," in *Proc. CVPR*, 2011, pp. 3161–3167.

[12] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Trans. PAMI*, vol. 30, no. 3, pp. 555–560, 2008.

[13] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *Proc. CVPR*, 2012, pp. 2112–2119.

[14] Y. Feng, Y. Yuan, and X. Lu, "Deep representation for abnormal event detection in crowded scenes," in *Proc. ACM MM*, 2016, pp. 591–595.

[15] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes.," in *Proc. CVPR*, 2010, vol. 249, p. 250.

[16] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates," in *Proc. CVPR*, 2009, pp. 2921–2928.

[17] R. Mehran, A. Oyama, and M. Shah, "Abnormal crowd behavior detection using social force model," in *Proc. CVPR*, 2009, pp. 935–942.

[18] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Real-time anomaly detection and localization in crowded scenes," in *Proc. CVPR Workshops*, 2015, pp. 56–62.

[19] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proc. CVPR*, 2011, pp. 3449–3456.

[20] Y. Cong, J. Yuan, and Y. Tang, "Video anomaly search in crowded scenes via spatio-temporal motion context," *IEEE Trans. on Information Forensics and Security*, vol. 8, no. 10, pp. 1590–1599, 2013.

[21] C.-Y. Chen and Y. Shao, "Crowd escape behavior detection and localization based on divergent centers," *IEEE Sensors Journal*, vol. 15, no. 4, pp. 2431–2439, 2015.

[22] D.-Y. Chen and P.-C. Huang, "Motion-based unusual event detection in human crowds," *Journal of Visual Communication and Image Representation*, vol. 22, no. 2, pp. 178–186, 2011.

[23] Y. Yuan, J. Fang, and Q. Wang, "Online anomaly detection in crowd scenes via structure analysis," *IEEE Trans. on Cybernetics*, vol. 45, no. 3, pp. 548–561, 2015.

[24] H. Lin, J. D. Deng, B. J. Woodford, and A. Shahi, "Online weighted clustering for real-time abnormal event detection in video surveillance," in *Proc. ACM MM*, 2016, pp. 536–540.

[25] M. J. Roshtkhari and M. D. Levine, "Online dominant and anomalous behavior detection in videos," in *Proc. CVPR*, 2013, pp. 2611–2618.

[26] Saad Ali and Mubarak Shah, "A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on.* IEEE, 2007, pp. 1–6.

[27] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on.* IEEE, 2008, pp. 1–8.

[28] Bruce D Lucas, Takeo Kanade, et al., "An iterative image registration technique with an application to stereo vision," 1981.

[29] Ross Messing, Chris Pal, and Henry Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE, 2009, pp. 104–111.

[30] Xi Li, Anthony Dick, Chunhua Shen, Anton Van Den Hengel, and Hanzi Wang, "Incremental learning of 3d-dct compact representations for robust visual tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 4, pp. 863–881, 2013.

[31] Samuel S Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 1, pp. 5–18, 2004.

[32] Antoni B Chan and Nuno Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 5, pp. 909–926, 2008.

[33] Dirk Helbing and Peter Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, pp. 4282, 1995.

[34] Saad Ali, Arslan Basharat, and Mubarak Shah, "Chaotic invariants for human action recognition," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on.* IEEE, 2007, pp. 1–8.

[35] Zoran Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on.* IEEE, 2004, vol. 2, pp. 28–31.

[36] Berthold KP Horn and Brian G Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[37] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[39] Karen Simonyan and Andrew Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[40] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[41] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning.* ACM, 2008, pp. 1096–1103.

[42] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue, "Modeling spatial-temporal clues in a hybrid deep learning framework for video classification," in *Proceedings of the 23rd ACM international conference on Multimedia.* ACM, 2015, pp. 461–470.

[43] Chris Stauffer and W Eric L Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.* IEEE, 1999, vol. 2, pp. 246–252.

[44] Dashan Gao and Nuno Vasconcelos, "Decision-theoretic saliency: computational principles, biological plausibility, and implications for neurophysiology and psychophysics," *Neural computation*, vol. 21, no. 1, pp. 239–271, 2009.

[45] David M Blei, Andrew Y Ng, and Michael I Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[46] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[47] M. J. Roshtkhari and M. D. Levine, "An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions," *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1436–1452, 2013.

[48] UCSD, "UCSD, anomaly detection dataset," available on-line: http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm, 2010.

[49] UMN, "UMN, unusual crowd activity dataset," available on-line: http://mha.cs.umn.edu/proj_events.shtml, 2006.

[50] PETS, "PETS2009: event recognition," available on-line: http://www.cvg.reading.ac.uk/PETS2009/a.html#s3, 2009.

[51] BEHAVE, "BEHAVE: Computer-assisted pre-screening of video streams for unusual activities," available on-line: http://homepages.inf.ed.ac.uk/rbf/BEHAVE/, 2007.

[52] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. CVPR*, 2011, pp. 3169–3176.

[53] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013.

[54] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Scandinavian Conference on Image Analysis*. Springer, 2003, pp. 363–370.