ADAPTIVE SPARSE REPRESENTATIONS
FOR VIDEO ANOMALY DETECTION

A Dissertation in
Electrical Engineering
by
Xuan Mo

© 2014 Xuan Mo

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2014
The dissertation of Xuan Mo was reviewed and approved* by the following:

Vishal Monga
Monkowski Assistant Professor of Electrical Engineering
Dissertation Advisor, Chair of Committee

W. Kenneth Jenkins
Professor of Electrical Engineering

William E. Higgins
Distinguished Professor of Electrical Engineering

Jesse L. Barlow
Professor of Computer Science and Engineering

Raja Bala
Principal Imaging Scientist of Xerox Research Center Webster
Special Member

Kultegin Aydin
Professor of Electrical Engineering
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School.
Abstract

Video surveillance systems are widely used in the transportation domain to identify unusual patterns such as traffic violations, accidents, unsafe driver behavior, street crime, and other suspicious activities. As the volume of video data increases, most video analysis requires significant human supervision. However, since human supervision is not scalable, a software-aided real-time video surveillance system is desirable. Video anomaly detection problem has been formulated as a well-known likelihood ratio test (LRT) problems, under the idealized assumption of the knowledge of the distributions of both normal events and anomalous events. Often, ample training corresponding to anomalous events is assumed to be available (supervised setting). In many real-world problems, however, both normal and anomalous distributions are generally unknown and difficult to estimate even when the training data is available.

The key open challenges in video anomaly detection comprise: 1.) presence of noise in surveillance videos and occlusions in transportation videos; 2.) detection of anomalies involving multiple objects; 3.) lack of training samples representing anomalous events; 4.) representation of event using multiple features (known for-
mally as video event encoding); 5.) identification of anomalies in unstructured scenario where preparation of a dictionary clearly separated into class-specific sub-dictionaries is impossible. Recently sparse reconstruction techniques have been used for image classification, and shown to provide excellent robustness to occlusion. This progress has also been leveraged for sparsity-based video-anomaly detection where test events are expressed as sparse linear combinations of example events from a given (normal or anomalous) class. This dissertation explores novel and adaptive sparse representations for addressing open challenges in video anomaly detection. First, we develop a new joint sparsity model for anomaly detection that enables the detection of joint anomalies involving multiple objects and then we propose outlier rejection measure for unsupervised video anomaly detection. Second, we introduce non-linearity into the linear sparsity model and dictionary design and optimization technique to enable superior class separability and dictionary compactness. Third, we propose to extend sparsity models based on single feature representations to more sophisticated sparse models based on multiple feature representations. Finally, we introduce low rank sparsity prior to our sparsity model which perfectly handle the unstructured scenario. The contributions in this dissertation successfully address all the five open challenges mentioned at the beginning of this paragraph. We extensively test on several real world video data sets involving both single and multiple object anomalies. Results show marked improvements in detection of anomalies in both supervised and unsupervised cases when using the proposed sparsity models.
Table of Contents

List of Figures x

List of Tables xiv

Acknowledgments xvii

Chapter 1

Introduction 1

1.1 Motivation ........................................ 1

1.2 Review of Related Work ............................ 3

1.2.1 Supervised Approaches ......................... 4

1.2.2 Unsupervised Approaches ....................... 4

1.2.2.1 Analysis-by-Synthesis Approaches ......... 5

1.2.2.2 Contextual and Behavioral Approaches .... 5

1.2.2.3 Trajectory-based Approaches ............... 6

1.2.3 Open Problems ................................. 9

1.3 Dissertation Contributions and Organization ........ 12
Chapter 2

Sparse Representations for Video Anomaly Detection

2.1 Introduction ................................................. 16

2.2 Existing Sparsity-based Techniques for Video Anomaly Detection . 17

2.2.1 Abnormal Behavior Detection via
        Sparse Reconstruction Analysis of Trajectory .......... 18

2.2.2 Online Detection of Unusual Events in Videos via
        Dynamic Sparse Coding .................................. 21

2.3 Conclusion .................................................. 22

Chapter 3

Joint Sparsity Model for Trajectory Based Video Anomaly Detection

3.1 Introduction .................................................. 24

3.2 Observations and Motivation for Joint Sparsity Model ............... 25

3.3 Joint Sparsity Model for Trajectory-based Video Anomaly Detection 27

3.3.1 Notation .................................................... 27

3.3.2 Joint Sparsity Model ...................................... 27

3.3.3 Auxiliary Optimization Problem .......................... 32

3.3.4 Classification into Event Classes .......................... 33

3.3.5 Anomaly Detection via Outlier Rejection .................. 34

3.4 Experimental Results ........................................ 35

3.4.1 Trajectory Extraction ..................................... 35

3.4.2 Video Datasets ............................................ 35

3.4.3 Detection Rates for Single Object Anomaly Detection .... 38
Chapter 6

Low Rank Sparsity Prior for Unstructured Video Anomaly Detection

6.1 Introduction ................................................. 91
6.2 Motivation: Low Rank Sparsity Prior ..................... 92
6.3 Low Rank Sparsity Prior for Anomaly Detection .......... 94
6.4 Experimental Result ....................................... 98
   6.4.1 Experimental Setup ................................... 98
   6.4.2 Comparison against state of the art Trajectory-based Video
       Anomaly Detection Techniques .......................... 99
   6.4.3 Performance Variation with Regular Parameter \(\lambda\) ..... 100
   6.4.4 Computational Benefits and Trade Off ............... 101
6.5 Conclusion ................................................ 102

Chapter 7

Conclusion ...................................................... 103

7.1 Summary of Contributions .................................. 103
7.2 Future Research .......................................... 105

Bibliography .................................................. 108
List of Figures

1.1 Examples of traffic-related anomalies: (a) unattended baggage; (b) car approaching a pedestrian; (c) crossing the yellow line; (d) running a red light. ................................................................. 2

1.2 Taxonomy of existing techniques. .......................................................... 3

1.3 Illustration of one class SVM method by Piciarelli et al. [1] on 2-D data. Figure obtained from [1]. ................................................................. 9

1.4 An example of noise video ................................................................. 10

1.5 An example of video occlusion .......................................................... 11

1.6 (a) structured scenario (b) unstructured scenario. .............................. 12

2.1 An example illustration of trajectory classification using a sparse reconstruction model. The training dictionary consist of 2 classes (red trajectories are class 1, blue trajectories are class 2), each class contains 4 different trajectories. The test trajectory (the green trajectory) can be well represented by the linear combination of the first trajectory and the third trajectory from class 1. ............... 19
3.1 Trajectory extraction: (a) background subtraction and blob analysis to identify objects, (b) blob centroid calculation and trajectory derivation.

3.2 Example frames of single-object anomalies: (a) a man suddenly falls on floor – from the CAVIAR data set and (b) a driver backs his car in front of stop sign – from the Xerox Stop Sign data set.

3.3 Example frames of multiple-object anomalies: (a) a vehicle almost hits a pedestrian – from the AVSS data set, (b) a car (marked by red rectangle) violates the stop sign rule – from the Xerox Stop Sign data set and (c) a car fails to yield to oncoming car while turning left – from the Xerox Intersection data set.

3.4 Synthetic trajectories: (a) three different paths for a car on intersection, (b) examples of normal training class, (c) anomaly type 1: running traffic light, (d) anomaly type 2: turning left fail to yield on coming traffic.

3.5 ROC curves of 2 different methods on CAVIAR data.

3.6 ROC curves of 2 different methods on Xerox Stop Sign data.

3.7 ROC curves of 3 different methods on Synthetic data.

4.1 Occlusion examples: (a) moving occlusion: a car is occluded by another car (b) static occlusion: a car is occluded by the stop sign.

4.2 An example of occluded trajectories from the Xerox Stop Sign video data.

4.3 ROC curves for single-object anomaly detection – CAVIAR video data set.
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>ROC curves for single-object anomaly detection – Xerox Stop Sign video dataset.</td>
<td>56</td>
</tr>
<tr>
<td>4.5</td>
<td>ROC curves for multiple-object anomaly detection – Xerox Intersection dataset.</td>
<td>58</td>
</tr>
<tr>
<td>4.6</td>
<td>An example of occluded trajectories from the Xerox Stop Sign video data.</td>
<td>67</td>
</tr>
<tr>
<td>4.7</td>
<td>Performance ratio curves with respect to the proportion of original dictionary size - Xerox Stop Sign video data set.</td>
<td>68</td>
</tr>
<tr>
<td>4.8</td>
<td>Running time with respect to the proportion of original dictionary size - Xerox Stop Sign video data set.</td>
<td>69</td>
</tr>
<tr>
<td>4.9</td>
<td>Performance ratio curves with respect to the proportion of original dictionary size - CAVIAR data set.</td>
<td>70</td>
</tr>
<tr>
<td>4.10</td>
<td>Running time with respect to the proportion of original dictionary size - CAVIAR data set.</td>
<td>71</td>
</tr>
<tr>
<td>5.1</td>
<td>Structure of simultaneous sparsity model and its two major applications: (1). multi-feature video anomaly detection; (2) multi-camera video anomaly detection.</td>
<td>75</td>
</tr>
<tr>
<td>5.2</td>
<td>(a) Row sparsity model: sparse coefficients in the same row should be activated simultaneously and (b) Simultaneous sparsity model: the sparse coefficient in the same class should be activated simultaneously</td>
<td>76</td>
</tr>
<tr>
<td>5.3</td>
<td>Structure of simultaneous sparsity model for multi-object: active curves of different objects should have the same shape.</td>
<td>80</td>
</tr>
</tbody>
</table>
5.4 Example anomalies in structured scenario: (a) a man suddenly falls on floor – from the CAVIAR data set, (b) a driver backs his car in front of stop sign – from the Xerox Stop Sign data set and (c) a car fails to yield to oncoming car while turning left – from the Xerox Intersection data set.

5.5 ROC curves for multiple-object anomaly detection – Xerox Intersection data set.

6.1 (a) structured scenario (b) unstructured scenario.

6.2 Example anomalies in unstructured scenario: (a) a vehicle almost hits a pedestrian – from the AVSS data set and (b) a car fails to yield to oncoming car while turning left. – from the PDTV data.

6.3 Detection rates curves with respect to the value of λ.
List of Tables

3.1 Confusion matrices of proposed method and the method in [1] on CAVIAR data for single object anomaly detection 39

3.2 Confusion matrices of proposed method and the method in [1] on Xerox Stop Sign Data for single object anomaly detection 39

3.3 Confusion matrices of proposed method, method in [1] and [2] on synthetic data for multiple object anomaly detection 41

4.1 Confusion matrices of proposed and state-of-the art trajectory based methods on Xerox Stop Sign occluded data – single-object anomaly detection 52

4.2 Confusion matrices of proposed and state-of-the art trajectory based methods on CAVIAR data – single-object anomaly detection 53

4.3 Confusion matrices of proposed and state-of-the art trajectory based methods on the Xerox Stop Sign data – single-object anomaly detection 54

4.4 Confusion matrices of proposed and state-of-the art trajectory based methods on the Xerox Stop Sign data – multiple-object anomaly detection 56
4.5 Detection rates of proposed and state-of-the art trajectory based methods on AVSS data – multiple-object anomaly detection. 57

4.6 Confusion matrices of proposed and state-of-the art trajectory based methods on the Xerox Intersection dataset – multiple-object anomaly detection. 58

4.7 Confusion matrices of proposed and state-of-the art trajectory based methods on Xerox Stop Sign occluded data – supervised, single-object anomaly detection. KSM detects anomalies using Eq. (2.4). 66

4.8 Confusion matrices on Xerox Stop Sign data - Dictionary Learning 68

4.9 Confusion matrices on CAVIAR data - Dictionary Learning 70

4.10 Confusion matrices on Xerox Stop Sign data: supervised and unsupervised cases - Dictionary Learning 71

4.11 Confusion matrices on CAVIAR data: supervised and unsupervised cases - Dictionary Learning 72

5.1 Confusion matrices of three state-of-the art video anomaly detection methods on the Xerox Stop Sign data – supervised, single-object anomaly detection. 87

5.2 Confusion matrices of proposed and state-of-the art video anomaly detection methods on CAVIAR data – supervised, single-object anomaly detection. 88

5.3 Confusion matrices of proposed and state-of-the art video anomaly detection methods on the Xerox Intersection data – supervised, multi-object anomaly detection. 88
5.4 Confusion matrices of proposed and state-of-the art video anomaly detection methods on the Xerox Intersection data – unsupervised, multi-object anomaly detection. ........................................ 89

6.1 Confusion matrices of proposed and state-of-the art video anomaly detection methods on PDTV data – unsupervised anomaly detection. ................................................................. 100

6.2 Confusion matrices and running time of LRSP and JSM on PDTV data – unsupervised anomaly detection. ................................................................. 101
Acknowledgments

I would like to begin by thanking my advisor Prof. Vishal Monga. The luckiest thing during my Ph.D. is that my advisor treat me as a friend more than a student. As a advisor, he spent countless hours discussing research ideas and teaching me how to become a good researcher. As a friend, he encouraged me when I felt frustrated and helped me to realize my potential. I also enormously appreciate his patience in revising my paper word by word. This Ph.D. would never have been achieved without his guidance and support.

I would also like to thank my committee members Prof. Kenneth Jenkins, Prof. William Higgins, Prof. Jesse Barlow and Dr. Raja Bala. Their valuable comments and suggestions on my dissertation have helped improve my dissertation very significantly. I am especially thankful to Raja for being my research mentor in Xerox. He lead me into the world of industry and advised me on my future career development. I am honored to have him as a co-author on the every paper I wrote.

I am thankful to all the lab mates in iPAL: Umamahesh Srinivas, Bosung Kang, Mu Li, Yuelong Li, Chul Lee, Hojjat Mousavi, Tiep Vu and Pramod Krishna. I can not forget the days we spend together in the lab. I am also greatly indebted to my friends at Penn State including but not limited to Nan Hu, Xin Lu, Guiyuan Han, Xu Zhang, Chen Xiao, Xiaoyang Wu, Quan Gan, Wei Lu, Xingxing Meng, Yuhao Wang, Mengran Gou, Qiheng Cai, Yue Li, Wenqing Yao, Yingyue Tao, Anqi Wang. I am grateful for all these unforgettable moments spent with you.

Finally, I would like to thank my parents for all their love, support, encouragement and sacrifices. They unconditionally support very aspect of my life. No words are sufficient to describe my love to them. Without them, this dissertation would never have been written.
Chapter 1

Introduction

1.1 Motivation

With an increasing demand for security and safety, video-based surveillance systems are being increasingly used in urban locations. Vast amounts of video footage are collected and analyzed for traffic violations, accidents, crime, terrorism, vandalism, and other suspicious activities. Since manual analysis of such large volumes of data is prohibitively costly, there is a desire to develop effective algorithms that can aid in the automatic or semi-automatic interpretation and analysis of video data for surveillance and law enforcement. An active area of research within this domain is video anomaly detection, which refers to the problem of finding patterns in data that do not conform to expected behavior, and that may warrant special attention or action. The focus of this thesis is in the detection of anomalies in the transportation domain. Examples include traffic violations, unsafe driver and pedestrian behavior, accidents. Fig. 1.1 shows some examples of transportation related anomalies.

Video-based anomaly detection (AD) has received much recent attention. An
Figure 1.1. Examples of traffic-related anomalies: (a) unattended baggage; (b) car approaching a pedestrian; (c) crossing the yellow line; (d) running a red light.

An excellent overview of techniques can be found in [3]. Anomalies or outliers, are observations that do not conform to a well-defined notion of normal behavior. For example, if most observations associated with nominal behavior belong to some set, then observations outside (or sufficiently far away from) this set can be considered as outliers or anomalies. A statistical framework is often used to describe anomaly detection. In this setting, we are given features, $\ell$, in a suitably high-dimensional space. Feature instances are distributed with a probability density function (pdf), $g_0(\ell)$, if they come from a nominal distribution. Anomalous instances are distributed with pdf, $g_1(\ell)$. In the statistical framework, the anomaly detection problem amounts to predicting whether an instance, is distributed according to nominal or
anomalous pdf. Thus, the detection problem can be stated as follows:

\[
H_0 : \ell \sim g_0(\ell) : \text{nominal distribution}
\]
\[
H_1 : \ell \sim g_1(\ell) : \text{anomalous distribution}
\]  

(1.1)

If both pdfs are either known or can be estimated from training data this task reduces to the well-known likelihood ratio test.

1.2 Review of Related Work

The existing techniques to video anomaly detection can be broadly classified into supervised and unsupervised approaches. Fig. 1.2 shows the taxonomy of existing techniques.

\[\text{Anomaly Detection}\]

\[\text{Supervised approaches}\]

\[\text{Unsupervised approaches}\]

\[\text{Analysis-by-synthesis}\]

\[\text{Contextual and behavioral}\]

\[\text{Trajectory-based}\]

\textbf{Figure 1.2.} Taxonomy of existing techniques.
1.2.1 Supervised Approaches

In the supervised video anomaly detection problem, the collection of anomalies are assumed to be known. One first constructs a dictionary of anomalies, and then, for each observed video, one checks if a match can be found in the dictionary. From the statistical perspective, the problem reduces to a conventional classification problem in machine learning. Once a set of features $\ell$ is selected, pdfs under nominal and anomalous distribution can be estimated (or implicitly characterized) and, finally, an LRT (which usually can be reduced to the evaluation of a distance metric) can be applied to detect anomalous behavior. A ratio histogram approach is proposed by Chuang et al. [4] to represent object features, and suspicious events such as abandoned luggage are detected using a finite state machine. The main drawback of the supervised approach is that anomalous instances are far fewer compared to the normal instances in the training data, and obtaining representative instances of anomalies is generally difficult. Also, the method does not generalize to new anomalous patterns.

1.2.2 Unsupervised Approaches

In the past decade, several researchers have focused on an alternative unsupervised approach. Unsupervised approaches generally imply that labels are unknown. However, there is an implicit assumption that the number of nominal instances far outnumber the anomalous ones. For this reason, there is an overlap between unsupervised and semi-supervised approaches where the nominal instances are provided as a training set. In these approaches, the central aspect is the modeling of nominal activity either automatically or based on a training set of nominal video patterns.
1.2.2.1 Analysis-by-Synthesis Approaches

One alternative approach to track-based approaches is that proposed by Boiman and Irani [5]. This approach is based on spatiotemporal intensity correlation among different snippets of video. An observed sequence is built from spatiotemporal segments extracted from a training sequence. In this analysis-by-synthesis method, only regions that can be built from large contiguous chunks of the training data are considered normal. Instead of extracting moving features, Malinici et al. [6] extract features of the scene as a whole and build an infinite hidden Markov model on these features to identify anomalies. Methods by Yilmaz et al. [7] and Gorelick et al. [8] also use spatiotemporal chunks of video to characterize behavior. In this case, the silhouette of a moving object is concatenated into a three-dimensional (3-D) volume whose shape is used to recognize specific types of behavior. Unfortunately, these methods seem only effective when very few moving objects are seen simultaneously. Also, in many of these approaches, either a video clip or a large part of the field of view is treated as an integral entity. Consequently, the whole video clip or a large part of the field of view is labeled as either nominal or anomalous.

1.2.2.2 Contextual and Behavioral Approaches

These approaches model the contexts and behavioral attributes of an object rather than the object itself. Context in a video generally means the location and time of an object passing through the field of view. Behavioral attribute refers to the non-contextual attributes such as the size, speed, direction, and color of the object passing by a specific location.

These methods can work at a pixel level or more generally on larger blocks of pixels. Some methods use summary of motion vectors or motion labels, for
each location to describe activity in the scene. Consequently, an image-like 2-D structure provides a summary of activities over a large time window, thus easing memory and processing requirements. Xiang et al. [9] use a 7-D vector to represent each moving blob. An expectation maximization (EM) algorithm is then used to cluster these 7-D vectors into a pre-defined number of clusters. Events which cannot be clustered into any of these pre-defined clusters are regarded as anomalies. Saligrama et al. [3] propose a motion label representation to encode events with decisions facilitated via a two-state Markov chain model. Wang et al. [10] propose using hierarchical Bayesian models, where the video data is divided into so-called “documents” and events are subsequently encoded as quantized features, or ”words”, within these documents. Simon et al. [11] encode events via spatio-temporal volumes, and employ decision trees to identify events. At the core of many of these approaches is a departure from traditional object-based tracking perspective in favor of a new perspective that relies on location-based statistical modeling.

1.2.2.3 Trajectory-based Approaches

A significant effort has been devoted to video anomaly detection in surveillance applications over the last decade. A commonly used approach in the application of transportation is based on the clustering of the trajectories of the detected moving objects; the obtained clusters are then used as a normality model for anomaly detection. The work by Johnson and Hogg [12] was probably one of the first research in this direction, using vector quantization for the compact representation of trajectories and multi-layer neural networks for the identification of common patterns. After that, lots of work tries to use trajectories as features. Related techniques rely upon object tracking to detect nominal object trajectories and deviations thereof. This approach is appealing for traffic-related anomalies since
there are many state-of-the-art tracking techniques that can be leveraged [13]. A common approach is to derive nominal vehicle paths and look for deviations thereof in live traffic video data [1, 14–16]. During the test or evaluation phase, a vehicle is tracked and its trajectory compared against the nominal classes. A statistically significant deviation from all classes indicates an anomalous trajectory.

Since the advent of video analysis including anomaly detection, trajectory of a moving object has played an important role. A number of methods have been developed for learning two-dimensional (2-D) motion trajectories [1], resulting from tracking of objects or people [14]. Here, a large number of normal individuals or objects are tracked over time during the training phase. The resulting trajectories are then summarized by a set of motion trajectories, often translated into a symbolic representation of the background activity. In the detection phase, trajectories extracted from the monitored video are compared against those extracted in the training phase.

Tracking is generally performed by means of graphical state-based representations, such as hidden Markov models or Bayesian networks. Kumar et al. [17] propose a bayesian networks classifier to recognize different types of traffic behaviors. This classifier is built upon the trajectory features which are obtained from low-level image measurements. In [18], the authors model shape activities of objects by a Hidden Markov Model (HMM), and define anomalies as a change in the shape activity model. Makris and Ellis [19] label the scene with topological information, which is then used within a Bayesian approach to detect anomalous trajectories. Johnson and Hogg [12] quantize trajectories and develop a multi-layer neural network to identify anomalies. Hu et al. [20] use a hierarchic clustering of trajectories depending on spatial and temporal information to detect anomalies. A common characteristic of trajectory-based approaches [14–16] is the derivation
of nominal classes of object trajectories in a training phase, and the comparison of new test trajectories against the nominal classes in an evaluation phase. A statistically significant deviation from all classes indicates an anomalous trajectory. Trajectory-based methods perform well when the quality of tracking is high, and indeed one of the arguments in favor of trajectory-based approaches is the ability to leverage the recent significant advances in tracking techniques [13].

From a statistical perspective, the tracks amount to features \( \ell \) here and a nominal distribution of tracks, \( g_0(\ell) \) is obtained in the training phase. For the anomalous tracks, the implicit assumption is that the tracks are uniformly distributed over the set of all tracks. In this statistical perspective, the optimal detection rule is a thresholding strategy, whereby outliers with respect to the nominal tracks are declared abnormal. In particular, one can define a negative log-likelihood function as follows:

\[
\Lambda(\ell) = -\log(g_0(\ell)) \quad \text{anomalous} \quad \gtrless \quad \tau. \quad (1.2)
\]

The state of the art trajectory-based video anomaly detection technique is proposed by Piciarelli et al. [1]. They use one class SVMs for anomaly detection by utilizing trajectory information as features.

In their method, trajectories are represented using 8 pairs of \( x \) and \( y \) coordinates, thus leading to feature vectors composed of 16 elements. Then, these trajectories are clustered with a one-class SVM. During the training phase, a classification hyperplane is learned. Fig. 1.3 shows an illustration of their method on \( 2 - D \) data. Based on the classification hyperplane, an outlier detection technique can be used to detect anomalies. If the angle \( \theta_X \) between a test trajectory \( X \) and the center \( C \) greater than a threshold, this test trajectory \( X \) is regarded as an
anomaly:

$$\theta_X > \theta_{th} \rightarrow X : \textbf{outlier},$$

(1.3)

Although there are advantages to using trajectories as motion features, there are clear disadvantages as well. First, tracking is a difficult task, especially in real time and in urban scenarios where often a large number of objects are present. Since the anomaly detection is directly related to the quality of tracking, a tracking error will inevitably bias the detection step. Second, since each individual or object monitored is related to a single trajectory, it is hard to deal with people occluding each other.

1.2.3 Open Problems

Video anomaly detection is generally difficult because of the following issues:

- **Noise and occlusion:** noise is very common in surveillance videos as shown in Fig. 1.4, where we may lose some important information. In addition, because of the limitation of camera’s visual angle, occlusions often occur in video data. Figs. 1.5 shows an example of occlusion, where a car is
occluded by another car. Noise and occlusions existing in the raw video clips make the anomaly detection to be a difficult task.

- **Multiple object interaction:** a variety of anomaly detection algorithms have been designed for video surveillance. However only few of them have considered the interaction between multiple objects. In real world scenario, “collective anomalies” that are caused by the joint observation of objects are also significant. For example, in the area of transportation, some events, e.g. accidents and dangerous driver-pedestrian behavior, are indeed based on joint and not just individual object behavior.

- **Unsupervised video anomaly detection:** most existing work can only deal with supervised anomaly detection where representative training for anomalous events is available. In real word applications, it is often not possible to gather a sufficiently large number of training samples representing anomalous events.

- **Multiple Event encoding:** most video anomaly detection techniques use only single event encoding, thus fail to exploit diversity in these multiple per-
Figure 1.5. An example of video occlusion

spectives on an event. It is not difficult to see that the distinct event feature representations contain *correlated yet complementary* information about the event (normal or anomalous).

- **Unstructured scenario:** Existing methods always can only handle structured scenario. Fig. 1.6(a) shows an example video frame of a structured scenario (detection of stop sign violations) where preparation of a dictionary clearly separated into class-specific sub-dictionaries is possible. In many other settings however, multiple objects and features are simultaneously extracted and a clear separation into normal event classes is difficult. An example of a video frame from such a scenario is shown in Fig. 1.6(b).
1.3 Dissertation Contributions and Organization

A snapshot of the main contributions of this dissertation is presented next. Publications related to the contribution in each chapter are also listed where applicable.

Chapter 2 discusses the background of sparsity model and introduces two most recent research [21], [22] which use sparsity model for the problem of video anomaly detection. The fundamental underlying assumption of these methods is that any new feature representation of a normal/anomalous event can be approximately modeled as a (sparse) linear combination pre-labeled feature representations (of previously observed events) in a training dictionary. Li et al. use object trajectories while Li et al. use spatiotemporal volumes. We choose sparsity model for the problem of video anomaly detection because of its robustness to noise and occlusion. This entire dissertation will focus on developing better sparsity models which is more suitable for our video anomaly detection problem.

In Chapter 3, we propose a novel and general trajectory-based joint sparse reconstruction framework that can effectively detect both single-object and multiple-
object anomalies. Additionally, a suitable outlier rejection measure is developed for the multiple-object case that obviates the need to build anomalous event classes, and enables unsupervised anomaly detection with high accuracy (note, labeled training for normal events is still assumed available). For single object anomaly detection, recent work in [21], [22] can be seen as special cases of our framework when matching event representations and dictionary building techniques are used. Experimental results on real and synthetic data demonstrate the effectiveness of our approach for both single-object and multi-object anomalies. This material was presented at the 2012 IEEE Asilomar Conference on Signals, Systems and Computers [23] and was published in the IEEE Transactions on Circuits and Systems for Video Technology [24].

Our second contribution presented in Chapter 4 is towards improving the accuracy and reducing the complexity of the sparsity models that are employed for representing and classifying trajectories. First, we introduce non-linearity into, i.e. kernelize the linear sparsity model to enable superior class separability and hence anomaly detection. Current sparsity based trajectory representation assumes a linear reconstruction model. This may not be true in practice. If the data set does not obey linear models, kernel methods that are popular in learning, can be applied to project the data into a high-dimensional nonlinear feature space in which the data becomes more linearly separable [25–27]. In practical implementation, the kernel trick is often used in order to avoid explicitly evaluating the data in the feature space [28]. Kernel orthogonal and basis pursuit [29, 30] algorithms and their applications [31] have been of much recent interest. We show that such kernelization naturally extends to the joint sparsity model as well and can be used to improve multi-object anomaly detection. Second, we develop a dictionary design and optimization technique that can effectively reduce the size of training
dictionaries that enable sparsity based classification/anomaly detection without adversely influencing detection performance. Experimental results show that significant computational advantages can be obtained with the proposed techniques with little performance loss over using large and manually labeled dictionaries of example trajectories.

These work was presented at the 2013 IEEE International Conference on Intelligent Transportation Systems (ITSC) [32] and was published in the IEEE Transactions on Circuits and Systems for Video Technology [24].

Chapter 5 presents our third contribution exploiting the diversity in multiple perspectives on an event. First, in structured scenario, we propose to extend sparsity models based on single feature representations to simultaneous sparse representations of multiple feature representations. In this model, the matrix of sparse coefficients does not confirm to the commonly seen row-sparsity and a modified greedy heuristic approach that extends simultaneous orthogonal matching pursuit (SOMP) is needed to solve the resulting optimization problem. This word was accepted by 2014 IEEE International Conference on Image Processing and will be submitted to IEEE Transactions on Circuits and Systems for Video Technology.

In unstructured scenario, we advocate a more general and practical sparsity model using a low-rank structure on the matrix of sparse coefficients. In Chapter 6, we find that enforcing a low-rank structure can ease the rigidity of traditional row-sparse constraints on sparse coefficient vectors/matrices. Because low-rank matrices are of course not always sparse, an additional $\ell_1$ regularization term is added. Further, if rank is substituted by its convex nuclear norm alternative, then significant computational benefits can be obtained over existing methods in sparsity based video anomaly detection. This work was presented at the 2014 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)
[33] and will be submitted to IEEE Transactions on Circuits and Systems for Video Technology.
Chapter 2

Sparse Representations for Video Anomaly Detection

2.1 Introduction

It is well-known that a large class of signals, including audio and images, can be expressed naturally in a compact manner with respect to well-chosen basis representations. Among the most widely applicable of such basis representations are the Fourier and wavelet bases. This has inspired a proliferation of applications that involve sparse signal representations for acquisition [34], compression [35], and modeling [36]. The central problem in compressive sensing (CS) is to recover a signal $x \in \mathbb{R}^n$ given a vector of linear measurements $y \in \mathbb{R}^m$ of the form $y = Ax$, where $m \ll n$. Assuming $x$ is compressible, it can be recovered from this underdetermined system of equations by solving the following problem [34]:

$$\min_x \|x\|_0 \text{ subject to } y = Ax,$$  \hspace{1cm} (2.1)
where $\|x\|_0$ is the $l_0$-"norm" that counts the number of non-zero entries in $x$. Under certain conditions, the $l_0$-norm can be approximated by the $l_1$-norm, leading to convex optimization formulations.

Although the concept of sparsity was introduced to solve inverse reconstructive problems, where it acts as a strong prior to the abbreviated ill-posed nature of the problems, recent work [37, 38] has demonstrated the effectiveness of sparse representation in classification applications too. The crucial observation is that a test image can be reasonably approximated as a linear combination of training images belonging to the same class, with (ideally) no contributions from training images of other classes. Therefore, with $A := [A_1 \ldots A_K]$ and $A_i$ representing the matrix of vectorized training images from the $i$-th class, the corresponding coefficient vector $x := [x_1^T \ldots x_K^T]^T$ is sparse and naturally encodes discriminative information. In other words, the semantic information of the signal of interest is often captured in the sparse representation. Albeit simplistic in formulation, this linear sparse representation model is rooted in well-understood optimization theory. The sparse representations exhibit robustness to a variety of real-world image distortions, leading to their widespread use in applications such as face recognition, remote sensing, and medical image classification for disease diagnosis.

### 2.2 Existing Sparsity-based Techniques for Video Anomaly Detection

A significant challenge in trajectory based video anomaly detection is the ability to handle occlusions amongst moving objects and their trajectories. Recently sparse reconstruction techniques have been applied to face recognition and demonstrated
to effectively deal with occlusions [37]. Inspired by this result, Li et al. applied sparse reconstruction towards the anomaly detection problem [21]. Object trajectories are extracted from video using state-of-the art tracking algorithms, encoded via a Least squares Cubic Spline Curves Approximation (LCSCA), and collected into event classes to form a training dictionary. The fundamental underlying assumption is that any new trajectory can be approximately modeled as a (sparse) linear combination of trajectories in the training dictionary. Zhao et al. [22] also adopt a sparsity based approach for anomaly detection, but instead use spatio-temporal volumes as event representations. The numerical value of an objective function, which is a regularized version of the sparse representation based reconstruction error [22] is used to determine if the spatio-temporal volume encoding of a new event is normal or otherwise.

2.2.1 Abnormal Behavior Detection via Sparse Reconstruction Analysis of Trajectory

Abnormal behavior detection via sparse reconstruction analysis of trajectory [21] is a recent novel and promising idea in the field of video anomaly detection. Trajectories of object are extracted from video by traditional object tracking algorithms. Each trajectory is then represented as a feature vector by a Least-squares Cubic Spline Curves Approximation (LCSCA) representation [21]. The fundamental underlying assumption in [21] is that any new trajectory can approximately be modeled as a (sparse) linear combination of training trajectories (or equivalent features). Let each trajectory representation lie in $\mathbb{R}^n$, and $T$ denote the number of training samples (i.e. example trajectory representations) from each of $K$ different classes, i.e. behavior patterns in a video which may be normal or anom-
Figure 2.1. An example illustration of trajectory classification using a sparse reconstruction model. The training dictionary consist of 2 classes (red trajectories are class 1, blue trajectories are class 2), each class contains 4 different trajectories. The test trajectory (the green trajectory) can be well represented by the linear combination of the first trajectory and the third trajectory from class 1.

The training samples (trajectory representations) from the \( i \)-th class are arranged as the columns of a matrix \( A_i \in \mathbb{R}^{n \times T} \). The dictionary \( A \in \mathbb{R}^{n \times KT} \) of training samples from all classes is then formed as follows: \( A = [A_1 \ A_2 \ldots \ A_K] \).

Given a sufficient number of training samples from the \( m \)-th trajectory class, a test image \( y \in \mathbb{R}^n \) from the same class is conjectured to approximately lie in the linear span of those training samples. Any trajectory feature vector is synthesized by a linear combination of the set of all training trajectory samples as follows:

\[
y \approx A \alpha = [A_1 \ A_2 \ldots \ A_K] \begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_K
\end{bmatrix},
\]

(2.2)
where each $\alpha_i \in \mathbb{R}^T$. Typically for an example trajectory $y$, only one of the $\alpha_i$’s will be active (corresponding to the class/event from which $y$ is generated). Thus the coefficient vector $\alpha \in \mathbb{R}^{KT}$ is modeled as sparse and is recovered by solving the following optimization problem:

$$\hat{\alpha} = \arg \min_{\alpha} \|\alpha\|_1 \text{ subject to } \|y - A\alpha\|_F \leq \epsilon,$$  \hfill (2.3)

where the objective is to minimize the number of non-zero elements in $\alpha$. It is well-known from the compressed sensing literature that minimizing the $l_0$ norm leads to an NP-hard problem [39]. Thus the $l_1$ norm is used as an effective substitute.

The residual error between the test trajectory and each class behavior pattern is computed to find the class to which the test trajectory belongs:

$$r_i(y) = \|y - A_i\hat{\alpha}_i\|_F \quad i = 1, 2, \ldots, K$$  \hfill (2.4)

Fig. 2.1 shows an example of classification using sparsity model. The training dictionary consist of 2 classes, each class contains 4 different trajectories. The test trajectory can be well represented by the linear combination of trajectory no. 1 and trajectory no. 3 from class 1 (see Fig. 2.1). This is in fact tantamount to saying that the coefficient vector $\alpha$ is indeed sparse - in this example, two of eight entries being active.

We advocate that sparsity model is also more flexible in the field of anomaly detection. Fig. 2.1 shows a toy example of classification. In this example, a test trajectory is put into a classifier where the training dictionary only has 2 classes and each class only contains one trajectory. If using SVM-based method, the test trajectory will largely be identified as anomaly, since it’s not reasonable to classify
the test trajectory into any of the training classes. Let $Tr_1$ refer to the training trajectory 1, $Tr_2$ to be the training trajectory 2 and $Te$ denotes the test trajectory. Using sparsity model, $Te \approx 0.5 \times Tr_1 + 0.5 \times Tr_2$. Outlier rejection measure in Eq. (3.16) will identify this test trajectory as anomaly, if the threshold $\tau_1 > 0.5$. Otherwise, the test trajectory will be regarded as normal event. Therefore, the sparsity model shows more flexibility with respect to different requirements.

Based on the aforementioned model, Li et al. [21] identify anomalies by apriori defining normal and anomalous event classes in video. Their advocacy of sparsity over other trajectory classification techniques viz. one-class SVMs [1] is based on two arguments: 1.) recent work in face recognition [37] has shown that sparsity based classification can be powerful even as feature descriptions are missing, e.g. occlusion of objects leading to missing trajectory information, and 2.) the sparse coefficients can additionally withstand noise and quality distortions to the video.

### 2.2.2 Online Detection of Unusual Events in Videos via Dynamic Sparse Coding

Zhao et al. [22] propose an online sparsity-based method for video anomaly detection. They use sparse linear combinations of spatio-temporal volumes under an $l_1$ sparsity model. But instead of using a fixed training dictionary and only solving for the sparse coefficients, they employ a principled convex optimization formulation that allows both a sparse reconstruction code, and an online dictionary to be jointly inferred and updated:

$$
(\alpha_1^*, ..., \alpha_m^*, A^*) = \arg \min_{\alpha_1, ..., \alpha_m, A} \sum_{i=1}^{m} J(y_i, \alpha_i, A),
$$

(2.5)
where $J(y_i, \alpha_i, A)$ is the objective function that measures how normal an event is. It includes reconstruction error, sparsity regularization using $l_1$ norm and smoothness terms. Subsequent to the optimization, the dictionary $A$ is augmented using newly observed events.

For anomaly detection, they define a threshold $\hat{\epsilon}$ that controls the sensitivity of the algorithm to anomalous events. A test spatio-temporal volume $y'$ will be detected as an anomalous event if the following criterion is satisfied:

$$J(y', \alpha', A^*) > \hat{\epsilon}$$ (2.6)

Both [21] and [22] show promise for the use of sparsity in video anomaly detection but exhibit two important limitations. First, they only address anomalies involving single objects. While such events arguably account for a large proportion of anomalies, there are important scenarios wherein the anomaly arises from an interaction among multiple objects. Consider for example, two vehicles following nominal individual trajectories but approaching within a dangerously close vicinity of each other. The second limitation associated particularly with [21] is that the anomalous events must be characterized $a$ priori into their own classes, i.e. supervised anomaly detection where representative training for anomalous events is available. In real world applications, it is often not possible to gather a sufficiently large number of training samples representing anomalous events.

### 2.3 Conclusion

In this chapter, we have concluded the existing sparsity based video anomaly detection techniques Li et al. and Zhao et al. We choose sparsity model for the problem
of video anomaly detection because of its robustness to noise and occlusion (Some result about sparsity under object occlusion or missing trajectory information are shown in Chapter 4). As discussed above, there are still some drawbacks in Li et al. and Zhao et al. Therefore, we have to improve the existing sparsity model.

Sparsity is a powerful prior in this model because of multiple reasons: a.) like in [21] and [22], when a new collection of multi-object trajectories manifests, it is expected to invoke only a few columns of the training dictionary that combine to create it, and b.) even more crucially object-wise correspondence is important in the linear combination for this model to physically meaningful leading to a (non-standard) block-diagonal sparse structure on coefficients detailed later in Chapter 3, c.) Finally, we observe that while the sparse structure conveys information about normal/anomalous event classes - in the absence of training data for anomalous events we can develop and use outlier rejection measures on the sparse coefficient matrix that can help with multi-object anomaly detection in unsupervised settings - a very challenging problem.
Joint Sparsity Model for Trajectory Based Video Anomaly Detection

3.1 Introduction

In this chapter, we propose a novel and general trajectory based joint sparse reconstruction framework for video anomaly detection. Our goal is to build a linear model where joint trajectory representations of multiple objects are written as linear combinations of corresponding joint trajectories in a training dictionary. Trajectories have long been popular in video analysis and anomaly detection [12], [17], [19], [20]. A common characteristic of trajectory-based approaches [1, 14–16] is the derivation of nominal classes of object trajectories in a training phase, and the comparison of new test trajectories against the nominal classes in an evaluation phase. A statistically significant deviation from all classes indicates an anomaly.

We must emphasize that our choice of trajectories (as opposed to spatio-temporal volumes for example in [22]) as the event encoder is motivated by two
principal reasons: 1.) interactions between multiple objects are quite naturally captured in trajectory representations, e.g. vehicles approaching within a dangerously close vicinity of each other can be caught, and 2.) recent advances in object tracking ensure trajectory extraction is both fast and reliable [13]. Nevertheless, in theory any event representation can be used with our model.

3.2 Observations and Motivation for Joint Sparsity Model

A variety of anomaly detection algorithms have been designed for video surveillance. However only a few of them have considered the interaction between multiple objects [2, 10, 18]. Among them, only Han et al. [2] use trajectory as features. They propose a multiple object tracking algorithm and corresponding rule-based anomaly detection approach. In their tracking algorithm, Each object is identified by an index $i$ and its state at time $t$ is represented by:

$$x_t^i = (p_t^i, v_t^i, a_t^i, s_t^i)$$  \hspace{1cm} (3.1)

where $p_t^i$ is the image location, $v_t^i$ represents the 2D velocity, $a_t^i$ and $s_t^i$ denote the appearance and scale of object $i$ at time $t$, respectively. $p_t^i$ and $v_t^i$ use continuous image coordinates. Then, they use a Hidden Markov Model as the probabilistic model to maximize the joint probability between the state sequence and the observation sequence.

While it is true that anomalies are generated by atypical trajectory/behavior of a single object, “collective anomalies” that are caused by the joint observation of objects are also significant. For example, in the area of transportation, some events,
e.g. accidents and dangerous driver-pedestrian behavior, are indeed based on joint and not just individual object behavior. It is possible in fact that the individual events corresponding to each object’s behavior are not necessarily anomalies by themselves. Take the example of a vehicle accidentally changing lanes due to an inattentive driver. Another vehicle in close proximity may have to also suddenly change lanes in order to avoid colliding with the first vehicle. Both lane changes, as isolated events are not necessarily anomalous, but when viewed in conjunction should logically be flagged as a joint anomaly.

Previous methods have employed probabilistic models to learn the relationship between different individual events. Han et al. [2] and Vaswani et al. [18] use an HMM-based method to track multiple trajectories followed by defining a set of rules to distinguish between normal and anomalous events. Wang et al. [10] present an unsupervised framework using hierarchical Bayesian models to model individual events and interactions between them.

For anomaly detection, they first collect the information from tracking result including the number of objects, their motion history and interaction, the timing of their behaviors. Then they interpret events based on the basic information about WHO (how many) participated into that event, WHEN the event occurred, WHERE the event happened and WHAT is the details of the event. Based on this interpretation, anomaly can be defined by some rules. For example, in a traffic intersection scenario, there are always 0 - 8 cars around. If 15 cars arrive this intersection simultaneously, it can be regarded as an anomalous event.

The sparsity based approach reviewed in Chapter 2 while powerful, does not capture interactions to detect 2 or more object anomalies. We describe next a new “joint sparsity model” for video anomaly detection which incorporates multiple object trajectories and their interactions. Hence even if individually the trajec-
tories may be considered normal, "collective anomalies" could occur and can be successfully detected in our proposed framework.

3.3 Joint Sparsity Model for Trajectory-based Video Anomaly Detection

3.3.1 Notation

We are interested in detection anomalies involving $P \geq 1$ objects. Their corresponding $P$ trajectories can be represented as a matrix: $Y = [y_1 \ y_2 \ \ldots \ y_P] \in \mathbb{R}^{n \times P}$, where $y_i$ correspond to $i^{th}$ trajectory. The training dictionary can be defined as: $A = [A_1 \ A_2 \ \ldots \ A_P] \in \mathbb{R}^{n \times PKT}$, where each dictionary $A_i = [A_{i,1} \ A_{i,2} \ \ldots \ A_{i,K}] \in \mathbb{R}^{n \times KT}, i = 1, 2, \ldots, P$, is formed by the concatenation of the sub-dictionaries from all classes belonging to the $i^{th}$ trajectory. The crucial aspect of this formulation is that the training trajectories for any class $j$, i.e. $A_{i,j}, i = 1, 2, \ldots, P$ are observed “jointly” from example videos. This generalizes the set-up of [21], [22].

3.3.2 Joint Sparsity Model

The test $P$ trajectories can now be represented as a linear combination of training samples as follows:

$$Y \approx AS$$

$$= [A_{1,1} \ A_{1,2} \ \ldots \ A_{1,K} \ \ldots \ A_{P,1} \ A_{P,2} \ \ldots \ A_{P,K}][\alpha_1 \ \ldots \ \alpha_P],$$  \hspace{1cm} (3.2)

where the coefficient vectors $\alpha_i$ lie in $\mathbb{R}^{PKT}$ and $S = [\alpha_1 \ \ldots \ \alpha_i \ \ldots \ \alpha_P]$. 
It is important to note that the $i$-th object trajectory of any observed set of test trajectories should *only lie* in the span of training trajectories corresponding to the $i$-th object. Therefore, the columns of $S$ should have the following structure:

\[
\alpha_1 = \begin{bmatrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \vdots \\ \alpha_{1,K} \\ 0 \\ 0 \end{bmatrix}, \quad \alpha_i = \begin{bmatrix} 0 \\ \alpha_{i,1} \\ \alpha_{i,2} \\ \vdots \\ \alpha_{i,K} \\ 0 \end{bmatrix}, \quad \alpha_P = \begin{bmatrix} 0 \\ \alpha_{P,1} \\ \alpha_{P,2} \\ \vdots \\ \alpha_{P,K} \end{bmatrix}.
\] (3.3)

where each of the sub-vectors $\{\alpha_{i,j}\}_{j=1}^{K}$, $i = 1, 2, \ldots, P$ lies in $\mathbb{R}^T$, while $0$ denotes a vector of all zeros in $\mathbb{R}^{KT}$. As a result, $S$ exhibits a block-diagonal structure.

From [21], we know that for a single object, its trajectory can be represented by a sparse linear combination of all the training samples. For the multiple trajectories scenario, we assume that training samples with non-zero weights (in the sparse linear combination) exhibit one-one correspondence across different trajectories. In other words, if the $i$-th trajectory training sample from the $j$-th class is chosen for the $i$-th test trajectory, then it is necessarily that other $P-1$ trajectories choose from $j$-th class with very high probability, albeit with possibly different weights.

We take a simple scenario which only has 2 objects and 2 training classes (normal and anomalous class) as an example to explain the structure of Eq. (3.2).
In this situation, $P = 2$, $K = 2$, Eq. (3.2) becomes:

$$Y \approx AS = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} \alpha_{1,1} & 0 \\ \alpha_{1,2} & 0 \\ 0 & \alpha_{2,1} \\ 0 & \alpha_{2,2} \end{bmatrix},$$

(3.4)

The test trajectory sample is thought of as a collective event. Therefore, all trajectories of the sample should be classified into one class. If the 1-st trajectory is classified into $j$-th class, the 2-nd trajectory should also be classified into $j$-th class, which means $\alpha_{1,j}$ and $\alpha_{2,j}$ should be activated simultaneously. This characteristic that some coefficients should be activated jointly captures the interaction between objects.

Moreover, we only care about the non-zero element in the matrix $S$. Define a new matrix $S'$:

$$S' = \begin{bmatrix} \alpha_{1,1} & \alpha_{2,1} \\ \alpha_{1,2} & \alpha_{2,2} \end{bmatrix},$$

(3.5)

In the structure of $S'$, “joint coefficients” are moved into the same row. The joint information can be captured by enforcing certain entire rows of $S'$ to be activated simultaneously.

In general, when there are $K$ classes and $P$ objects, the structure of $S'$ is:

$$S' = \begin{bmatrix} \alpha_{1,1} & \ldots & \alpha_{i,1} & \ldots & \alpha_{P,1} \\ \alpha_{1,2} & \ldots & \alpha_{i,2} & \ldots & \alpha_{P,2} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha_{1,K} & \ldots & \alpha_{i,K} & \ldots & \alpha_{P,K} \end{bmatrix} \in \mathbb{R}^{KT \times P}.$$

(3.6)
The question that remains to be addressed is the particular way of transforming \( S \). Such a transformation is realized by defining matrices \( H \in \mathbb{R}^{PKT \times P} \) and \( J \in \mathbb{R}^{KT \times PKT} \):

\[
H = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}, \quad J = \begin{bmatrix} I_{KT} & I_{KT} & \ldots & I_{KT} \end{bmatrix}.
\]  

(3.7)

The vectors \( \mathbf{1} \) and \( \mathbf{0} \) are in \( \mathbb{R}^{KT} \) and contain all ones and zeros respectively, and \( I_{KT} \) is the \( KT \)-dimensional identity matrix. Finally, we have:

\[
S' = J (H \circ S),
\]

(3.8)

where the \( \circ \) indicates matrix Hadamard (entry-wise) product.

Therefore, we can now solve for the sparse coefficients via the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \| J (H \circ S) \|_{\text{row,0}} \\
\text{subject to} & \quad \| Y - AS \|_F \leq \epsilon,
\end{align*}
\]

(3.9)

where \( \| \cdot \|_{\text{row,0}} \) refers to the number of non-zero rows in the matrix and the cost function minimization seeks the \( J (H \circ S) \) with the minimum number of non-zero rows, while the constraint ensures good approximation (\( \| \cdot \|_F \) denotes the Frobenius norm).

The well known row sparsity problem:

\[
\begin{align*}
\text{minimize} & \quad \| S \|_{\text{row,0}} \\
\text{subject to} & \quad \| Y - AS \|_F \leq \epsilon,
\end{align*}
\]

(3.10)
Algorithm 1 SOMP

\textbf{Input:} Dictionary $A = [a_1 \ a_2 \ldots \ a_{P\times PKT}]$, data matrix $Y = [y_1 \ y_2 \ldots \ y_P]$, a stopping criterion \{Make sure all columns in $A$ and $Y$ have unit norm\}

Initialization: residual $R_0 = Y$, index set $\Lambda_0$: empty set, iteration counter $k = 1$

1: \textbf{while} stopping criterion has not been met \textbf{do}

1. Find the index of the atom that best approximates all residuals: $\lambda_k = \arg \max_i || R_{k-1}^T a_i ||_2$

2. Update the index set $\Lambda_k = \Lambda_{k-1} \cup \{\lambda_k\}$

3. Compute $G_k = (A^T_{\Lambda_k} A_{\Lambda_k})^{-1} A^T_{\Lambda_k} Y$, $A^T_{\Lambda_k}$ consists of the $k$ atoms in $A$ indexed in $\Lambda_k$

4. Determine the residual $R_k = Y - A^T_{\Lambda_k} G_k$

5. $k \leftarrow k + 1$

2: \textbf{end while}

\textbf{Output:} Index set $\Lambda = \Lambda_{k-1}$, the sparse representation $S$ whose nonzero rows indexed by $\Lambda$ are $k$ rows of the matrix $(A^T_{\Lambda} A_{\Lambda})^{-1} A^T_{\Lambda} Y$

is non-convex but can be solved using greedy pursuit algorithms widely used in the literature. Simultaneous Orthogonal Matching Pursuit (SOMP) [40] - enumerated in Algorithm 1 - is amongst the most popular algorithms used. In SOMP, the support of the solution is sequentially updated (i.e., the atoms in the dictionary $A$ are sequentially selected). At each iteration, the atom that simultaneously yields the best approximation to all of the residual vectors is selected.

$$\lambda_k = \arg \max_i || R_{k-1}^T a_i ||_2$$ \hspace{1cm} (3.11)

Our proposed joint sparsity model for representing multiple object trajectories involves solving Eq. (3.9), which looks quite similar to Eq. (3.10) but the Hadamard operator from $S$ to $S'$ makes the problem much more involved. We can observe from Algorithm 1 that the original SOMP algorithm effectively gives $k_0$ distinct atoms from a dictionary $A$ that best approximates the data matrix $Y$ for $k_0$ iterations,
we apply the general formulation even when the Hadamard operator is present. At every iteration $k$, SOMP measures the residual for each atom in $\mathbf{A}$ and creates an orthogonal projection with the highest correlation.

This idea can be extended to our proposed joint sparsity setting. If the atom of $j$-th trajectory we selected comes from $i$-th training, the other $P - 1$ atoms of trajectories should be also chosen from $i$-th training. Then Eq. (3.11) in SOMP can be modified as:

$$
\lambda_k = \arg \max_i \sum_j \| \mathbf{R}_{j,k-1}^T \mathbf{a}_{j,i} \|_2
$$

(3.12)

where $\mathbf{R}_{j,k-1}^T$ refers to the residual of $j$-th trajectory in iteration $k - 1$, and $\mathbf{a}_{j,i}$ represents the $i$-th training of $j$-th trajectory. After employing this special rule of choice for atom selection, each row of parameter matrix $\mathbf{S}'$ will be activated simultaneously or inactivated simultaneously, thus the row sparsity requirements will inherently hold. The implementation details of this algorithm can be found in a technical report [41].

3.3.3 Auxiliary Optimization Problem

The problem in (3.9) is non-convex, and the solution obtained using any known optimization solvers is sub-optimal. To improve the solution, we propose an auxiliary optimization problem which is in fact convex, with guarantee of global minima.

From the optimization problem Eq. (3.9), we can first get a sub-optimal result: $\hat{\mathbf{S}}$. We then design a *membership matrix* $\mathbf{E} \in \mathbb{R}^{PKT \times P}$ which has zeros at locations of non-zero entries in $\hat{\mathbf{S}}$ and ones elsewhere. With this enforcement of locations of
non-zero entries, we have the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \|Y - AS\|_F \\
\text{subject to} & \quad s_i^T e_i = 0, i = 1, 2, \ldots, P
\end{align*}
\]

(3.13)

where \(s_i\) and \(e_i\) refer to the \(i\)-th column of \(S\) and \(E\) respectively. The initial choice of \(S\) is the \(\hat{S}\) described previously.

This problem can be further simplified to mitigate computational complexity. Each column of \(S\) can be optimized in parallel, since the constraints are separable. So we have:

\[
\begin{align*}
\text{minimize} & \quad \|y_i - As_i\|_2 \\
\text{subject to} & \quad s_i^T e_i = 0,
\end{align*}
\]

(3.14)

3.3.4 Classification into Event Classes

Having obtained the sparse coefficient matrix \(S\), we compute class-specific residual errors and identify the class of the test event \(Y\) as that which gives the minimum residual:

\[
\text{identity}(Y) = \arg \min_i \|Y - A\delta_i(S)\|_F,
\]

(3.15)

where \(\delta_i(S)\) is the matrix whose only nonzero entries are the same as those in \(S\) associated with class \(i\) (in all \(P\) trajectories). When sufficient representation (example training trajectories) for anomalous events is available, then anomalous classes are simply one or more of the \(K\) classes in this joint sparsity based classification framework.
3.3.5 Anomaly Detection via Outlier Rejection

If training for anomalies is missing/statistically insignificant, we can not use (3.15) to identify anomalies. Inspired by the outlier rejection measure in [37]:

\[
SCI(\alpha) = K \cdot \frac{\max_i \|\rho_i(\alpha)\|_1/\|\alpha\|_1 - 1}{K - 1}, \quad SCI(\alpha) < \tau_1 \rightarrow y : \text{outlier}, \quad (3.16)
\]

and \(\rho_i(\alpha)\) is the new vector whose only nonzero entries are the entries in \(\alpha\) that are associated with class \(i\). we model anomalies as outliers, given training from expected normal event classes that form the dictionary \(A\). Eq. (3.16) can be used to detect single object anomalies. We extend it to the multiple object case:

\[
JSCI(S) = K \cdot \max_i \|\delta_i(S')\|_{row,0}/\|S'\|_{row,0} - 1, \quad (3.17)
\]

where JSCI is the Joint-SCI. If \(JSCI(S) < \tau_2\), a multiple object anomaly is identified.

![Figure 3.1. Trajectory extraction: (a) background subtraction and blob analysis to identify objects, (b) blob centroid calculation and trajectory derivation.](image)
3.4 Experimental Results

3.4.1 Trajectory Extraction

Trajectory extraction is accomplished using well-known techniques. First background subtraction is accomplished via the use of a Gaussian Mixture Model (G-MM) [42]. In order to eliminate the effect of noise, blob analysis is then used here to identify the location of the moving vehicle. We calculate the number of connected foreground pixels, and deem the connected segment to be a vehicle if this exceeds a threshold. As seen in Fig. 3.1(a), the car is successfully detected by this technique. Next, we calculate and track the centroid of the blob over time in order to obtain the object trajectory. Fig. 3.1(b) shows an example of the extracted trajectory, which is represented mathematically as a coordinate pair \([x(t), y(t)]\). Li et al. [21] use a LCSCA (Least-squares Cubic Spline Curves Approximation) representation of trajectories. We first approximate a raw trajectory using a basic B-spline function [43] with 50 knots (50 x-coordinates and 50 y-coordinates) and these knots are extracted to represent the trajectory.

3.4.2 Video Datasets

As discussed above, if an anomaly is generated by single object, we can call it single-object anomaly. Figs. 3.2 (a) and (b) show consecutive frames from 2 examples of single-object anomalies. In Fig. 3.2 (a), a man suddenly falls on the floor (see Fig. 3.2(a2)) when walking across the lobby. In Fig. 3.2 (b), instead of turning left or right in front of the stop sign, the driver suddenly backs his car - see Figs. 3.2 (b3)-(b4). By setting the number of objects \(P = 1\), our joint sparsity model reduces to the model by Li et al [21] and can be used to detect anomalous trajectories of
Figure 3.2. Example frames of single-object anomalies: (a) a man suddenly falls on floor – from the CAVIAR data set and (b) a driver backs his car in front of stop sign – from the Xerox Stop Sign data set.

On the other hand, if an anomaly happens via the interaction of multiple objects, we call it a multiple-object anomaly. The video clip frames from 3 examples of multiple-object anomalies are shown in Figs. 3.3 (a), (b) and (c). In Fig. 3.3 (a), a pedestrian walk crossing the street loses his hat and retraces his footsteps to pick it up from the road. At this time, a vehicle comes in very close proximity to the pedestrian and comes to a sudden halt - see Fig. 3.3 (a2). In Fig. 3.3 (b), the second vehicle (marked by a red rectangle) comes to a complete stop when waiting for the vehicle in front of it (Fig. 3.3 (b1)), but does not actually stop at the stop sign - see Figs. 3.3 (b2)-(b3). In Fig. 3.3 (c), a car fails to yield to oncoming car while turning left - see Figs. 3.3 (c2)-(c3). The examples in Fig. 3.3 (a) and (b) are in fact from a real-world transportation database (which cannot be made public for proprietary reasons) which we refer to as the Xerox Stop Sign database. An example video clip is however made available at: http://youtu.be/M6_PJigg5CY. And the example in Fig. 3.3 (c) comes from another proprietary transportation
Figure 3.3. Example frames of multiple-object anomalies: (a) a vehicle almost hits a pedestrian – from the AVSS data set, (b) a car (marked by red rectangle) violates the stop sign rule – from the Xerox Stop Sign data set and (c) a car fails to yield to oncoming car while turning left – from the Xerox Intersection data set.

Figure 3.4. Synthetic trajectories: (a) three different paths for a car on intersection, (b) examples of normal training class, (c) anomaly type 1: running traffic light, (d) anomaly type 2: turning left fail to yield on coming traffic.

database which we address as the Xerox Intersection database. A representative video clip is made available at: http://youtu.be/ZGKtkVtWEFU.

Because sufficient training for multi-object anomalies in particular is rarely available, we additionally also generate some synthetic trajectories to test our
algorithm. Crossroads are well-known to be important in real-world video surveillance systems. We therefore simulate 3 different paths around crossroads: going straight, turning left and turning right on intersection - see Fig. 3.4 (a). Fig. 3.4 (b) shows some normal event examples of our crossroad data. We also simulate two anomalies, running traffic light - Fig. 3.4 (c), and failure to yield to oncoming traffic while turning left - see Fig. 3.4 (d). The synthetic trajectories are represented as loci in x-y spatial coordinates, depicted within a short time duration.

We test the proposed joint sparsity model on both synthetic data and real data. For single-object anomalies, we test on CAVIAR [44] data set in Fig. 3.2 (a) and the Xerox Stop Sign video database - represented in Fig. 3.2 (b). For multiple-object anomalies, we test on the aforementioned simulated trajectories. We compare against two well-recognized techniques in trajectory based anomalous event detection: the work of Piciarelli et al [1] using one class SVMs, and the multi-object tracking and rule-based anomaly detection technique of Han et al [2].

### 3.4.3 Detection Rates for Single Object Anomaly Detection

For the CAVIAR data set, our dictionary consists of 10 normal trajectory classes and 3 anomalous trajectory classes, each class contains 10 different trajectories. 21 normal trajectories and 19 anomalous trajectories are used as independent test data. Because training for anomalous events is well-represented in this database, (2.4) is used to classify the test trajectories. The confusion matrices of the proposed method and the approach in [1] are shown in Table 3.1. The improvements are significant and attributed to the merits of the sparsity model in enabling superior classification.
Table 3.1. Confusion matrices of proposed method and the method in [1] on CAVIAR data for single object anomaly detection

<table>
<thead>
<tr>
<th>Joint Sparsity Model</th>
<th>Piciarelli et al. [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>95.2%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>4.8%</td>
</tr>
</tbody>
</table>

Table 3.2. Confusion matrices of proposed method and the method in [1] on Xerox Stop Sign Data for single object anomaly detection

<table>
<thead>
<tr>
<th>Joint Sparsity Model</th>
<th>Piciarelli et al.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>94.1%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>5.9%</td>
</tr>
</tbody>
</table>

For Stop Sign data set, the training class is classified into 9 normal trajectory classes (containing 8 trajectories each) and 1 anomalous trajectory class (containing 4 trajectories). We use an independent set of 34 normal trajectories and 8 anomalous trajectories to test our approach. Table 3.2 shows the confusion matrices of both approaches. Again, the superior performance of the sparsity model is evident.

**ROC Curves:** By varying the threshold $\tau_1$ in (3.16), we generate receiver operating characteristic (ROC) curves on CAVIAR data as well as the Xerox Stop Sign video database. Figs. 3.5 and 3.6 reveal that the proposed method outperforms Piciarelli et al. by a considerable margin.

### 3.4.4 Detection Rates for Multiple Object Anomaly Detection

In our synthetic data, there are 10 different 2-object normal event classes (containing 100 trajectory pairs each) and 2 different anomalous classes corresponding to the examples in Figs. 3.4 (c) and (d). 50 normal trajectory pairs and 50 anomalous
trajectory pairs were used for testing. Note in this case, success is simply determined by rate of correct classification into normal vs. anomalous categories. The confusion matrices of our method against the two competing trajectory based techniques are shown in Table 3.3. In [2], anomalies are detected by the use of context
Table 3.3. Confusion matrices of proposed method, method in [1] and [2] on synthetic data for multiple object anomaly detection

<table>
<thead>
<tr>
<th></th>
<th>Joint Sparsity Model</th>
<th>Piciarelli et al.</th>
<th>Han et al. [2]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>98.0%</td>
<td>8.0%</td>
<td>86.0%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>2.0%</td>
<td>92.0%</td>
<td>14.0%</td>
</tr>
</tbody>
</table>

based rules on the result of multi-object tracking. This puts an unreasonable burden on defining these rules and is often restrictive in practice, i.e. not all anomalies can be anticipated. In the proposed joint sparsity model, interactions between distinct object trajectories are better captured and departures from expected “joint behavior” (particularly in the case training for anomalies is absent/limited) is employed which enables the improvement in detection rates. The improvements over the technique of Piciarelli et al [1] is expected since this technique is really for single object anomaly detection, and captures only those multi-object anomalies where all object trajectories are simultaneously anomalous.

**ROC Curves:** Similarly, by changing threshold $\tau_2$ in (3.17), the ROC curves of three different methods on synthetic multi-object data are shown in Figs. 3.7. The merits of the joint sparsity model in terms of improved detection are readily apparent.

### 3.5 Conclusion

In this chapter, we present a new joint sparsity model for video anomaly detection. An over-complete training dictionary is derived comprising joint observations of multi-object events; and a test event is reconstructed as a sparse linear combination of events in the training dictionary. For the joint sparsity model to be meaningful, constraints are posed on the structure of sparse coefficients which leads to a
new simultaneous sparsity optimization problem. In the supervised case, where anomalies are pre-characterized into classes, the anomaly detection reduces to a sparsity based classification problem. In the more realistic unsupervised scenario where anomalies cannot be sufficiently pre-characterized, the anomaly detection is accomplished via a multi-object outlier rejection measure. Experimental results on real and synthetic data demonstrate that our approach exhibits superior performance over state-of-the-art techniques, and paves the way for a powerful framework for sparsity-based anomaly detection.

Figure 3.7. ROC curves of 3 different methods on Synthetic data
Chapter 4

Practical Methods for Sparsity Based Video Anomaly Detection

4.1 Introduction

In Chapter 3, we have proposed a joint sparsity model that enables the detection of joint anomalies involving multiple objects. This extension is highly nontrivial since it leads to a new simultaneous sparsity problem which we solve using a greedy pursuit technique. Remaining challenges include: (1) the ability of the linear sparsity model to effectively allow for class separation, and (2) increased computational burden and the need for generous manually labeled training. Our work focuses on overcoming these limitations. In this chapter, we will propose two practical methods to improve our sparsity models. First, we introduce non-linearity into the linear sparsity model to enable superior class separability and hence anomaly detection. Second, we develop a dictionary design and optimization technique that can effectively reduce the size of training dictionaries that enable sparsity based classification/anomaly detection without adversely influencing detection perfor-
4.2 Kernel-based Sparsity Model for Video Anomaly Detection

4.2.1 Kernel Functions

The effectiveness of the proposed joint sparsity model largely depends on the structure of the trajectory data. If the data is not linearly separable enough, the trajectory-based sparsity model may not enable sufficiently accurate reconstruction to be reliable from a classification standpoint. Kernel-based algorithms [28] implicitly exploit the higher-order nonlinear structure of the data which is not captured by the linear models. Kernel methods can be applied to transform the data into a feature space via a transformation $\phi(\cdot)$ such that the resulting transformed trajectory vectors becomes more separable [25–27] and comply with the linear sparsity model.

The kernel function $\kappa : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, is usually defined as the inner product:

$$\kappa(x, z) = \langle \phi(x), \phi(z) \rangle$$

Some popular kernel functions include:

1. Gaussian radial basis function (RBF): $\kappa(x, z) = e^{-\gamma \|x-z\|^2}$, for $\gamma > 0$.

2. homogeneous polynomial kernel: $\kappa(x, z) = (x \cdot z)^d$.

3. Inhomogeneous polynomial kernel: $\kappa(x, z) = (x \cdot z + 1)^d$. 
4.2.2 Kernel Sparsity Model for Object Trajectory

The transformed training trajectory vectors are written as: $a_i \mapsto \phi(a_i)$, where $a_i$ is the $i$-th column of $A$. Let $\phi(y)$ denote the representation of the test trajectory in this space. The test trajectory of original sparsity model in Eq. (2.2) can now be represented as follows:

$$
\phi(y) = \begin{bmatrix}
\phi(a_1) & \cdots & \phi(a_{KT})
\end{bmatrix}
\begin{bmatrix}
\alpha'_1 & \cdots & \alpha'_{KT}
\end{bmatrix}^T = A_\phi \alpha',
$$

(4.1)

where $\alpha'$ is also assumed to be sparse.

Similar to Eq. (2.3), the new sparse coefficient vector $\alpha' \in \mathbb{R}^{KT}$ can be recovered by solving:

$$
\hat{\alpha}' = \arg \min_{\alpha'} \|\alpha'\|_1 \text{ subject to } \|\phi(y) - A_\phi \alpha'\|_2 < \epsilon,
$$

(4.2)

The problem in Eq. (4.2) can be approximately solved by kernel orthogonal/basis matching pursuit algorithms [29–31,45]. Note that in the above problem formulation, we are solving for the sparse vector $\alpha'$ directly in the feature space using the implicit feature vectors, but not evaluating the kernel functions at the training points.

The well-known row sparsity problem in Eq. (3.10) can be extended to the (kernelized) feature space as follows:

$$
\text{minimize} \quad \|S_\phi\|_{\text{row,0}}
\text{subject to} \quad \|Y_\phi - A_\phi S_\phi\|_F \leq \epsilon,
$$

(4.3)

Let $\Theta_A \in \mathbb{R}^{PKT \times PKT}$ be the kernel matrix whose $(i,j)$-th entry is $\kappa(a_i, a_j)$.
\( \theta_{A,Y} \in \mathbb{R}^{PKT} \) be the vector whose \( i \)-th entry is \( \kappa(a_i, y) \). The correlation (dot product) between the test trajectory \( \phi(y) \) and a training trajectory from dictionary \( \phi(a_i) \) is then computed by:

\[
c_i = \langle \phi(y), \phi(a_i) \rangle = \kappa(y, a_i) = (\theta_{A,Y})_i
\] (4.4)

The orthogonal projection coefficient of \( \phi(y) \) onto a set of selected training trajectories \( \{\phi(a_i)\}_{i \in \Lambda} \) is given as:

\[
p_{\Lambda} = ((\Theta_A)_{\Lambda,\Lambda})^{-1}(\theta_{A,Y})_{\Lambda}
\] (4.5)

And the residual vector between \( \phi(y) \) and its approximation using the selected training trajectories \( \{\phi(a_n)\}_{n \in \Lambda} = (A_{\phi})_{:, \Lambda} \) is then expressed as:

\[
\phi(r) = \phi(y) - (A_{\phi})_{:, \Lambda}((\Theta_A)_{\Lambda,\Lambda})^{-1}(\theta_{A,Y})_{\Lambda}
\] (4.6)

The correlation between \( \phi(r) \) and an atom \( \phi(a_i) \) can be computed by:

\[
c_i = \langle \phi(r), \phi(a_i) \rangle = (\theta_{A,Y})_i - (\Theta_A)_{i,\Lambda}((\Theta_A)_{\Lambda,\Lambda})^{-1}(\theta_{A,Y})_{\Lambda}
\] (4.7)

Let \( C \in \mathbb{R}^{PKT \times P} \) be the correlation matrix whose \((i, j)\)-th entry is the correlation between \( \phi(a_i) \) and \( \phi(r_j) \), where \( \phi(r_j) \) is the residual vector of \( \phi(y_j) \). From Eq. (4.7), the correlation matrix can be expressed as:

\[
C = \Theta_{A,Y} - (\Theta_A)_{:,\Lambda}((\Theta_A)_{\Lambda,\Lambda})^{-1}(\Theta_{A,Y})_{:,\Lambda}
\] (4.8)

The new training atom is then can be chosen as the one associated with the
Algorithm 2 KSOMP

Input: Dictionary $A = [a_1 \ a_2 \ \ldots \ a_{PKT}]$, data matrix $Y = [y_1 \ y_2 \ \ldots \ y_P]$, kernel function $\kappa$, a stopping criterion

Initialization: compute the kernel matrices $\Theta_A$ and $\Theta_{A,Y}$. Set index set $\Lambda_0 = \arg\max_i \| (\Theta_A)_{i,:}\|_2$ and iteration counter $t = 1$.

1: while stopping criterion has not been met do
   1. Compute the correlation matrix
      \[ C = \Theta_{A,Y} - (\Theta_A)_{\Lambda_{t-1}}((\Theta_A)_{\Lambda_{t-1},\Lambda_{t-1}} + \lambda I)^{-1}(\Theta_{A,Y})_{\Lambda_{t-1},:} \]
   2. Select the new index as $\lambda_t = \arg\max_i \| C_{i,:}\|_2$
   3. Update the index set $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$
   4. $t \leftarrow t + 1$
2: end while

Output: Index set $\Lambda = \Lambda_{t-1}$, the sparse representation $S_\phi$ whose nonzero rows indexed by $S_\phi = (\Theta_{A,A} + \lambda I)^{-1}(\Theta_{A,Y})_{\Lambda,:}$

row of $C$, which has the maximal $l_2$ norm. Referring to the SOMP algorithm in Algorithm 1, the KSOMP algorithm, which is used to solve Eq. (4.9), is proposed as Algorithm 2. Note that a regularization term $\lambda I$ is added in order to have a stable inversion. $\lambda$ is usually a small number (we choose $\lambda = 0.0001$ in our experiments).

The kernelized version of our proposed joint sparsity model in Eq. (3.9) is given as:

\[
\begin{align*}
\text{minimize} & \quad \| J (H \circ S_\phi) \|_{\text{row,0}} \\
\text{subject to} & \quad \| Y_\phi - A_\phi S_\phi \|_F \leq \epsilon,
\end{align*}
\]

(4.9)

which is very similar to Eq. (4.3) but for the presence of the matrices $J, H$ and the Hadamard operator. Recall the modification to SOMP employed in Chapter 3, a similar trick will allow us to adapt kernelized SOMP to yield a solution for
our problem in Eq (4.9). In particular, instead of using the selection rule:

\[
\lambda_t = \arg \max_i \|C_{i,:}\|_2, \tag{4.10}
\]

in Step 2 of KSOMP, we can jointly select atoms of trajectories from the same training:

\[
\lambda_t = \arg \max_i \sum_j \|C_{i,:}^j\|_2, \tag{4.11}
\]

where \(C_{i,:}^j\) refers to the correlation matrix of the \(i\)-th trajectory.

Note that in the transformed space, the residual becomes:

\[
\|\phi(y) - A \phi \alpha'\|_2
\]

\[
= \left[\sum_{i=1}^n (\phi(y)_i - (A \phi \alpha')_i)^2\right]^{\frac{1}{2}}
\]

\[
= \left[\sum_{i=1}^n (\phi(y)_i - \sum_{j=1}^{KT} \alpha_j' (A \phi)_{i,j})^2\right]^{\frac{1}{2}}
\]

\[
= \left[\sum_{i=1}^n (\phi(y)_i)^2 - 2 \phi(y)_i \sum_{j=1}^{KT} \alpha_j' (A \phi)_{i,j} + \sum_{j=1}^{KT} (A \phi)_{i,j})^2\right]^{\frac{1}{2}}
\]

\[
= \left[\sum_{i=1}^n (\phi(y)_i)^2 - 2 \sum_{j=1}^{KT} \alpha_j' \sum_{i=1}^n \phi(y)_i (A \phi)_{i,j} + \sum_{i=1}^n (A \phi)_{i,j})^2\right]^{\frac{1}{2}}
\]

\[
= (\kappa(y, y) - 2 \alpha'^T \theta_A y + \alpha'^T \Theta_A \alpha')^{\frac{1}{2}}
\]

Let \(\hat{S}_\phi\) be the optimum solution of Eq. (4.9). The residual for the kernelized joint sparsity model corresponding to the \(i\)-th class is then given by:

\[
\|Y_\phi - A \phi \delta_i(\hat{S}_\phi)\|_F = (\sum_{j=1}^n (\|\phi(y_j) - A \phi(\delta_i(\hat{S}_\phi))_{:,j}\|_2)^2)^{\frac{1}{2}}, \tag{4.12}
\]

where \(\phi(y_j)\) is the \(j\)-th column of \(Y_\phi\) and \((\delta_i(\hat{S}_\phi))_{:,j}\) is the \(j\)-th column of \(\delta_i(\hat{S}_\phi)\).
More precisely, in terms of the kernel function, this is given by:

$$r_i(Y_\phi) = \left( \sum_{j=1}^{p} \left( \kappa(y_j, y_j) - 2(\delta_i(\hat{S}_\phi))_{i,j}^T (\Theta_{A,y})_{\Omega,\Omega_i} + (\delta_i(\hat{S}_\phi))_{i,j}^T (\Theta_A)_{\Omega_i,\Omega_i} (\delta_i(\hat{S}_\phi))_{i,j} \right) \right)^{\frac{1}{2}},$$

(4.13)

where $\Omega_i$ is the index set associated with the $i$-th training class.

The class of $Y_\phi$ is determined by:

$$\text{identity}(Y_\phi) = \arg \min_i \|Y_\phi - A_\phi \delta_i(\hat{S}_\phi)\|_F,$$

(4.14)

### 4.2.3 Kernel Parameters Optimization

We focus on picking parameters for the RBF kernel $\kappa(x, z) = e^{-\gamma \|x-z\|^2}$ which will be used in all our experiments.

For different choices of the RBF parameter $\gamma$, multiple training dictionaries are generated $A_\phi(\gamma)$, i.e., $A_\phi(\gamma)$ is a function of $\gamma$. Inspired by cross-validation, we split the training data $A_\phi(\gamma)$ into two subsets $B_\phi(\gamma)$ and $C_\phi(\gamma)$, such that both $B_\phi(\gamma)$ and $C_\phi(\gamma)$ have representation from the $K$ classes. Now if the dictionary in the sparsity model is chosen to be equal to $B_\phi(\gamma)$ and a (transformed) test trajectory is picked from $C_\phi(\gamma)$, then, ideally we expect perfect classification into one of the $K$ classes. Therefore, a good kernel is one that will enable close to ideal classification of test samples from $C_\phi(\gamma)$ - which means that only a small number of $\hat{S}_\phi(\gamma)$ are activated (non-zero) and for one particular class. Here the outlier rejection measure:

$$JSCI(\hat{S}_\phi(\gamma)) = \frac{K \cdot \max_i \|\delta_i(\hat{S}_\phi(\gamma))\|_1/\|\hat{S}_\phi(\gamma)\|_1 - 1}{K - 1},$$

(4.15)

where $\delta_i(\hat{S}_\phi(\gamma))$ is the vector whose only nonzero entries are the same as those in
$\hat{S}_p(\gamma)$ associated with class $i$. $JSCI(\hat{S}_p(\gamma))$ will be very close to 1 if the classification is accurate. Therefore the best parameter $\gamma$ can be chosen by solving the following kernel parameter optimization problem:

$$\arg\max_{\gamma} JSCI(\hat{S}_p(\gamma)),$$

(4.16)

### 4.2.4 Experimental Results

Our proposed algorithm for multi-object trajectory based anomaly detection is called Joint Kernel-based Sparsity Model (abbreviated to JKSM) or equivalently Kernel Sparsity Model (KSM) in the case of single object anomaly detection. We test the KSM and JKSM algorithms on several challenging video datasets. For single-object anomalies, we test on the CAVIAR [44] data set in Fig. 3.2 (a) and the Xerox Stop Sign video database - represented in Fig. 3.2 (b). For multiple-object anomalies, we test on AVSS [46] data set -see Fig. 3.3 (a), the Xerox Stop Sign data set - multi-object example in Fig. 3.3 (b), and Xerox Intersection data set - see Fig. 3.3 (c). We also compare our experimental results against three well-recognized techniques in trajectory based anomalous event detection: 1.) the recent sparsity based technique of Li et al. [21], 2.) the approach of Piciarelli et al. [1] using one class SVMs, and 3.) the multiple-object tracking and rule-based anomaly detection technique of Han et al. [2].

#### 4.2.4.1 Benefits of sparsity under object occlusion or missing trajectory information:

Because of the limitation of camera’s visual angle, occlusions often occur in video data. Figs. 4.1 (a) and (b) show two examples of occlusions. In Fig. 4.1 (a), a car
Figure 4.1. Occlusion examples: (a) moving occlusion: a car is occluded by another car (b) static occlusion: a car is occluded by the stop sign.

Figure 4.2. An example of occluded trajectories from the Xerox Stop Sign video data. is occluded by another car (we call it a moving occlusion)- see Fig. 4.1(a3). In Fig. 4.1 (b), the car is occluded by the stop sign (static occlusion) - see Fig. 4.1 (b2).

Tracking algorithms continue to strive to improve and perform well even under occlusion [47–49]. Since this is a fundamentally hard problem, occlusion occurs in practice and leads to missing or corrupted trajectory data. The merits of sparse representations for occluded trajectories have also been argued by Li et al. [21] but not explicitly demonstrated. Next, we perform a simple experiment to illustrate this.
Table 4.1. Confusion matrices of proposed and state-of-the art trajectory based methods on Xerox Stop Sign occluded data – single-object anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>Piciarelli et al. [1]</th>
<th>Li et al. [21]</th>
<th>KSM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>61.5%</td>
<td>50.0%</td>
<td>84.6%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>38.5%</td>
<td>50.0%</td>
<td>15.4%</td>
</tr>
</tbody>
</table>

We work with the Xerox stop sign data set and 95 trajectories (including 76 training trajectories without occlusion and 19 occluded trajectories) obtained from 39 video clips. Our training dictionary consists of 9 normal trajectory classes (containing 8 trajectories each) and 1 anomalous trajectory classes (containing 4 trajectories). An independent set of 13 normal but occluded trajectories and 6 anomalous but occluded trajectories are used to test our approach. Fig. 4.6 shows an example of occluded trajectories from Xerox Stop Sign data. Note that occluded trajectory locations are replaced by a constant value (e.g. zero) for the duration that object tracking is lost. This is a common characteristic of frame-based tracking approaches [42,50,51].

Because training for anomalous events is well-represented in this database, Eq. (2.4) is used to classify the test trajectories. The RBF function is chosen as the kernel for the KSM algorithm. The confusion matrices of three methods - KSM and the techniques by Picarelli and Li et al. are reported in Table 4.7. Note that both Li et al. which is also a sparsity based anomaly detection technique, as well the proposed KSM methods do better than Picarelli et al owing to the robustness of sparse coefficients under occlusion. Further, KSM is mildly better than Li et al. for this example owing to the non-linearity in the sparsity model as introduced by the use of the kernel. A more thorough evaluation of the three methods on real-world databases is reported next.
Table 4.2. Confusion matrices of proposed and state-of-the-art trajectory based methods on CAVIAR data – single-object anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>Piciarelli et al. [1]</th>
<th>Li et al. [21]</th>
<th>KSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>85.7%</td>
<td>90.5%</td>
<td>95.2%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>14.3%</td>
<td>9.5%</td>
<td>4.8%</td>
</tr>
</tbody>
</table>

4.2.4.2 Detection Rates for Single-Object Anomaly Detection

For the CAVIAR data set, we test on 27 video clips from which 170 trajectories are extracted. Our training dictionary consists of 10 normal trajectory classes and 3 anomalous trajectory classes, each class contains 10 different training trajectories. 21 normal trajectories and 19 anomalous trajectories are used as independent test data. Because training for anomalous events is well-represented in this database, Eq. (2.4) is used to classify the test trajectories as normal or anomalous. Our proposed KSM method employs the RBF function as kernel and uses Eq. (4.16) to optimize the kernel parameter. The confusion matrices of KSM are compared with the approach in [1] and the sparsity model by Li et al. [21] in Table 4.2. First, we note the benefits of sparsity based anomaly detection in the form of improved detection rates of Li et al. over Piciarelli’s approach. Second KSM serves to further improve detection rates by virtue of the use of a non-linear kernel that enhances the accuracy of the underlying sparsity reconstruction - which is assumed linear by Li et al.

For the Xerox Stop Sign data set, 118 trajectories from 39 video clips are extracted. The training dictionary comprises 9 normal trajectory classes (containing 8 trajectories each) and 1 anomalous trajectory class (containing 4 trajectories). Again, we have training for anomalies. So we use Eq. (2.4) to classify a given test trajectory. An independent set of 34 normal trajectories and 8 anomalous trajectories are used to test our approach. Table 4.3 shows the confusion matrices
Table 4.3. Confusion matrices of proposed and state-of-the art trajectory based methods on the Xerox Stop Sign data – single-object anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>Piciarelli et al. [1]</th>
<th>Li et al. [21]</th>
<th>KSM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>85.3%</td>
<td>37.5%</td>
<td>91.2%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>14.7%</td>
<td>62.5%</td>
<td>8.8%</td>
</tr>
</tbody>
</table>

of three approaches. Again, the benefits of classification using sparsity model vs. SVM-based classifier are readily apparent. As with the CAVIAR data, a comparison of Li et al. and KSM demonstrates a significant advantage brought about by employing the kernel function.

**ROC Curves:** In order to test the robustness of the proposed method, we remove all the training for anomalous events from the dictionary and use all 40 test trajectories on the CAVIAR data set and all 42 test trajectories on Xerox Stop Sign data set. In this case, (3.16) is used to determine whether the test trajectory is normal or anomalous. We can observe from Eq. (3.16) that if we increase the threshold $\tau_1$, more events will be identified as anomalies, thus may lead to an increase of the false positive rate. On the other hand, if $\tau_1$ decreases, more events will be classified into normal events, therefore, the false negative rate will increase correspondingly. By varying the threshold $\tau_1$ in (3.16), we can get the relationship between true positive rate and false positive rate for both Li et al. and KSM approaches. As to method in Piciarelli et al., we can obtain similar relationship by changing the threshold angle $\theta_{th}$ in Eq. (1.3). By varying the individual thresholds for each algorithm, we obtain receiver operating characteristic (ROC) curves for these three different methods on CAVIAR data (Figs. 4.3) as well as the Xerox Stop Sign video database (Fig. 4.4). Figs. 4.3 and 4.4 reveal that the proposed method outperforms the approaches by Piciarelli et al. and Li et al. by a considerable margin.
Figure 4.3. ROC curves for single-object anomaly detection – CAVIAR video data set.

4.2.4.3 Detection Rates for Multiple-Object Anomaly Detection

For multiple (here 2) object anomaly detection, we will compare against Han et al. [2]\textsuperscript{1} and Picarelli et al. [1]. Since the one class SVM method in Picarelli et al.\textsuperscript{1} is really proposed for single object anomaly detection, we build two intuitively motivated extensions to evaluate its performance in the multi (here 2) object setting:

1. If either of the trajectories corresponding to the two objects is an anomaly, the joint event is called anomalous. We denote this method as Picarelli et al.\textsuperscript{1}.

2. If both of the two trajectories corresponding to the two objects is individually found to be anomalous, only then is the joint event anomalous. We denote this method as Picarelli et al.\textsuperscript{2}.

\textsuperscript{1}The predefined anomaly detection rules in Han et al are based on underlying scenario.
To ensure the application of Picarelli et al., we separate every 2-object event into 2 individual events with known individual class labels (normal or anomalous) so that we can obtain results for the aforementioned extensions, i.e. Picarelli et al.\_1 and Picarelli et al.\_2.

For the Xerox Stop Sign data set, we identify 4 different 2-object normal event classes. Each class contains 15 training trajectory pairs. This database inherently contains 3 multiple-object anomalies, one of which is illustrated in Fig. 3.3 (b) with the actual video at: http://youtu.be/tPR-LI3NmiM. Using the outlier rejection in Eq. (3.17), all three multiple-object anomalies are successfully detected by our proposed JKSM. The detection rates of all methods are shown in Table 4.4.

### Table 4.4. Confusion matrices of proposed and state-of-the art trajectory based methods on the Xerox Stop Sign data – multiple-object anomaly detection.

<table>
<thead>
<tr>
<th>Detection Rates</th>
<th>Picarelli et al._1</th>
<th>Picarelli et al._2</th>
<th>Han et al. [2]</th>
<th>JKSM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3/3</td>
<td>1/3</td>
<td>2/3</td>
<td>3/3</td>
</tr>
</tbody>
</table>

Figure 4.4. ROC curves for single-object anomaly detection – Xerox Stop Sign video dataset.
Table 4.5. Detection rates of proposed and state-of-the art trajectory based methods on AVSS data – multiple-object anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>Piciarelli et al.,1</th>
<th>Piciarelli et al.,2</th>
<th>Han et al.,2</th>
<th>JKSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detection Rates:</td>
<td>2/2</td>
<td>0/2</td>
<td>2/2</td>
<td>2/2</td>
</tr>
</tbody>
</table>

For the AVSS data set, 3 different 2-object normal event classes (containing 24 training trajectory pairs each) are chosen. The database was experimentally found to contain 2 different anomalies - corresponding videos can be seen at: http://youtu.be/mU5R056zInc and http://youtu.be/jEzLkWF65Io. Our outlier rejection measure in Eq. (3.17) was again successfully able to detect both these anomalies. Table 4.4 shows the detection rates of 4 methods.

In Xerox Intersection data, there are 91 trajectory pairs extracted from 13 video clips. We manually build our training dictionary into 6 different 2-object normal event classes (containing 6 trajectory pairs each) and 6 different anomalous classes (containing 4 trajectory pairs each). 17 normal trajectory pairs and 14 anomalous trajectory pairs are used for testing. Again, we use the RBF kernel function. The confusion matrices of our method against the two competing trajectory based techniques are shown in Table 4.6.

It is easy to see from Table 4.6 that the proposed JKSM method leads to the best detection rates. The improvement over the techniques of Piciarelli et al. [1] is expected since this technique is really for single-object anomaly detection and the extensions Piciarelli et al.,1 and Piciarelli et al.,2 - will either strongly compromise detection or lead to high false alarm. In [2], anomalies are detected by the use of context based rules on the result of multiple-object tracking. This puts an unreasonable burden on defining these rules and is often restrictive in practice, i.e. not all anomalies can be anticipated. In the proposed joint sparsity model, interactions between distinct object trajectories are better captured and departures
Table 4.6. Confusion matrices of proposed and state-of-the art trajectory based methods on the Xerox Intersection dataset – multiple-object anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>Piciarelli et al. 1</th>
<th>Piciarelli et al. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>58.8%</td>
<td>7.1%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>41.2%</td>
<td>92.9%</td>
</tr>
<tr>
<td>Han et al. [2]</td>
<td>JKSM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>82.4%</td>
<td>35.7%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>17.6%</td>
<td>64.3%</td>
</tr>
</tbody>
</table>

Figure 4.5. ROC curves for multiple-object anomaly detection – Xerox Intersection dataset.

from expected “joint behavior” (particularly in the case training for anomalies is absent/limited) is employed which enables the improvement in detection rates.

**ROC Curves:** By changing the threshold $\tau_2$ correspond to Eq. (3.17) and $\theta_{th}$, we can generate the ROC curves for JKSM, JSM (joint sparsity model without the kernel) and Piciarelli et al. (Here, we average the ROC curves of Piciarelli et al. 1 and Piciarelli et al. 2 as the ROC curve of Piciarelli et al.). For Han et al, we loosen and tighten the anomaly detection rules to obtain the relationship between
true positive rate and false positive rate. The ROC curves of these four different methods on all 31 test trajectory pairs from Xerox Intersection data set are shown in Figs. 4.5. The statistical merits of the joint sparsity model are clearly revealed by the ROC curves. Again, the use of the RBF kernel enables JKSM to further improve performance.

4.3 Discriminative Dictionary Learning Technique

4.3.1 Motivation: Dictionary Learning

Sparsity-based approaches exhibit two important limitations. First, the effectiveness of this sparsity model largely depends on the structure of training data. It is true that an over complete training dictionary $\mathbf{A}$ which explicitly concatenates trajectories from event classes naturally induces sparsity. However, the presence of redundant example trajectories in the training dictionary also increases the computational complexity. This is a common challenge in the field of dictionary learning [52,53]. The second limitation is that in most sparsity models, dictionaries are manually labeled. Manual dictionary labeling is labor intensive and often delays practical deployment. In real world applications, it is hence desirable that labels may be determined in an automatic, unsupervised manner. We propose solutions to both of these problems. First we propose a dictionary pruning and optimization technique which can effectively prune the size of the training dictionaries while maintaining acceptable detection performance. Secondly, we suggest the use of state of the art automatic trajectory clustering techniques for initializing dictionaries which can alleviate the burden of manual labeling.
4.3.2 Dictionary Pruning and Optimization

As we discussed before, redundant information on the training dictionary will increase the computational complexity of the sparsity model. In real world scenarios, some normal trajectories are indeed very similar to each other. In Fig. 2.1, trajectories in each training class are quite similar. Therefore, we can try to reduce or eliminate redundancy in the training trajectories with minimal sacrifice in anomaly detection performance.

4.3.2.1 Notation

For ease of exposition, consider a classification problem with data from two different classes $C_1$ and $C_2$. Let $\{y^i_j\}, i=1, \ldots, N_j$ be the $N_j$ training samples corresponding to class $C_j, j = 1, 2$. (In this report, subscripts refer to class indices.) The training samples from class $C_j$ are collected into a matrix $Y_j, j = 1, 2$. Suppose that we also have access to a “good” choice of initial dictionary $A = [A_1 A_2]$, where $A_j$ corresponds to class $C_j, j = 1, 2$. Accordingly, any training sample $y$ has a sparse representation $x$ in terms of the dictionary $A$:

$$y = Ax = [A_1 A_2]x = [A_1 A_2] \begin{bmatrix} I_1(x) \\ I_2(x) \end{bmatrix},$$

where $I_j(x)$ is the indicator function operating on $x$ to extract out those components which correspond to $A_j$.

Our goal is to jointly learn dictionaries $A_1$ and $A_2$ which encourage samples from class $C_i$ to be represented (with a low reconstruction error tolerance) as linear combinations of atoms (or columns) from $A_i$ while not having any (ideally) contribution from the complementary dictionary $A_j, j \neq i$. 
4.3.2.2 Optimization problems

1. Dictionary Initialization

The classes in the training dictionary \( A \) can be initialized in one of two ways: (i) Each of the (training) trajectories in class \( C_j, j = 1, 2 \) is manually labeled by a human operator (ii) Labels and hence dictionaries \([A_1, A_2]\) are automatically generated on a collection of trajectories using unsupervised techniques such as trajectory clustering techniques [54, 55].

2. Sparse Coding within Each Class

with fixed dictionary \( A_1, A_2 \) minimize objective function with respect to \( x^j_i \).

it can be formulated as:

\[
\hat{x}^j_i = \arg \min_{x^j_i} \|x^j_i\|_{0/1} \text{ subject to } \|y^j_i - A_j x^j_i\|_2 < \epsilon_1, i = 1, ..., N_j; j = 1, 2
\]

(4.18)

Instead of using the entire training dictionary \( A \), the sparse coefficient vector \( x^j_i \) is only recovered by the corresponding training dictionary \( A_j \). Given sufficient training, we also seek sparsity within classes. Since the sparsity within class is weaker than the sparsity among the entire dictionary, the threshold \( \epsilon_1 \) in Eq. (4.18) should be chosen smaller than usual cases.

3. Redundant Dictionary Column Removal

In order to reduce the size of training dictionary, we remove one training trajectory from each class every time. We want to remove the training which contribute least to the test trajectory reconstruction. These training can be
selected by:
\[
\hat{\lambda}_j = \arg \min_{\lambda_j} \| \sum_{i=1}^{N_j} |\hat{x}_j^i(\lambda_j)| \|_1,
\]
(4.19)
where \(\hat{x}_j^i(\lambda_j)\) is the \(\lambda_j\)-th element of vector \(\hat{x}_j^i\), \(\hat{\lambda}_j\) is the index of the training which will be removed from dictionary \(A_j\).

4. Discriminative Dictionary Learning

For training samples from class \(C_1\), we expect strong contributions from atoms in \(A_1\) - leading to low reconstruction error bounded by some \(\epsilon_2\) - while simultaneously requiring little or no contribution from \(A_2\). Consequently, we would like to minimize the difference of the two reconstruction errors, weighted by a suitable parameter \(\lambda_1\) learnt from training. In addition, we also do not want \(A_1\) and \(A_2\) converge to random matrices. Therefore we add constraints to make sure \(A_1\) and \(A_2\) have very little changes.

\[
(A_1^*, A_2^*) = \arg \min_{A_1, A_2} \| Y_1 - \tilde{A}_1 \tilde{X} \|_F - \lambda_1 \| Y_1 - \tilde{A}_2 \tilde{X} \|_F + \| Y_2 - \tilde{A}_2 \tilde{X} \|_F - \lambda_2 \| Y_2 - \tilde{A}_1 \tilde{X} \|_F
\]
subject to \(\| Y_1 - \tilde{A}_1 \tilde{X} \|_F \leq \epsilon_2, \| Y_2 - \tilde{A}_2 \tilde{X} \|_F \leq \epsilon_3, \| \tilde{A}_1 - A_1^0 \|_F \leq \delta_1, \| \tilde{A}_2 - A_2^0 \|_F \leq \delta_2\)
(4.20)

where \(\tilde{A}_1 = [A_1 0]\), i.e., the matrix \(A\) with \(A_2\) replaced by zeros; likewise \(\tilde{A}_2 = [0 A_2]\), \(A_1^0\) and \(A_2^0\) are the initial input of \(A_1\) and \(A_2\).

Steps 2-4 are repeated until no further improvement is possible or a sufficient number of training trajectories have been removed in Step 3.
4.3.2.3 Multi-class tasks

The algorithm is described for binary classification and it easily generalizes to the $K$-class scenario by solving $K$ such problems. The positive term becomes:
\[ \sum_{k=1}^{K} \| Y_k - \bar{A}_k \tilde{X} \|_F, \]
the subtracted term in the cost function is replaced by a sum of such terms from all other classes $C_i, i \neq k$.

4.3.3 Trajectory Clustering for Dictionary Initialization

In Step 1 of our dictionary learning algorithm, two different ways can be used to initialize the training dictionary: manual labeling or unsupervised techniques. In order to alleviate the burden of manual labeling, we apply trajectory clustering techniques which can automatically group similar trajectories.

A variety of trajectory clustering algorithms have been designed for automatic activity analysis in video surveillance. A key component of any trajectory clustering algorithm is an effective distance measure of trajectories. A desirable property is that the distance measure between two trajectories be small when both trajectories fall in the same class, and conversely large when the trajectories are in different classes. Commonly used measures are Euclidean distance, Hausdorff distance and dynamic time warping (DTW) [55]. Given a distance measure, a compatible clustering method (for example k-means clustering in the case of Euclidean distance) can be applied. In this paper, we employ a state of the art trajectory clustering technique described in Rodriguez-Serrano and Singh [54] for initializing the training dictionary.

In [54], each trajectory (position and velocity) is modeled using a hidden Markov model (HMM), and the distance between two trajectories is computed as the probability product kernel (PPK). The authors use a spectral clustering
algorithm which consists of the following steps:

1. Given an affinity function \( h(x_i, x_j) \), we compute the affinity matrix \( B \) with entries \( b_{ij} = h(x_i, x_j) \) indicating similarity between trajectories \( T_i \) and \( T_j \).

2. Build diagonal matrix \( D \) as \( d_{ii} = \sum_{j=1}^{N} b_{ij} \).

3. Compute the Laplacian \( L \) as \( L = D^{-1/2}BD^{-1/2} \).

4. Let \( L = Z\Lambda Z^T \) be the eigenvalue decomposition of \( L \).

5. Form the matrix \( Z \) by stacking the eigenvectors in columns.

6. Compute the matrix \( M \) by normalizing all the rows in \( Z \) to have norm 1 (mapping to the unit hypersphere):

   \[
   m_{i,j} = \frac{z_{ij}}{\sqrt{\sum_{r=1}^{k} z_{ir}}}
   \]  

   (4.21)

7. In this new representation, apply a standard clustering algorithm. The authors chose the fast global k-means algorithm [56]. Global k means is a variation of k-means which starts with a single centroid and progressively adds centroids in a controlled manner. It has been shown that it has several advantages with respect to the ordinary k means, such as no need for initialization and that it allows significant speed-ups.
4.3.4 Experimental Results

4.3.4.1 Benefits of sparsity under object occlusion or missing trajectory information:

Because of the limitation of camera’s visual angle, occlusions often occur in video data. Figs. 4.1 (a) and (b) show two examples of occlusions. In Fig. 4.1 (a), a car is occluded by another car (we call it a moving occlusion) - see Fig. 4.1(a3). In Fig. 4.1 (b), the car is occluded by the stop sign (static occlusion) - see Fig. 4.1 (b2).

Tracking algorithms continue to strive to improve and perform well even under occlusion [47–49]. Since this is a fundamentally hard problem, occlusion occurs in practice and leads to missing or corrupted trajectory data. Because our optimization problems in (3.9) and (4.9) are well-conditioned, perturbation theory arguments apply and lend our method robustness under limited trajectory occlusion. In particular, consider:

\[
\hat{S}_o = \arg \min \|J(H \circ S_o)\|_{row,0} \\
\text{subject to} \quad \|Y_o - AS_o\|_F \leq \epsilon,
\]

where \(Y_o\) is a set of occluded test trajectories and \(S_o\) denotes the corresponding sparse coefficients. If \(\|Y_o - Y\|_F < \eta\) then \(\|\hat{S}_o - \hat{S}\|_F < \zeta\), i.e. a small perturbation to the problem should cause only a small perturbation to the solution [57]. It means that if the occluded trajectory set \(Y_o\) is not very different from the original trajectory set \(Y\), then the optimized sparse coefficients under trajectory occlusion will be close enough to \(\hat{S}\), i.e. the solution in the absence of occlusion. Because anomaly detection rests on the structure of the sparse coefficient matrix, this intuitively provides occlusion robustness.

In addition, we perform a simple experiment to illustrate this. We work with
Table 4.7. Confusion matrices of proposed and state-of-the art trajectory based methods on Xerox Stop Sign occluded data – supervised, single-object anomaly detection. KSM detects anomalies using Eq. (2.4)

<table>
<thead>
<tr>
<th></th>
<th>Picarelli et al. [1]</th>
<th>Li et al. [21]</th>
<th>Zhao et al. [22]</th>
<th>KSM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>61.5%</td>
<td>50.0%</td>
<td>84.6%</td>
<td>33.3%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>38.5%</td>
<td>50.0%</td>
<td>15.4%</td>
<td>66.7%</td>
</tr>
</tbody>
</table>

the Xerox stop sign data set and 95 trajectories (including 76 training trajectories without occlusion and 19 occluded trajectories) obtained from 39 video clips. Our training dictionary consists of 9 normal trajectory classes (containing 8 trajectories each) and 1 anomalous trajectory classes (containing 4 trajectories). An independent set of 13 normal but occluded trajectories and 6 anomalous but occluded trajectories are used to test our approach. Fig. 4.6 shows an example of occluded trajectories from Xerox Stop Sign data. Note that occluded trajectory locations are replaced by a constant value (e.g. zero) for the duration that object tracking is lost. This is a common characteristic of frame-based tracking approaches [42,50,51].

Because training for anomalous events is well-represented in this database, Eq. (2.4) is used to classify the test trajectories. The RBF function is chosen as the kernel for the KSM algorithm. The confusion matrices of four methods - KSM and the techniques by Picarelli et al., Li et al. and Zhao et al. are reported in Table 4.7. Note that all Li et al. and Zhao et al. which are also a sparsity based anomaly detection technique, as well the proposed KSM methods do better than Picarelli et al. owing to the robustness of sparse coefficients under occlusion. Further, KSM is mildly better than Li et al. and Zhao et al. for this example owing to the non-linearity in the sparsity model as introduced by the use of the kernel. A more thorough evaluation of the three methods on real-world databases is reported next.
4.3.4.2 Dictionary optimization for reducing complexity

In this part of the experiments, the training dictionaries have been manually labeled as initialization (supervised case). We first employ a Xerox Stop sign data set to evaluate our dictionary learning approach. This data set comprises trajectories of cars driving through a Stop sign. An example anomaly of Xerox Stop sign data can be seen in Fig. 3.2 (b), where a driver backs his car in front of the Stop sign.

For simplicity, the dictionary only contains 2 normal event classes each comprising 16 trajectories. After the discriminative dictionary learning, each class only contains 6 columns. Since we do not have an anomalous event class, the outlier rejection measure in Eq. (3.16) is used to identify anomalies. An independent set of 19 normal trajectories and 7 anomalous trajectories are used to test our approach. Table 4.8 shows the confusion matrices for the sparsity model using the original dictionary versus the pruned dictionary. We observe that even when the new dictionary is much smaller (37.5% of the original dictionary), there is relatively little loss in anomaly detection performance.

We also generate performance ratio curves (the detection rates using partial dictionary divided by the detection rates using entire dictionary, abbreviated to
Table 4.8. Confusion matrices on Xerox Stop Sign data - Dictionary Learning

<table>
<thead>
<tr>
<th></th>
<th>Sparsity Model (original dictionary)</th>
<th>Learned Dictionary (with 37.5% columns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>94.7%</td>
<td>14.3%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>5.3%</td>
<td>85.7%</td>
</tr>
</tbody>
</table>

Figure 4.7. Performance ratio curves with respect to the proportion of original dictionary size - Xerox Stop Sign video data set.

PRC) with respect to the proportion of the dictionary we used. Fig. 4.7 reveals that our dictionary pruning technique can reduce the size of the dictionary significantly without losing much detection performance. Especially, the two points show that using only 25% of original dictionary size, the system still exhibits around 80% performance. With 38% of original dictionary size, there is no compromise in anomaly detection performance with only about a 10% loss in normal event detection performance.

In order to test the computational complexity of our method, the running time figure (seconds) with respect to the proportion of original dictionary size has been plotted in Fig. 4.8. Because the computational complexity of solving Eq.
(2.3) clearly increases with the number of columns in the dictionary A, significant computational gains can be arrived at by pruning. In particular, when using 25% of the original dictionary size, anomaly detection with the pruned dictionary will only cost 20% of the execution time, while yet retaining 80% detection performance.

We then test our dictionary learning algorithm on the public CAVIAR data set. These videos include people walking alone, meeting with others, window shopping, entering and exiting shops, fighting and passing out, and leaving a package in a public place. Fig. 3.2 (a) shows an example anomaly from CAVIAR data set - a man suddenly falls on the floor when walking across the lobby. For the CAVIAR data set, our dictionary consists of 2 normal trajectory classes each class contains 20 different trajectories. Each class only contains 8 columns after the dictionary learning steps are carried out. 18 normal trajectories and 11 anomalous trajectories are used as independent test data. The confusion matrices of sparsity model using entire dictionary and pruned dictionary are shown in Table 4.9. Again, using
Table 4.9. Confusion matrices on CAVIAR data - Dictionary Learning

<table>
<thead>
<tr>
<th></th>
<th>Sparsity Model (original dictionary)</th>
<th>Learned Dictionary (with 40.0% columns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>88.9%</td>
<td>18.2%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>11.1%</td>
<td>81.8%</td>
</tr>
</tbody>
</table>

Figure 4.9. Performance ratio curves with respect to the proportion of original dictionary size - CAVIAR data set.

By varying the portion of dictionary columns used in the sparsity model, we obtain PRC (Fig. 4.9) and running time figure (Fig. 4.10). It is easy to see from the Fig. 4.9 that 90% performance is accomplished when using only 40% of the original dictionary size and 80% performance is obtained by using 30%. Similarly, Fig. 4.10 shows that the computational complexity of the sparsity model reduces with a pruned dictionary. By applying our dictionary pruning algorithm, if we use 40% of the columns, we can retain 90% performance with a cost of only 30% running time.
Figure 4.10. Running time with respect to the proportion of original dictionary size - CAVIAR data set.

Experimental results on both data sets show that significant computational advantages can be obtained with the proposed techniques with little performance loss over using a large over complete basis of example trajectories.

4.3.4.3 Unsupervised dictionary initialization

In this part of the experiments, we use the trajectory clustering technique from [54] to initialize our training dictionary (unsupervised case). We used $k = 2$ clusters.

Table 4.10. Confusion matrices on Xerox Stop Sign data: supervised and unsupervised cases - Dictionary Learning

<table>
<thead>
<tr>
<th></th>
<th>Hand Labelled (with 37.5% columns)</th>
<th>Automatically Labelled (with 37.5% columns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>84.2%</td>
<td>14.3%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>15.8%</td>
<td>85.7%</td>
</tr>
</tbody>
</table>

Table 4.10 and 4.11 compare the confusion matrices obtained for unsupervised versus supervised dictionary initialization for Xerox Stop Sign data and CAVIAR
Table 4.11. Confusion matrices on CAVIAR data: supervised and unsupervised cases - Dictionary Learning

<table>
<thead>
<tr>
<th></th>
<th>Hand Labelled (with 40.0% columns)</th>
<th>Automatically Labelled (with 40.0% columns)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
</tr>
<tr>
<td>Normal</td>
<td>83.3%</td>
<td>27.3%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>16.7%</td>
<td>72.7%</td>
</tr>
</tbody>
</table>

data respectively. These two tables indicate that clustering technique will only cause a slight performance drop in detection rates, while alleviating the burden of manual labeling.

### 4.4 Conclusion

This chapter presents two practical methods for sparsity based video anomaly detection. First, we propose a kernelization of the joint sparsity model so as to further improve anomaly detection where linear sparse reconstruction models are not appropriate. The kernel function projects the non-separable trajectories into a nonlinear feature space in which the trajectories becomes more separable. So, the trajectories used in joint sparsity model differs from the original trajectories. Second, we propose a dictionary design and optimization technique that can effectively reduce the size of training dictionaries that enable sparsity based classification/anomaly detection without adversely influencing detection performance. This chapter is focused on how to build the best dictionary using trajectory features. Another idea is, in stead of trajectory, another feature which is more separable itself may lead to better results. Or the combination of different features may fit our sparsity models better. Therefore, in the next chapter, we try to exploit the diversity of multiple features.
Simultaneous Sparsity Model for Multi-perspective Video Anomaly Detection

5.1 Introduction

In Chapter 3 and 4, we built sparsity model upon trajectory features. As discussed above, a variety of event encoding techniques have been proposed for the problem of video anomaly detection. Most video anomaly detection techniques use only single event encoding, thus fail to exploit diversity in these multiple perspectives on an event. In this chapter, we propose to extend sparsity models based on single feature representations to simultaneous sparse representations of multiple feature representations.

The precursor to anomaly detection is an effective encoding of events in the video sequence. The purpose of event encoding is to extract features from the video that are most useful in differentiating among different events. A variety of
event encoding techniques have been proposed for the problem of video anomaly detection, including: trajectory extraction and encoding [1], features from local spatio-temporal volumes [11, 22] and motion labels [3]. A typical anomaly detection approach uses one of these event representations with a corresponding decision rule. It is not difficult to see that the distinct event feature representations contain correlated yet complementary information about the event (normal or anomalous). To exploit the diversity in these multiple perspectives on an event, we extend existing sparse coding based anomaly detection techniques to a simultaneous sparse coding model where the test observation is a matrix whose columns are distinct (complementary) encodings of the same underlying event, i.e., corresponding feature vectors. Given multiple training event representations in a dictionary, the identity of the test event is now encoded by the sparse structure of the coefficient matrix. It is this structure that we first quantify and then exploit for classifying events. We also show that in the absence of training from anomalous events, i.e. unsupervised anomaly detection, outlier rejection measures on the sparse coefficient matrix can be used with relatively little drop in performance.

5.2 Motivation: Multi-Perspective

Most existing approaches uses only one of event representations with a corresponding decision rule (e.g. sparsity model). It is not difficult to see that the distinct event feature representations contain correlated yet complementary information about the event (normal or anomalous). To exploit the diversity in these multiple perspectives on an event, we claim that complementary information from different event encodings can help reveal its identity (normal/anomalous), and hence our goal is to effectively capture their interactions via a new simultaneous sparsity
Figure 5.1. Structure of simultaneous sparsity model and its two major applications: (1) multi-feature video anomaly detection; (2) multi-camera video anomaly detection.

model as shown in Fig. 5.1. In simultaneous sparsity model, we take advantage of different event encodings. Fig. 5.1 shows the basic structure of the simultaneous sparsity model as well as its two major applications: (1) multi-feature video anomaly detection; (2) multi-camera video anomaly detection. We will focus on multi-feature video anomaly detection in this chapter.

In multi-feature setting, an event can be represented by a matrix \( Y = \{ y_i \} \in \mathbb{R}^{N \times M}, i = 1, \ldots, M \), where \( y_i \in \mathbb{R}^N \) is the \( i \)-th feature vector or the \( i \)-th way of representing the event. The training dictionary can be defined as: \( A = [A_1 \ A_2 \ \ldots \ A_M] \in \mathbb{R}^{N \times MKT} \), where each dictionary \( A_i = [A_{i,1} \ A_{i,2} \ \ldots \ A_{i,K}] \in \mathbb{R}^{N \times KT}, i = 1, 2, \ldots, M \), is formed by the concatenation of training sub-dictionaries from all classes belonging to the \( i \)-th event encoding. Next, the test features are expected to be represented as a sparse linear combination of training features: \( Y \approx AX \), where \( X \in \mathbb{R}^{MKT \times M} \) is the sparse coefficient matrix.
Figure 5.2. (a) Row sparsity model: sparse coefficients in the same row should be activated simultaneously and (b) Simultaneous sparsity model: the sparse coefficient in the same class should be activated simultaneously.

The most common way of quantifying matrix sparsity $X$ is:

$$\begin{align*}
\text{minimize} & \quad \|X\|_{\text{row,0}} \\
\text{subject to} & \quad \|Y - AX\|_F \leq \epsilon,
\end{align*}$$

(5.1)

where $\|\|_{\text{row,0}}$ refers to the number of non-zero rows in the matrix. Fig. 5.2(a). shows an example of such a row sparsity model which has demonstrated success [58, 59] in many multi-task image classification problems. Crucially, efficient algorithmic techniques such as the simultaneous orthogonal matching pursuit (SOMP) have been developed [60] to solve the problem in (5.1).

5.3 Simultaneous Sparsity Model for Single-object Video Anomaly Detection

While the row-sparse structure is widely used, it is really not physically meaningful for our set-up. This is because while each feature representation is expected to be a sparse linear combination of training samples from the dictionary, the location
of ‘active’ or non-zero coefficients do not necessarily belong to the same row. For accurate event classification, it is expected that $M$ test features are effectively approximated by a linear combination of training samples within the same event class. That said, because the $M$ feature representations are distinct, the particular training samples they pick within each class may be different. In other words, $M$ features vector have the same sparsity pattern at a class level, but not necessarily at the level of training within a class. We therefore need to relax the row-sparsity restriction. Fig. 5.2(b) illustrates the sparse structure inherent to our coefficient matrix.

Instead, we propose the use of active curves to measure class level sparsity in our simultaneous sparsity model [61]. An active curve $c \in \mathbb{R}^M$ is defined to be the indexes of $M$ coefficients corresponding to $M$ features. Each active curve $c_i$ constrains the sparse coefficients of $X$ that must be activated along a certain 1-D path, where the $j$-th element of $c_i(j)$ forces that in the $j$-th column of $X$ only the $c_i(j)$ coefficient can be activated. For example in Fig. 5.2(b), the first active curve is $c_1 = [2, 3, ..., 1]$. Typically $L$ active curves with $L << N$ are expected, where $L$ can be regarded as the sparsity level of matrix $X$. More importantly, these active curves must not be fixed apriori but instead adaptively selected based on the test feature matrix to allow for effective reduction of the reconstruction error. Based on these active curves, a simultaneous norm term may be defined as follows:

$$
\|X\|_C = \|\|x_{c_1}\|_{\ell_2}, \|x_{c_2}\|_{\ell_2}, ..., \|x_{c_L}\|_{\ell_2}\|^T\|_{\ell_0}
$$

(5.2)

where $x_{c_i}$ denotes the $i$-th row vector associated with the $i$-th active curve $c_i$. We first calculate $\ell_2$ norm on each active curve and the apply $\ell_0$ norm over the $\ell_2$ norm of the active curves. $\|X\|_C$ means the number of non-zero active curves. In stead
of minimizing number of non-zero rows in row sparsity model, we minimize the number of non-zero active curves. The optimization problem becomes:

\[
\text{minimize} \quad \|X\|_C \\
\text{subject to} \quad \|Y - AX\|_F \leq \epsilon.
\]

(5.3)

This kind of norm is called “matrical” norm, which has been successfully applied in the field of image restoration [62].

### 5.4 Simultaneous Structured Sparsity Model for Multi-object Video Anomaly Detection

We are also interested in detection of anomalies involving \( P \geq 1 \) objects. Their corresponding \( P \) objects can be represented as a matrix: \( Y = [Y^1 \ Y^2 \ \ldots \ Y^P] \in \mathbb{R}^{N \times MP} \), where \( Y^j \) correspond to \( j^{th} \) trajectory. The training dictionary can be defined as: \( A = [A^1 \ A^2 \ \ldots \ A^P] \in \mathbb{R}^{N \times MKTP} \), where each dictionary \( A^j = [A^j_1 \ A^j_2 \ \ldots \ A^j_M] \in \mathbb{R}^{N \times MKT}, j = 1, 2, \ldots, P \), is formed by the concatenation of the sub-dictionaries from all classes belonging to the \( j^{th} \) object. The crucial aspect of this formulation is that the training trajectories for any class \( j \) are observed “jointly” from example videos.

The test \( P \) objects can now be represented as a linear combination of training samples as follows:

\[
Y \approx AX = [A^1 \ A^2 \ \ldots \ A^P][X^1 \ \ldots \ X^P],
\]

where the coefficient matrices \( X^j \) lie in \( \mathbb{R}^{MKTP \times M} \) and \( X = [X^1 \ \ldots \ X^j \ \ldots \ X^P] \).
It is important to note that the $j$-th object trajectory of any observed set of test trajectories should *only lie* in the span of training trajectories corresponding to the $j$-th object. Therefore, the columns of $X$ should have the following structure:

$$
X^1 = \begin{bmatrix}
\bar{X}^1 \\
0 \\
0 \\
\vdots \\
0
\end{bmatrix},
X^j = \begin{bmatrix}
0 \\
\vdots \\
\bar{X}^j \\
0
\end{bmatrix},
X^P = \begin{bmatrix}
0 \\
\vdots \\
0 \\
\bar{X}^P
\end{bmatrix}.
$$

(5.4)

where each of the sub-matrices $\bar{X}^j, j = 1, 2, \ldots, P$ lies in $\mathbb{R}^{M_{KT} \times M}$, while $0$ denotes a matrix of all zeros in $\mathbb{R}^{M_{KT} \times M}$. As a result, $X$ exhibits a block-diagonal structure.

When $P = 1$, $X = \bar{X}^1$, the sparsity structure in Eq. 5.4 reduces to single-object simultaneous sparsity model in Eq. 5.3, and $\bar{X}^1$ has the simultaneous structured sparsity properties in Fig. 5.2(b). Therefore, the single-object model is extended to multi-object model, we expect all $\bar{X}^j, j = 1, \ldots, P$ have the properties of simultaneous sparsity model. Furthermore, we want all the $\bar{X}^j, j = 1, \ldots, P$ have some “joint” attributes. We assume that training samples with non-zero weights (in the sparse linear combination) exhibit one-one correspondence across different features. In other words, if the $j$-th object training sample from the $i$-th class is chosen for the $j$-th object test, then it is necessarily that other $P-1$ objects choose from $i$-th class with very high probability, albeit with possibly different weights. This can be observed more clearly in Fig. 5.3

From Fig. 5.3, we can see that, inspired by our joint sparsity model in [24], correspondent coefficients from different objects should be activated simultaneously.
Figure 5.3. Structure of simultaneous sparsity model for multi-object: active curves of different objects should have the same shape.

Therefore, active curves of different objects should have the same shape.

Our optimization problem becomes:

\[
\begin{align*}
\text{minimize} & \quad \|\bar{X}^1\|_C \\
\text{subject to} & \quad \|Y - AX\|_F \leq \epsilon \\
& \quad \|\bar{X}^j\|_C = \|\bar{X}'\|_C \quad \text{for any } j \neq j'.
\end{align*}
\]  \hspace{1cm} (5.5)

Since the active curves are chosen adaptively in [61]. We can extend the algorithm to fit the multi-object scenario by “jointly” select active curves among objects. It can be simplified to the following problem:

\[
\begin{align*}
\text{minimize} & \quad \|Y - AX\|_F \\
\text{subject to} & \quad \|ar{X}\|_C \leq L \\
& \quad \|ar{X}^j\|_C = \|ar{X}'\|_C \quad \text{for any } j \neq j'.
\end{align*}
\]  \hspace{1cm} (5.6)
where $L$ is the upper bound of the number of active curves. In Eq. 5.6, we are trying to minimize the reconstruction error with the constraints that using at most $L$ active curves on the coefficient matrix of each object.

We propose to solve problem Eq. 5.6 by a greedy algorithm referred to as SSM, which is described in detail in Algorithm 3. The proposed algorithm has a similar algorithmic structure as SOMP [60], which includes the following general steps:

1. Select new candidates based on the current residue.
2. Merge the newly selected candidate set with previous selected atom set.
3. Estimate the representation coefficients based on the merged atom set.
4. Prune the merged atom set to a specified sparsity level based on the newly estimated representation coefficients.
5. Update the residue. This procedure is iterated until certain conditions are satisfied.

At each iteration of SSM, there is a procedure called the active curve mapping (ACM) as follows:

$$ I_L = \mathbb{P}_{ACM}(X, L) \quad (5.7) $$

which provides an index matrix $I_L \in \mathbb{R}^{L \times MP}$ for all the $M$ features and $P$ objects corresponding to the top-$L$ active curves. The implementation of this mapping is detailed in Algorithm 4. In each iteration of ACM, it will select a new active curve. This is achieved via three steps:

1. For each feature, find the best $P$ coefficients for each class. Since, correspondent coefficients from different objects should be activated simultaneously.
Algorithm 3 Simultaneous Sparsity Model

**Input:** Dictionary $\mathbf{A} = [\mathbf{A}^1 \mathbf{A}^2 \ldots \mathbf{A}^P] \in \mathbb{R}^{N \times MKP}$, data matrix $\mathbf{Y} = [\mathbf{Y}^1 \mathbf{Y}^2 \ldots \mathbf{Y}^P] \in \mathbb{R}^{N \times MP}$, sparsity level $L$

Initialization: residual $\mathbf{R} = \mathbf{Y}$, index set $\mathbf{I}$: empty set

while stopping criterion has not been met do

1. Compute correlation matrix: $\mathbf{E} = \mathbf{A}^T \mathbf{R}$

2. Select atoms via joint active curve mapping: $\mathbf{I}_{new} = \mathbb{P}_{ACM}(\mathbf{E}, 2L)$

3. Update index matrix: $\mathbf{I} = [\mathbf{I}^T, \mathbf{I}_{new}^T]^T$, and set $\mathbf{B} = \mathbf{0}$

4. Prune atoms via active curve mapping: $\mathbf{I} = \mathbb{P}_{JDS}(\mathbf{B}, L)$ and $\mathbf{X} = \mathbf{0}$

5. Update representation coefficients with the selected atoms:

   for $j = 1, 2, \ldots, MP$ do
   
   $i = \mathbf{I}(::, j)$ (where $\mathbf{I}(::, j)$ denotes $j$-th column of $\mathbf{I}$, $i$ is a set of indices)
   
   $\mathbf{B}(i, j) = (\mathbf{A}(::, i)^T \mathbf{A}(::, i))^{-1} \mathbf{A}(::, i)^T \mathbf{Y}(::, j)$
   
   end for

end while

**Output:**

$\mathbf{X} = \mathbf{0}$

for $j = 1, 2, \ldots, MP$ do

$i = \mathbf{I}(::, j)$

$\mathbf{X}(i, j) = (\mathbf{A}(::, i)^T \mathbf{A}(::, i))^{-1} \mathbf{A}(::, i)^T \mathbf{Y}(::, j)$

recalculate the coefficients with the selected atoms

end for

We define the best $P$ coefficients to be:

$$
\arg \max_{m \in \{1, 2, \ldots, M\}} \sum_{j=1}^{P} \tilde{X}^j(c, m) \quad (5.8)
$$

where $c$ is the index vector for the $k$-th class.

2. Combine the maximum absolute coefficients across the features for each class as the total response
Algorithm 4 Active Curve Mapping $\mathcal{P}_{ACM}(X, L)$

**Input:** Coefficient matrix $X$, desired number of active curves $L$, number of classes $K$, number of features $M$

Initialization: residual: index matrix $I_L = 0$

for $l = 1, 2, ..., L$ do

for $k = 1, 2, ..., K$ do

Get the index vector for the $k$-th class:
$c = \text{find}(X, i)$

for $m = 1, 2, ..., M$ do

1. Find the index $t$ and the value $v$ of the best $P$ coefficients for the $k$-th class, $m$-th feature

$$[v, t] = \max_{m \in \{1, 2, ..., M\}} \sum_{j=1}^{P} X_j^l(c, m)$$

$$V(k, m) = v, \quad \tilde{I}(k, m) = c(t)$$

end for

2. Combine the max-coefficients for each class

$$s(k) = \sqrt{\sum_{m=1}^{M} V(k, m)^2}$$

end for

3. Find the best cluster of atoms belonging to the same class across all classes:

$$[\hat{v}, \hat{t}] = \max(s)$$

$$I_L(l, :) = \tilde{I}(t, :) \quad X(\tilde{I}(t, :)) = 0$$

end for

**Output:**
Index matrix $I_L$ for the top-$L$ active curves

3. Select the active curve as the one that gives the maximum total response.

After a active curve is determined, we keep a record of the selected indexes as a row of $I_L$ and set their associated coefficients in the coefficient matrix to be zero in order to ensure that none of the coefficients will be selected again.
5.5 Experimental Validation

We evaluate our algorithms by testing on several real transportation data sets for structured scenario. For structured scenario, the training dictionary is well labeled with class information. We detect anomalies using simultaneous sparsity model (abbreviated to SSM) to take advantage of structure information in the dictionary. Additionally, both supervised and unsupervised scenarios are used in testing. In the supervised case, training dictionaries contain both example normal and anomalous features. In the unsupervised case, only normal event features are available for training and the aforementioned outlier rejection measure in Eq. (3.17) is used for anomaly detection.

In our multi-feature setting, object trajectories and spatio-temporal volumes (STVs) are used to represent events. We use well-known techniques to extract trajectories [42] as a collection of coordinate pairs \([x(t), y(t)]\). We approximate a raw trajectory using a basic B-spline function [43] with 50 knots (50 x-coordinates and 50 y-coordinates) and these knots are finally used to form the trajectory feature vector. There have been several techniques in extracting STVs. In this chapter, we extract STVs using the method in [11]. Three groups of features are considered in each STV: 1). average position and size of the STV, 2). intrinsic activity of the STV and 3). interaction occurring among different STVs.

5.5.1 Experimental Setup

As discussed above, simultaneous sparsity model can deal with both single-object and multi-object anomalies. For single-object anomalies, we test on CAVIAR data set (an example video clip is available at: http://youtu.be/M6_PJigg5CY) and Xerox Stop Sign data set [44]. Figs. 5.4 (a) show an example anomaly in CAVIAR
data set, a man suddenly falls on the floor (see Fig. 5.4(a2)) when walking across the lobby. Figs. 5.4 (b) show an example anomaly in CAVIAR data set, instead of turning left or right in front of the stop sign, the driver suddenly backs his car - see Figs. 5.4 (b3)-(b4). We compare our experimental results against three well-recognized techniques in single-object anomalous event detection: 1.) the recent sparsity based technique of Li et al. [21], 2.) the approach of Piciarelli et al. [1] using one class SVMs and 3.) the decision tree model using spatio-temporal volumes Simon et al. [11].

For multi-object anomalies, we test on Xerox Intersection data set. A representative video clip is made available at: http://youtu.be/ZGKtkVtWEFU. Figs. 5.4 (c) show an example anomaly in Xerox Intersection data set, a car fails to yield to oncoming car while turning left - see Figs. 5.4 (c2)-(c3). We also compare our experimental results against another three video anomaly detection methods that can deal with multi-object anomalies: 1.) the sparsity model with online dictionary learning of Zhao et al. [22], 2.) the multiple-object tracking and rule-based anomaly detection technique of Han et al. [2] and 3.) our work - joint sparsity model [24].

5.5.2 Single-Perspective Single-Object Anomaly Detection

We compare three state-of-the-art methods: Piciarelli et al. [1], Simon et al. [11] and Li et al. [21]. The method in Piciarelli et al. [1] is based on trajectory features combined with support vector machines for decisions. Simon et al. applies decision tree on spatio-temporal volumes to identify anomalies. Li et al. detects anomalies using the single perspective sparsity model in Chapter 2. For the experiment involving the Xerox Stop Sign data set, 118 events from 39 video clips are extracted.
Figure 5.4. Example anomalies in structured scenario: (a) a man suddenly falls on floor – from the CAVIAR data set, (b) a driver backs his car in front of stop sign – from the Xerox Stop Sign data set and (c) a car fails to yield to oncoming car while turning left – from the Xerox Intersection data set.

The training dictionary comprises 9 normal event classes (containing 8 events each) and 1 anomalous event class (containing 4 events). Because training for anomalous events is well-represented in this database, Eq. (2.4) is used to classify the test trajectories as normal or anomalous. An independent set of 34 normal events and 8 anomalous events are used to test our approach. We test in a supervised scenario, i.e. training from normal as well as anomalous events is made available to all the methods. Table 5.1 shows the confusion matrices of three approaches. Note that all Li et al. which is a sparsity based anomaly detection technique do better than Piciarelli et al. and Simon et al.. The benefits of classification using sparsity model vs. SVM-based classifier and decision tree based method are readily apparent.
Table 5.1. Confusion matrices of three state-of-the-art video anomaly detection methods on the Xerox Stop Sign data—supervised, single-object anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>Piciarelli et al. [1]</th>
<th>Simon et al. [11]</th>
<th>Li et al. [21]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>85.3%</td>
<td>37.5%</td>
<td>82.4%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>14.7%</td>
<td>62.5%</td>
<td>17.6%</td>
</tr>
</tbody>
</table>

5.5.3 Multi-Perspective Single-Object Anomaly Detection

We then test our proposed simultaneous sparsity model (SSM) on CAVIAR data set, where we use trajectory and STV features together, i.e. the SSM framework is instantiated with $M = 2$. Specifically for SSM, Eq. (3.15) is used to classify the test event as normal or anomalous. For the CAVIAR data set, we test on 27 video clips from which 170 events are extracted. Our training dictionary consists of 10 normal event classes and 3 anomalous event classes, each class contains 10 different training events. 21 normal events and 19 anomalous events are used as independent test data. We compare SSM against a majority voting technique for decision fusion to identify anomalies. Detection rates of SSM against three competing single-perspective techniques (Piciarelli et al., Simon et al. and Li et al.) and the majority vote of these three techniques are shown in Table 5.2. First, we note the benefits of sparsity based anomaly detection in the form of improved detection rates of Li et al and Zhao et al. over Piciarelli’s approach. Second, it is easy to see from Table 5.2 that SSM expectedly improves upon the results of using single-perspective techniques by virtue of exploiting feature diversity in encoding an event. Third, Remarkably, SSM is also better than the majority voting technique which demonstrates that exploiting interactions at the level of features is superior to decision fusion.
Table 5.2. Confusion matrices of proposed and state-of-the art video anomaly detection methods on CAVIAR data – supervised, single-object anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>Piciarelli et al. [1]</th>
<th>Simon et al. [11]</th>
<th>Li et al. [21]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>85.7%</td>
<td>26.3%</td>
<td>85.7%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>14.3%</td>
<td>73.7%</td>
<td>14.3%</td>
</tr>
</tbody>
</table>

Table 5.3. Confusion matrices of proposed and state-of-the art video anomaly detection methods on the Xerox Intersection data – supervised, multi-object anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>82.4%</td>
<td>35.7%</td>
<td>88.2%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>17.6%</td>
<td>64.3%</td>
<td>11.8%</td>
</tr>
</tbody>
</table>

5.5.4 Multi-Perspective Multi-Object Anomaly Detection

In Xerox Intersection data, there are 91 events extracted from 13 video clips. We manually build our training dictionary into 6 different 2-object normal event classes (containing 6 event each) and 6 different anomalous classes(containing 4 event each). 17 normal events and 14 anomalous events are used for testing.

For multiple (here 2) object anomaly detection, we will first compare against Han et al. [2], JSM [24] in supervised setting. The confusion matrices of our method against the two competing techniques are shown in Table 5.3. It is easy to see from Table 5.3 that the proposed SSM method leads to the best detection rates. In Han et al, anomalies are detected by the use of context based rules on the result of multiple-object tracking. This puts an unreasonable burden on defining these rules and is often restrictive in practice, i.e. not all anomalies can be anticipated. For JSM, interactions between distinct object trajectories are better captured and departures from expected “joint behavior” (particularly in the case
Table 5.4. Confusion matrices of proposed and state-of-the-art video anomaly detection methods on the Xerox Intersection data – unsupervised, multi-object anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>Zhao et al. [22]</th>
<th>JSM [24]</th>
<th>SSM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>76.5%</td>
<td>35.7%</td>
<td>82.4%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>23.5%</td>
<td>64.3%</td>
<td>17.6%</td>
</tr>
</tbody>
</table>

training for anomalies is absent/limited) is employed which enables the improvement in detection rates over Han et al. More importantly, SSM outperforms Han et al and JSM because of capturing the diversity of different event encodings.

5.5.5 Multi-Perspective Multi-Object Anomaly Detection in Unsupervised Setting

We then compare against proposed SSM against Zhao et al. [22] and JSM in unsupervised setting. In this case, we do not use training from anomalous events and perform anomaly detection instead by using the outlier rejection measure in Eq. (3.17). This scenario is of immense practical significance since training from anomalous events is often not available in real-world applications. The confusion matrices of our method against the other two methods are shown in Table 5.4. Again, SSM performs best in terms of detection rates owing to exploiting the diversity in these multiple perspectives (trajectory and STVs) on an event.

Finally, we report statistically relevant results in the form of receiver operating characteristic (ROC) curves on Xerox Intersection data in Fig. 5.5. It is easy to see from Fig. 4 that the SSM curve consistently lies above competing techniques and hence can facilitate a superior detection vs. false alarm trade-off. Remarkably, SSM-unsupervised is quite close to SSM-supervised and as the ROC curves in Fig. 5.5 reveal, SSM-unsupervised still outperforms even supervised methods that use
Figure 5.5. ROC curves for multiple-object anomaly detection – Xerox Intersection data set.

a single feature representation.

5.6 Conclusion

This chapter proposes to exploit complementary merits of different feature representations of an event via a simultaneous sparse coding method. Experiments on the benchmark databases reveal that significantly improved performance in event classification/anomaly detection can be achieved via the use of such a model over existing methods that use a single feature representation. Simultaneous sparse representations as advocated in our work can in fact be applied to other multiperspective anomaly detection scenarios such as features from multiple camera views of an event.
Low Rank Sparsity Prior for Unstructured Video Anomaly Detection

6.1 Introduction

In Chapters 3 - 5, our sparsity based video anomaly detection methods show promise but open challenges remain in that existing methods assume object specific and class specific event dictionaries making them applicable mostly in highly structured scenarios. Second, using conventional sparsity models on matrices/vectors, the computational burden is often high. In this work, we advocate a more general and practical sparsity model using a low-rank structure on the matrix of sparse coefficients. We find that enforcing a low-rank structure can ease the rigidity of traditional row-sparse constraints on sparse coefficient vectors/matrices. Because low-rank matrices are of course not always sparse, an additional $l_1$ regularization term is added. Further, if rank is substituted by its convex nuclear norm
alternative, then significant computational benefits can be obtained over existing methods in sparsity based video anomaly detection. Experimental evaluation on benchmark video datasets reveal, our method is competitive with state-of-the art while providing robustness benefits under occlusion.

6.2 Motivation: Low Rank Sparsity Prior

In the previous chapter, we propose a well-structured simultaneous sparsity model which captures the diversity of different event encodings and is capable of detecting multi-object anomalies. The aforementioned set-up in Eq. (5.3) - Eq. (5.6) assumes that training is available from both normal and anomalous events and hence anomaly detection reduces to a classification problem. In the absence of training from anomalous events (the more practical scenario), outlier rejection measures in Eq. (3.17) may be used to detect anomalies.

In order to build such structured sparsity model, A careful preparation of the dictionary $A$ is needed often with training examples that are manually labeled to belong to particular event classes. Multi-object or multi-view anomaly detection
leads to a sparse coefficient matrix (and not vector), in those cases training dictionaries are often labeled not only per class but also per object [21], [24] leading to a row-sparse structure for the coefficient matrix. Such elaborate preparation of the dictionary is sometimes unrealistic and invariably burdensome requiring a pre-analysis of video footage prior to anomaly detection. Fig. 6.1(a) (the video is available at: http://youtu.be/M6_PJigg5CY) shows an example video frame of a structured scenario (detection of stop sign violations) where preparation of a dictionary clearly separated into class-specific sub-dictionaries is possible. In many other settings however, multiple objects and features are simultaneously extracted and a clear separation into normal event classes is difficult. An example of a video frame from such a scenario is shown in Fig. 6.1(b) (the video is available at: http://youtu.be/jEzLkWF65Io).

We therefore seek a more general and practical sparsity prior which can deal with unstructured scenarios when we do not have a well-classified training dictionary. Note that, row-sparse matrices that have been used in [22], [24] for anomaly detection are also low-rank. Inspired by this observation and known connections between low-rank and sparse matrices [63], we propose using a low rank sparsity prior. In addition, if rank is substituted by its convex nuclear norm alternative, then significant computational benefits can be obtained over existing methods in sparsity based video anomaly detection.
6.3 Low Rank Sparsity Prior for Anomaly Detection

In unstructured scenario, a well-structured dictionary is not available. All observed training events corresponding to normal events are collected together as a big dictionary: $A \in \mathbb{R}^{N \times T}$. The test matrix $Y$ contains $Q$ events (we choose to deal with $Q$ events simultaneously instead of handling each event individually because we want the coefficient matrix $S$ to be close to a square matrix where low rank sparsity prior can play more important role in optimization problem).

$Y = \{Y_i\} \in \mathbb{R}^{N \times M}, i = 1, \ldots, Q$, where $Y_i \in \mathbb{R}^{N \times M_i}$ and $\sum_{i=1}^{Q} M_i = M$.

Under a linear model $Y \approx AS$, and given sufficient training, the coefficient matrix $S \in \mathbb{R}^{T \times M}$ is expected to be sparse. Making a departure from the typical $\|\|_{\text{row,0}}$ norm, we propose to use a low-rank structure to measure the sparsity of $S$.

A couple of examples of simultaneously sparse and low rank matrices are:

$$S_1 = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix} \quad S_2 = \begin{pmatrix}
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Then, we propose to replace (2.3) by:

$$\begin{align*}
\text{minimize} & \quad \text{rank}(S) \\
\text{subject to} & \quad \|Y - AS\|_F \leq \epsilon,
\end{align*}$$

(6.1)

A convex relaxation of (6.1) can be obtained via substituting $\text{rank}(X)$ by $\|X\|_* = \sum_i \sigma_i(X)$ [64] (where $\|\|_*$ denotes nuclear norm and $\sigma_i(X)$ is the $i$-th
singular value of $X$) [65]. This results in a convex optimization problem:

$$
\begin{align*}
\text{minimize} & \quad \|S\|_* \\
\text{subject to} & \quad \|Y - AS\|_F \leq \epsilon.
\end{align*}
$$ (6.2)

While low-rank and sparse matrix structures often simultaneously exist (as is expected here as well), in general the two are not the same, and low-rank does not imply sparsity.

To encourage sparse matrices which are simultaneously low-rank, we further add a $l_1$ regularization term to the cost function and convexity still holds:

$$
\begin{align*}
\text{minimize} & \quad \|S\|_* + \lambda\|S\|_1 \\
\text{subject to} & \quad \|Y - AS\|_F \leq \epsilon.
\end{align*}
$$ (6.3)

where $\|S\|_1 = \sum |s_{ij}|$, sum of absolute value of every element in $S$.

**Anomaly Detection:**

Once we get the optimal coefficient matrix $\hat{S}$, the recovered events on $Y$ can be computed using sub-matrix of $\hat{S} = \{\hat{S}_1, ..., \hat{S}_Q\}$:

$$\hat{Y}_i = A\hat{S}_i,$$ (6.4)

if the test event $Y_i$ are very similar to the recovered event, then this test event can be regarded as normal event.

$$\frac{\|Y_i - \hat{Y}_i\|_F}{\|Y_i\|_2} < \tau \rightarrow Y_i \text{ is normal.}$$ (6.5)

**Computational Complexity:**

The problem in (6.3) can in fact be cast as a semi-definite program (SDP),
which is tractable and can be solved efficiently in polynomial time.

Define $\|s\|$ to be spectral norm of a matrix: $\|S\|_s = \sigma_{\text{max}}(S)$. The spectral norm is the dual norm of the nuclear norm. We have:

$$\|S\|_s = \max\{\text{trace}(S^T E) : \|E\|_s \leq 1, E \in \mathbb{R}^{T \times M}\}. \quad (6.6)$$

The spectral norm can be written into a SDP problem [66]:

$$\|E\|_s = \min_t \quad s.t. \quad \begin{pmatrix} tI_T & E \\ E^T & tI_M \end{pmatrix} \succeq 0. \quad (6.7)$$

where $\succeq 0$ denotes positive semi-definite.

If the SVD of $S$ is: $S = U \Sigma V^T$, where $U \in \mathbb{R}^{T \times R}$, $\Sigma = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_R) \in \mathbb{R}^{R \times R}$, $U \in \mathbb{R}^{M \times R}$. From duality, we can obtain the following SDP characterization of the nuclear norm:

$$\|S\|_* = \min_{W_1, W_2} \frac{1}{2}(\text{trace}(W_1) + \text{trace}(W_2))$$

subject to $\begin{pmatrix} W_1 & S \\ S^T & W_2 \end{pmatrix} \succeq 0. \quad (6.8)$

where $W_1 = U \Sigma U^T \in \mathbb{R}^{T \times T}$ and $W_2 = V \Sigma V^T \in \mathbb{R}^{M \times M}$.

The $l_1$ norm term has the following SDP characterization:

$$\|S\|_1 = \min_Z 1^T_T Z 1_M$$

subject to $-Z \preceq S \preceq Z. \quad (6.9)$

where $1_T \in \mathbb{R}^T$ refers to the vector that has 1 in every entry.
The quadratic constraint can be formulated as [67]:

\[ \|Y - AS\|_F \leq \epsilon \iff \begin{pmatrix} 1_N & Y - AS \\ (Y - AS)^T & \epsilon I_M \end{pmatrix} \succeq 0. \quad (6.10) \]

Therefore, the Eq. (6.3) can be written into a SDP as:

\[
\begin{align*}
\min_{W_1, W_2, Z, S} & \quad \frac{1}{2} \left( \text{trace}(W_1) + \text{trace}(W_2) \right) + \lambda Z^T Z M \\
\text{subject to} & \quad \begin{pmatrix} W_1 & S \\ S^T & W_2 \end{pmatrix} \succeq 0 \\
& \quad -Z \preceq S \preceq Z \\
& \quad \begin{pmatrix} 1_N & Y - AS \\ (Y - AS)^T & \epsilon I_M \end{pmatrix} \succeq 0
\end{align*} \quad . \quad (6.11)
\]

The Eq. (6.11) is a SDP, it can be solved using any off-the-shelf interior point solver (e.g. CVX [68]). CVX solves such SDP using the technique from [69]. The authors are trying to solve the rank minimization problem in Eq. (1.1) from [65], which has been formulated into a semidefinite program (SDP) [66]:

\[
\begin{align*}
\min_{W_1, W_2, Z, C, D} & \quad \frac{1}{2} \left( \text{trace}(W_1) + \text{trace}(W_2) \right) + \lambda Z^T Z M \\
\text{subject to} & \quad \begin{pmatrix} W_1 & C \\ C^T & W_2 \end{pmatrix} \succeq 0 \\
& \quad -Z \preceq D \preceq Z \\
& \quad C + D = B
\end{align*} \quad . \quad (6.12)
\]

This semi-definite program can then be solved using a “custom” interior point method [70], [69] and has an average complexity of \(O(N^2TM)\) where \(N, T, M\) are
as stated above.

Several fast $l_1$-minimization algorithms have been published [71]. The Homotopy method is amongst the most popular algorithms and has a computational complexity of its $j$-th iteration as $O(jN^2 + jNT)$ [72]. Let $J$ denote the number of iterations, the total complexity becomes: $O(\sum_{j=1}^{J}(jN^2 + jNT)) = O(J^2N^2 + J^2NT)$. Here, $J$ depends on the number of non-zero elements in $\alpha$, so $O(J) = O(N)$. Therefore, the computational complexity of evaluating one test trajectory (event representation) using (2.3) is $O(N^4 + N^3T)$. Since there are $M$ trajectories, the total computational complexity is $O(N^4M + N^3TM)$. The proposed method hence has much lower complexity.

6.4 Experimental Result

We evaluate our algorithms by testing on several real transportation data sets for structured scenario. For unstructured, scenario, class-specific sub-dictionaries are not available. Simultaneous sparsity model is not applicable here, therefore, low rank sparsity prior is used to identify anomalies. In this case, all the normal events are group as a dictionary. Since, we do not have the training for anomalies, only unsupervised case are tested for unstructured video anomaly detection.

6.4.1 Experimental Setup

In unstructured scenario, a well-constructed and class-specific training dictionary is not available. We can not use simultaneous sparsity model since SSM requires the information of dictionary structure. In this case, low rank sparsity prior can be used to detect anomalies. Figs. 6.2 (a) and (b) show consecutive frames from 2 example anomalies in unstructured scenario. In Figs. 6.2 (a), a pedestrian walk
crossing the street loses his hat and retraces his footsteps to pick it up from the road. At this time, a vehicle comes in very close proximity to the pedestrian and comes to a sudden halt - see Fig. 6.2 (a2). In Fig. 6.2 (b), a car fails to yield to oncoming car while turning left - see Figs. 6.2 (b2)-(b3).

6.4.2 Comparison against state of the art Trajectory-based Video Anomaly Detection Techniques

For unstructured sparsity scenario, we test our proposed low rank sparsity prior (LRSP) on a well-known Public Data Set for Traffic Video (PDTV) [73], which is a real surveillance video data set with challenging real and anomalous events. Figs. 6.2 (b2) and (b3) show an example anomaly in PDTV data, where a car fails to yield to oncoming car while turning left. For the experiment involving the PDTV data set, we obtain a training dictionary consisting of 319 normal events (No training corresponding to anomalous events was used). 117 normal events and
Table 6.1. Confusion matrices of proposed and state-of-the art video anomaly detection methods on PDTV data – unsupervised anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>Piciarelli et al.</th>
<th>Simon et al.</th>
<th>LRSP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
<td>Anomaly</td>
<td>Normal</td>
</tr>
<tr>
<td>Normal</td>
<td>70.9%</td>
<td>41.7%</td>
<td>72.6%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>29.1%</td>
<td>58.3%</td>
<td>27.4%</td>
</tr>
</tbody>
</table>

24 anomalous events are used as independent test data. We use trajectory and STV features together in LRSP and compare against Piciarelli et al and Simon et al. The confusion matrices of three methods are shown in Table 6.1. The benefits of LRSP are readily apparent. LRSP outperforms Piciarelli et al and Simon et al because of using sparsity structure and exploiting the diversity in multiple perspectives (trajectory feature and STV feature) on events.

6.4.3 Performance Variation with Regular Parameter $\lambda$

In our optimization problem in (6.3), there is parameter $\lambda$ which controls the relative importance of $\| \cdot \|_*$ and $\| \cdot \|_1$ terms. In Fig. 6.3, we plot the detection rate curves against the value of $\lambda$ for PDTV data. Fig. 6.3 reveals that $\lambda \in [0.25, 0.75]$ leads to good performance. Both excessively low and high values of $\lambda$ lead to a loss in performance. In particular, when $\lambda$ is large, the cost function reduces largely to the $\| \cdot \|_1$ matrix norm and the performance drop is very significant. This emphasizes the value of the low-rank term which allows greater generality over row-sparsity and can capture sparse matrix structures arising in real-world scenarios. Note the results of LRSP in Tables 6.1 - 6.2 are reported using the “best” $\lambda$. 
Figure 6.3. Detection rates curves with respect to the value of $\lambda$

Table 6.2. Confusion matrices and running time of LRSP and JSM on PDTV data – unsupervised anomaly detection.

<table>
<thead>
<tr>
<th></th>
<th>LRSP</th>
<th>JSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run time</td>
<td>219 seconds</td>
<td>783 seconds</td>
</tr>
<tr>
<td>Normal</td>
<td>78.6%</td>
<td>82.9%</td>
</tr>
<tr>
<td>Anomaly</td>
<td>21.4%</td>
<td>66.7%</td>
</tr>
</tbody>
</table>

6.4.4 Computational Benefits and Trade Off

We now compare LRSP against JSM, where Since JSM detect video anomaly using well-structured dictionary with class information. For the PDTV data, Table 6.2 shows the confusion matrices and run time (total run time of running all the 141 test trajectories) of LRSP and JSM. We can see that the proposed LRSP method runs much faster than the JSM with a small loss in detection rates. This is expected because JSM has the benefit of pre-labeled event classes. In structured scenarios, the performance of JSM in fact serves as the practical upper bound for LRSP.
6.5 Conclusion

In this chapter, we propose low rank sparsity prior for video anomaly detection. We find that enforcing a low-rank structure can ease the rigidity of traditional row-sparse constraints on sparse coefficient vectors/matrices. A significant practical benefit with the proposed method is that it is not necessary to assign class labels to the normal trajectories, and therefore the manual effort in building the training dictionary is much reduced. All the normal trajectories are collected together as a big dictionary, and there is no need to group training trajectories (or events) into different classes as is done in [21], [22], [32]. Because low-rank matrices are of course not always sparse, an additional $l_1$ regularization term is added. Further, if rank is substituted by its convex nuclear norm alternative, then significant computational benefits can be obtained over existing methods in sparsity based video anomaly detection. Experiments on benchmark video datasets reveal that our method is competitive with the state-of-the art while providing robustness benefits under occlusion.
Conclusion

7.1 Summary of Contributions

This dissertation developed several adaptive sparsity models for the problem of video anomaly detection. Here, I briefly summarize the contributions of this dissertation:

Capturing interaction between multiple objects: In Chapter 3, we focused on multi-object anomaly detection and extended the approaches in [21,22] towards a joint sparsity model where a matrix (instead of a vector) of sparse coefficients results. This extension is highly non-trivial because the structure of this matrix of sparse coefficients is not naturally row sparse. The model is meaningful in the multi-object scenario only when there is object-wise correspondence in the linear combinations. To incorporate this very challenging constraint, we therefore developed and solved a new joint sparsity problem with the help of a new greedy pursuit algorithm. In addition, a suitable outlier rejection measure is developed for the multiple-object case that obviates the need to build anomalous event classes, and enables unsupervised anomaly detection with high accuracy.
**Practical improvement:** In Chapter 4, we proposed two practical methods to improve the performance of existing sparsity based video anomaly detection techniques. First, in order to make the trajectory more linearly separable, we introduced non-linearity into the linear sparsity model. The kernel function project the low dimension non-separable trajectories into a high dimension nonlinear feature space in which the trajectories becomes more separable which leads to a better performance on classification of events. Second, we proposed a discriminative dictionary learning technique that can effectively reduce the size of training dictionaries that enable sparsity based classification/anomaly detection without adversely influencing detection performance.

**Exploiting diversity of event encodings:** In structured scenario, the object behaviors in video scene are relatively simple, thus object specific and class specific event training dictionaries are available. With well-structured training dictionaries, in Chapter 5, we focused on multi-perspective anomaly detection and extend the existing approaches towards a simultaneous sparsity model where multiple event encodings has been employed. Given multiple training event representations in a dictionary, the identity of the test event is now encoded by the sparse structure of the coefficient matrix. It is this structure that we first quantify and then exploit for classifying events. First, for single-object anomaly detection, we introduced active curves to generalize the sparsity model from training level to class level. Second, for multi-object anomaly detection, we further extended our sparsity model to a block-diagonal structure which can handle the interaction between multiple objects. Third, we also showed that in the absence of training from anomalous events, i.e. unsupervised anomaly detection, outlier rejection measures on the sparse coefficient matrix can be used with relatively little drop in performance.
**Video anomaly detection in unstructured scenario:** In unstructured scenario, where class specific and object specific training dictionary is not available, we cannot use the simultaneous sparsity model to detect anomalies. Therefore, in Chapter 6, we proposed low rank sparsity prior for unstructured video anomaly detection. We found that enforcing a low-rank structure can ease the rigidity of traditional row-sparse constraints on sparse coefficient vectors/matrices. A significant practical benefit with the low rank sparsity prior is that it is not necessary to assign class labels to the normal trajectories, and therefore the manual effort in building the training dictionary is much reduced. All the normal events are collected together as a big dictionary, and there is no need to group training events into different classes as is done in [21], [22], [24]. Because low-rank matrices are of course not always sparse, an additional $l_1$ regularization term is added. Further, if rank is substituted by its convex nuclear norm alternative, then significant computational benefits can be obtained over existing methods in sparsity based video anomaly detection.

### 7.2 Future Research

- **Multi-camera view video anomaly detection:** In Chapter 5, we proposed a simultaneous sparsity model for multiple event encoding video anomaly detection. In real world scenario, multi-camera system has been widely used in video surveillance application. If we treat the information obtained by each camera as one perspective of the event. It is nature to extend our simultaneous sparsity model for the problem of multi-camera view anomaly detection.
• **Practical methods for simultaneous sparsity model and low rank sparsity prior:** Chapter 4 proposed two practical methods for our joint sparsity model. kernalization method and discriminative dictionary learning may be a useful extension of simultaneous sparsity model and low rank sparsity prior and can be pursued in future research.

• **More computationally efficient video anomaly detection technique:** In this dissertation, we proposed several sparsity models for video anomaly detection. It is true that some sparsity models has cheaper computational complexity (low rank sparsity prior) than the state of the art methods. However, it still takes several seconds or minutes to run the algorithm. One purpose of video anomaly detection is to identify anomalies from vast amounts of video. Therefore, one possible future research can be reducing the computational complexity of existing approaches.

• **Adaptive event encoding:** Event encoding (feature extraction) is always an active area of ongoing research and here we exploited diversity among different event encodings using simultaneous sparsity model. The features extracted by existing video anomaly detection methods like trajectory or STVs can only reflect some aspects of the events. In the research field of video deep learning, people are trying to adaptively extract features based on the contents of the video and the demand of video analysis. We believe a deeper investigation towards adaptive event encoding is a worthwhile research pursuit.

• **Design of dictionary:** Building dictionary is significant for any classification problem. A good dictionary can provide better performance. In video anomaly detection area, there is no mature dictionary design technique. Es-
pecially for surveillance video, since the length of video is usually very long, people simply extract a small set of video chunk as dictionary. An interesting research direction would be the investigation of better dictionary design techniques that can best represent the events from video.
Bibliography


Vita

Xuan Mo

Xuan Mo received his B.S. degree in Automation Science and Electrical Engineering from Beihang University, Beijing China in 2007 and his M.S. degree in Automation from Tsinghua University, Beijing, China in 2010. He joined the Ph.D. program at the Department of Electrical Engineering, Penn State University in 2010. Since then, he has been a member of the Information Processing and Algorithms Laboratory (iPAL) working with the Professor Vishal Monga as a graduate research assistant. During the summers of 2011 and 2012, he was a summer intern at Xerox Research Center in Webster, NY, where he worked on video anomaly detection.

Journal Publications:

