ANALYSIS, DETECTION AND CLASSIFICATION OF SIGNALS USING SCALAR AND VECTOR SPARSE MATRIX TRANSFORMS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Leonardo R. Bachega

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2013

Purdue University

West Lafayette, Indiana

To my parents.

ACKNOWLEDGMENTS

I would like to thank my advisor prof. Charles A. Bouman, for guiding me during the years of research that led to this document. Dr. Bouman pushed me out of my comfort zone and helped me realize some of my true potential. He taught me how to strive for excellence and to never accept “anything less than perfection.”

All members of my committee had significant impact on this work. Prof. Sam Midkiff funded me during the early years of graduate school and encouraged me to find my passion. Prof. Mimi Boutin has always been supportive and available to help with various issues. She taught the course on Statistical Pattern Recognition that introduced me to this field of research. Dr. James Theiler has co-authored the papers that form a large part of the material here presented and, despite of being physically distant, has been always available for valuable discussions.

Several faculty members of the CNSIP area at Purdue constributed with ideas, suggestions, and general advice over the past few years. It is worth mentioning professors M. Bell, T. Talavage, and J. Allebach. Prof. N. Shroff, from Ohio State University, co-authored one of my papers and provided me with many insights on wireless sensor networks.

I would like to thank all my lab and office mates for their support and friendship: Guangzhi Cao, with whom I co-authored papers; Singanallur Venkatakrishnan (Venkat) for numerous discussions, and for proof reading a large portion of this material; Eri Haneda and Jordan Kisner, for countless conversations and advices; and all my other office mates: Zhou, Jianing, Rui, Aditya, Zeeshan, Dilshan, Haitao, Pengchong, Ruoqiao, Yandong and Burak.

This dissertation would never have been written without the support of Dongchun Zhu, my lovely girfriend, who has been beside me during the whole process, and shared with me the good and the bad moments, the moments of struggle and the moments of victory, and whose kind words of encouragement kept me going until the end. Dongdong has taught

me so much about “Chinese discipline,” an art that I have just started to grasp, and without which I would have never had finished this document. Thank you Dongdong, I love you very much!

I am deeply grateful for having a wonderful family. My parents, Sônia Maria and José Alberto Bachega have given me unconditional support, always, throught my entire life. My siblings Fernando and Luciana. My uncles José Roberto and Carlos Antônio Ruggiero, who provided me guidance on several career issues. Many other relatives and friends who have been in my thoughts during all these years: Landa, Marta, Raul (*um classico!*), Francisco, Eliete, Dú, Mário, Denise, Célia, Fernandes, Maria Ida, Emília, João, Marta, Pedro, Bia, Paulo, Juliano, and many others.

Finally, I’d like to thank those who directly or indirectly helped shape my Purdue experience and this work. All my friends, especially, Ruben, Carlos, Marcelo, Nauman, Henry, Michelle, Karla, Gale, José Newton, Gustavo, Vibhav, Denise, Chandra, Mandoye, Dalton, Xing Fang, Ana Marcia, Ricardo, Varun, Saikat, Coco, Denise, Marcello, Di and many others. José E. Moreira, from IBM Research, played an important role in encouraging me to apply to, come to, and stay in the graduate program. All members of the ECE staff, especially Matt Golden, Bonnie Jean Misner, Sherry Leuck, Michelle Wagner, Angela Rainwater, and Joanne Lax, for their time, patience and willingness to help with a broad range of issues.

TABLE OF CONTENTS

Page

LIST OF TABLES viii

LIST OF FIGURES ix

ABSTRACT xiii

1 INTRODUCTION 1

2 FAST SIGNAL ANALYSIS AND DECOMPOSITION ON GRAPHS USING THE SPARSE MATRIX TRANSFORM 6

2.1 Introduction 6

2.2 SMT Estimation and Design 7

2.3 Fast SMT Design for Graphical Data 10

2.4 Experimental Results 12

2.5 Conclusions 14

3 HYPOTHESIS TESTING IN HIGH-DIMENSIONAL SPACE WITH THE SPARSE MATRIX TRANSFORM 16

3.1 Introduction 16

3.2 The Sparse Matrix Transform (SMT) 17

3.2.1 Design of the SMT transform 17

3.2.2 Application of the SMT transform 18

3.3 Hypothesis Testing 19

3.3.1 Matched Filter 19

3.3.2 Power Detector 20

3.3.3 Classification 21

3.3.4 Hypothesis Testing using SMT 22

3.4 Experimental Results 22

3.4.1 Face Recognition 22

Page

3.4.2 Hyperspectral Image Processing 23

3.5 Conclusions 25

4 EVALUATING AND IMPROVING LOCAL HYPERSPECTRAL ANOMALY DETECTORS 28

4.1 Introduction 28

4.2 Hyperspectral Anomaly Detection 29

4.2.1 Anomaly Detection as Binary Classification 29

4.2.2 Sliding Window-based Detection 30

4.2.3 Covariance Estimation Methods 31

4.3 The Sparse Matrix Transform (SMT) 34

4.3.1 SMT Covariance Estimation 35

4.3.2 SMT Model Order 36

4.3.3 Shrinkage SMT 37

4.4 Ellipsoid Mean Log-Volume 37

4.5 Experiments 38

4.6 Conclusions 42

5 DISTRIBUTED SIGNAL DECORRELATION AND DETECTION IN WIRELESS SENSOR NETWORKS USING THE SPARSE MATRIX TRANSFORM 43

5.1 Introduction 43

5.2 The Scalar Sparse Matrix Transform 47

5.3 Distributed Decorrelation with the Vector Sparse Matrix Transform . . . 50

5.3.1 The Vector SMT Model 50

5.3.2 The Design of the Vector SMT 53

5.3.3 The Vector SMT Design with Communication Energy Constraints 55

5.3.4 Model Order Identification 57

5.4 Anomaly Detection 59

5.4.1 Problem Formulation 59

5.4.2 Ellipsoid Volume as a Measure of Detection Accuracy 60

5.5 Experimental Results 61

Page

5.5.1 Simulation experiments using artificial model data 63

5.5.2 Simulation experiments using artificial moving sphere images . 64

5.5.3 Simulation experiments using artificial 3D sphere cloud images . 67

5.5.4 Simulation experiments using real multi-camera images 72

5.6 Conclusions 78

LIST OF REFERENCES 81

A CHANGE IN LIKELIHOOD DUE TO THE DECORRELATING TRANSFORM, T 87

B THE CORRELATION SCORE 88

VITA 90

LIST OF TABLES

Table Page

2.1 Comparison of the maximum expected cross-validation log-likelihood values for all methods studied using face images (with p=644). The SMT with $M = 8$ has the largest log-likelihood, which is 92.37 greater than the value produced by the PCA+Shrinkage method. 13

5.1 Correlation score values for all pairs of views in the courtyard dataset. The correlation score measures the correlation of camera outputs between pairs of camera views. Pairs of cameras capturing the same events simultaneously have the highest correlation scores. 73

LIST OF FIGURES

Figure Page

2.1 Diagram of the (a) SMT transform (b) SMT transform with graphical structure. 9

2.2 Execution time of both the original and the fast algorithms as the dimensionality increases, and $r = 1.0$, $M = 8$, $n = 40$ remain fixed. 12

2.3 Experimental results for eigen-faces: (a) Examples of faces in the dataset used in our experiments; (b) The first 40 eigen-faces from the SMT decomposition; (c) Eigen-faces from PCA decomposition. 14

3.1 Face recognition rates for ranks 1-60 using different classifiers, SMT, LDA, Shrinkage/Identity (S-I), and Shrinkage/SMT (S-SMT), trained with 221 individuals / 3 images per individual. 24

3.2 Area under the ROC curve for the SMT as the number of Givens rotations varies. Only a few SMT's Givens rotations are necessary to get most of the detection accuracy given by the exact generalized eigen-decomposition of the true covariance matrices. 25

3.3 SCRR for hyper-spectral image AVIRIS-FLA using several different estimators: Sample covariance, SMT, Shrinkage /Identity (S-I), Shrinkage/SMT (S-SMT), graphical-SMT (gc-SMT), and Shrinkage/graphical-SMT (S-gc-SMT). Average of 10 trials (each with different signal t and different set of n samples). 26

4.1 Square sliding window used in the RX detection algorithm. The pixels in the outer window are used to compute the covariance estimate $\hat{R}$ of the background surrounding the pixel x. The pixels within the inner window (referred as the *guard window*) are not used in the covariance computation to avoid that potential anomalous pixels contaminate the estimate $\hat{R}$. 32

4.2 RGB rendering of the 800×280 pixel *Blindrad* hyperspectral dataset, captured using a HyMap sensor with 126 channels. 39

4.3 Coverage plots with the expected ellipsoid log-volume *vs.* probability of false alarm for various outer window sizes. 40

4.4 Expected ellipsoid log-volume *vs.* the dimension of the sliding window fixed probabilities of false alarm in both the original, (a) and (b), and the quasilocal, (c) and (d), coordinate systems. 41

Figure Page

5.1 A camera network where each camera captures an image of the environment from one viewpoint and encodes the image into a vector output. The aggregated outputs from all cameras form the high-dimensional vector, x. Cameras i and j have overlapping views. Because outputs from cameras with overlapping views tend to be correlated, so does the aggregated vector x. 48

5.2 (a) In the product $\tilde{\mathrm{x}} = T_m^t \mathrm{x}$, the transform T_m operates over the p-dimensional vector x, changing only the components associated with the sub-vectors $\mathrm{x}^{(i_m)}$, $\mathrm{x}^{(j_m)}$ (shaded). (b) scalar SMT decorrelation, $\tilde{\mathrm{x}} = E^t \mathrm{x}$. Each E_k plays the role of a decorrelating “butterfly”, operating on a single pair of coordinates. (c) 8-point FFT, seen as a particular case of the scalar SMT where the butterflies are constrained in their ordering and rotation angles. (d) Vector SMT decorrelation, $\tilde{\mathrm{x}} = T^t \mathrm{x}$, with each T_m decorrelating a sub-vector pair of x instead of a single coordinate pair. T_m is an instance of the scalar SMT with decorrelating butterflies operating only on coordinates of a single pair of sub-vectors. . . 52

5.3 (a) Mapping from the $2h \times 2h$ matrix E, result of the optimization in (5.16), to the $p \times p$ block sparse matrix T_m associated with the (i_m, j_m) sub-vector pair. (b) The vector SMT design algorithm. 56

5.4 The experimental setup: (a) Summary of the several approaches to sensor output decorrelation compared and their main properties. (b) Steps for decorrelation and anomaly detection used in our experimental results. Each sensor encodes its output as an h-dimensional vector using PCA. Experiments with artificial data replace the sensor vector outputs with artificially generated random vector data. The outputs are processed in the network before a detection decision is made. 62

5.5 Generation of a data sample, x aggregated from correlated h-dimensional sensor outputs $\mathrm{x}^{(i)}$, $i = 1, \cdots, L$. (a) First we draw each $\mathrm{x}^{(i)}$ independently from the $\mathcal{N}(0, R)$ distribution, with $[R]_{rs} = \rho^{|r-s|}$. Then, we permute individual coordinates of x across all $x^{(i)}$, $i = 1, \cdots, L$ to spread correlations among all sensor outputs. (b) Each $x^{(i)}$ is the output of a sensor i connected to other sensors in a hierarchical network with binary tree topology. 63

5.6 Vector SMT model accuracy *vs.* communication energy consumption using 100 training data samples from AR(1) model. Comparison of different vector SMT pairwise transforms for a range of communication energies using 100 training data samples from AR(1) model: (a) average log-likelihood over 300 test samples; (b) ellipsoid log-volume covering 99% of the test samples (1% false alarm rate). The choice of scalar SMT MDL produces the best increase in accuracy, measured by both metrics. 65

Figure Page

5.7 Comparison of vector SMT energy constraint parameter values for a range of communication energies using 100 training data samples from AR(1) model: (a) average log-likelihood over 300 test samples; (b) ellipsoid log-volume covering 99% of the test samples (1% false alarm rate). Vector SMT models with larger μ are the most accurate for fixed small energy values. For large energy values, the constrained models tend to exhibit sub-optimal accuracies compared to the unconstrained vector SMT. 66

5.8 Simulated 3D space with bouncing sphere: the sphere takes random positions along the line indicated by the double arrow (a) typical behavior; (b) anomalous behavior. The camera views: (c) top (X-Y dimensions); (d) side (X-Z dimensions). The detection accuracies using independent processing and vector SMT joint processing: (e) ROC curve; (f) “coverage plot” with log-volume of ellipsoid *vs.* probability of false alarm. Because there are only two camera views, centralized and vector SMT processing methods are equivalent. . . . 68

5.9 Eigen-image pairs of the moving sphere experiment sorted according to their corresponding eigenvalues in decreasing order (left-to-right): (a) when the camera views are processed independently, each eigenvector models a single view; (b) when the camera views are processed jointly using the vector SMT, each eigenvector models both views together. 69

5.10 The twelve camera views of a 3D sphere cloud sample: (a) a typical sample (hollow cloud); (b) an anomalous sample (dense cloud). The images suggest the difficulty to discriminate anomalous from typical samples by looking at each view independently. Instead, the discriminant information is contained in the joint camera views. 71

5.11 Anomaly detection accuracy using the sphere cloud data: (a) ROC analysis; (b) log-volume of ellipsoid *vs.* probability of false alarm. Vector SMT decorrelation yields to the most accurate detection results for all false alarm rates. (c) log-volume of ellipsoid for 1% false alarm rate, i.e., 99% coverage *vs.* communication energy. 71

5.12 The courtyard dataset from the UCR Videoweb Activities Dataset [78]: eight cameras, with ids 1 to 8 from left to right, monitor a courtyard from different viewpoints. Several activities in the courtyard are captured simultaneously by several cameras. 72

5.13 Eigen-images from camera views of the courtyard dataset. (a) independent processing of camera views: each eigen-image corresponds to a single view and does not contain correlation information among multiple views; (b) joint processing modeled by the vector SMT: each eigen-image contains joint information of all correlated views. 74

Figure Page

5.14 Detection accuracy measured by the ellipsoid log-volume for the courtyard data set. Coverage plots showing the log-volume *vs.* probability of false alarm: (a) model order, $M = 7$, matching the energy of centralized processing, (b) model order, $M = 4$, matching 50% of the energy consumed for the centralized processing; (c) log-volume *vs.* communication energy for fixed probability of false alarm, $P_{FA} = 0.008$. When the communication energy is equal to the level required to execute the scalar SMT at a centralized node, the vector SMT has better detection accuracy. When the energy level is 50% of the level required by the centralized approach, the vector SMT has similar accuracy. 75

5.15 Detection accuracy of anomalies artificially generated by a 4-fold increase of the largest eigenvalue of a single view: (a) view 2; (b) view 6; (c) view 8. The detection accuracy increases with decorrelation when the anomalies are in camera views that are highly correlated with other views. When the anomaly is inserted in a uncorrelated view, decorrelation methods do not increase the detection accuracy. 76

5.16 ROC analysis of the Ocean's Eleven anomaly, generated by swapping images of a single camera view between samples: (a) camera view 2, and (b) camera view 6, which are highly correlated with other views; (c) camera view 8, which is uncorrelated with other views. 77

5.17 Samples used in the experiment detecting people coalescing in the middle of the courtyard: (a) Typical samples; (b) Anomalous samples, with images of people coalescing. 78

5.18 ROC analysis comparing the detection accuracy when detecting people coalescing in the middle of the courtyard. Detection using vector and scalar SMTs are highly accurate for small probabilities of false alarm. In this experiment the vector SMT consumes approximately 60% of the communication energy required for the scalar SMT. 79

ABSTRACT

Bachega, Leonardo R. Ph.D., Purdue University, August 2013. Analysis, Detection and Classification of Signals Using Scalar and Vector Sparse Matrix Transforms. Major Professor: Charles A. Bouman.

Several pattern recognition problems require accurate modeling of signals with high dimensionality, p, often from a limited number of samples, n. We present high-dimensional signal analysis techniques based on the Sparse Matrix Transform (SMT). The recently proposed SMT successfully models high-dimensional signals in various application domains when n is small, including the case with $n < p$. The resulting decorrelating transform is sparse, full rank, and inexpensive to apply, typically requiring only $O(p)$ computation.

Our main contribution is the vector SMT, a novel method for sparse matrix transform computation in distributed environments such as in wireless sensor networks (WSNs). We envision a scenario where each sensor generates a vector output. Together, all sensor outputs form a p-dimensional aggregated vector, x. The vector SMT algorithm then performs distributed decorrelation of x by applying pair-wise transforms to pairs of sensor outputs (i.e., subvectors of x) until x is fully decorrelated. Simulations with multi-view camera networks show that the vector SMT effectively decorrelates the multiple camera views with low total communication between sensors. Because our method enables joint processing of multiple views, we observe significant improvements to anomaly detection accuracy in artificial and real data sets compared to when the views are processed independently.

Another important contribution is the graphical-SMT algorithm, a new, fast design method for sparse matrix transforms, suited for signals with underlying graphical structure such as images and networks. Finally, we develop an SMT-based, sparse framework for hypotheses testing and apply it to classification and anomaly detection using human faces and hyperspectral image data sets.

1. INTRODUCTION

Modern age is marked by the constant and ubiquitous production of data. The advent of computer systems and inexpensive capturing devices such as digital cameras, smart phones and scanners, just to mention a few, made it possible for virtually anyone to publish data recorded from a wide variety of human activities. In addition to traditional datasets such as data from conducted polls, scientific experiments, medical and clinical trials, now we have data available from a myriad of new sources unimaginable a couple of years ago. New images posted by Facebook users, YouTube videos together with textual data from web sites such as Wikipedia, are a few examples of these new sources of data.

In an age when creating data is so common, making sense of it is paramount. The science of “learning from data” has played key role in many fields. Traditional examples can be found in medicine (identification of risk factors for several types of cancer, or the prediction of heart attacks from patient data), military (identification of hidden targets from hyperspectral data), surveillance (face detection and recognition), just to mention a few. Recently, increasing attention has been given to the analysis of data collected in wireless sensor and camera networks [1–3]. As sensing technologies become widely available, many monitoring applications deploying a large number of sensors have emerged [1, 2, 4]. In these networks, a large number of sensors produce copious amounts of data. Because these sensors usually operate under contrained battery power and narrow communication bandwidth, this data deluge imposes serious challenges to the way these data is communicated and processed.

Numerous methods to make sense of data have emerged in the past few decades, combining techniques of statistical analysis with computer algorithms. Together these methods have produced the entire fields of *Machine Learning* and *Pattern Recognition* [5–8]. In the core of these fields are methods for spectral analysis, detection, and classification of data.

Learning the covariance structure of the data is a fundamental step for the accuracy of these methods.

Unfortunately, when the datasets are high-dimensional, i.e, the number of dimensions, p is very large, one often does not have the privilege of having enough number of training samples, n available for good estimates of the data covariance structure. More, specifically, in order to form a good unbiased estimate of the covariance, one needs n, to be substantially larger than p. As a result, classical methods often perform poorly in this $n \ll p$ scenario (if they work at all). This *curse of dimensionality* presents a real challenge since, as argued in [9], the $n \ll p$ scenario is rather the common one in most applications.

Various methods have been proposed to get around this course of dimensionality when it comes to covariance estimation and provide full rank covariance estimates in scenarios when $n \ll p$. These methods impose some sort of regularizing constraint to a full-rank covariance estimate, usually at the expense of introducing some (hopefully small) bias to the estimate.

Shrinkage methods [10–12] are widely used, and work by estimating the covariance matrix as a combination of the rank-deficient sample covariance and a positive definite target such as the identity matrix or the diagonal of the sample covariance. The intuition behind these methods is that a combination of an estimator that over-fits the data (i.e, the rank-deficient sample covariance) with an estimator that under-fits the data will produce a more accurate final estimate. The right combination is normally determined using cross-validation. Shrinkage covariance estimation has been used to model high-dimensional signal such as hyperspectral images [13], and stock price data [14]. In [15], shrinkage is used for covariance estimation for a matched filter in the hyperspectral image domain. In [16] a regularized linear discriminant analysis (LDA) method relying on shrinkage estimation is proposed and has been applied to datasets from several domains.

Lasso-based methods [17] constrain the covariance or its inverse to be sparse by imposing a L1-norm constraint over the columns of the matrix. Other methods, known as Sparse Principal Component Analysis (S-PCA) impose the L1-norm constraint to the eigenvectors of the covariance matrix [18, 19]. A theoretical justification for the lasso-based methods is provided in [20].

Finally, banding and thresholding have also been used to obtain sparse estimates of large covariance matrices [21, 22].

The Sparse Matrix Transform (SMT) [23, 24] estimates the eigen-decomposition of a high-dimensional signal by assuming that the eigen-transformation can be represented as a sparse matrix transform, i.e., as a SMT, and then maximizing the Gaussian likelihood over the sample set of size n under this SMT constraint. The SMT is formed by a finite product of Givens rotations [25], so it decomposes the eigen-decomposition into a product of very sparse transformations.

The SMT eigen-decomposition assumption has two major advantages. First, the approach can improve the accuracy of the estimated transform for a fixed quantity of data [26]. Second, the eigen-decomposition then has the form of an SMT, which is very fast to apply, i.e. is $O(p)$. These characteristics of the SMT make it very attractive to be deployed to the covariance learning for detection and classification of high-dimensional data with the potential of yielding to faster and more accurate methods.

A naïve implementation of the SMT design apporach requires $O(p^3)$ computation to design the sparse transform from the observed training data. While applying the SMT is always fast, designing it can therefore be burdensome when p is very large.

Our first constribution is an algorithm for the SMT design called *Graphical-SMT* that takes advantage of underlying graphical structure of the data and designs the eigen-transform with average empirical complexity $O(p \log p)$, as opposed to the $O(p^3)$ complexity observed in the original SMT design method proposed in [24], and show how this algorithm performs the eigen-analysis of high-dimensional data used to estimate the data's high-dimensional covariance.

Second, we develop SMT-based methods for classification and detection of high-dimensional random signal. These methods rely on the SMT for the estimation of the high-dimensional covariance structure of the data. Because of the nice properties of the SMT mentioned above, these resulting classification and detection methods can operate directly over high-dimensional data at a low computational cost while being more accurate than competing methods.

Finally, our most important contribution is the **vector SMT**, a novel method for sparse matrix transform computation in distributed environments such as in wireless sensor networks (WSNs). This vector SMT generalizes the concept of sparse matrix transform to decorrelation of sequences of vector pairs instead of coordinate pairs. We envision a scenario where each sensor generates a vector output. Together, all sensor outputs form a p-dimensional aggregated vector, x. The vector SMT algorithm then performs distributed decorrelation of x by applying pair-wise transforms to pairs of sensor outputs (i.e., subvectors of x) until x is fully decorrelated. Being able to select the most correlated pair of vectors at a given time is central to our method. We introduce the concept of a *correlation score*, a generalization of the correlation coefficient between two random variables to pairs of random vectors. This correlation score is closely related to the mutual information between a pair of random vectors [27], and the concept of total correlation [28]. We show that for a pair of scalar random variables, this correlation score is equal to the absolute value of the correlation coefficient. We also propose a principled way of incorporating communication energy constraints when selecting a pair of correlated vector outputs in a WSN based on Lagrange multipliers. Simulations with multi-view camera networks show that the vector SMT effectively decorrelates the multiple camera views with low total communication between sensors. Because our method enables joint processing of multiple views, we observe significant improvements to anomaly detection accuracy in artificial and real data sets compared to when the views are processed independently.

The rest of this document is organized as follows:

- In Chapter 2, we describe the Sparse Matrix Transform (SMT) design together with a new algorithm, here referred as the *graphical-SMT* , that explores the underlying graphical structure of the dataset and designs the sparse transform with average complexity $O(p \log p)$ [24]. We also show how to use the SMT to perform the eigenanalysis of the data. In particular, we show results with a face dataset and how the SMT is used to produce the so-called *Eigenfaces*;

- In Chapter 3, we show how the SMT enables two tasks central to problems in machine learning and pattern recognition: detection and classification of high-dimensional signals [29,30]. We show results suggesting that SMT-based methods perform well even in a scenario when the number of training samples is much smaller than the number of dimensions of the dataset, i.e, $n \ll p$. Several results using hyperspectral and facial image datasets are presented;

- In Chapter 4, we address two issues related to anomaly detection in hyperspectral images. Our first contribution is to formulate and employ a mean-log-volume approach for evaluating local anomaly detectors. Our mean-log-volume approach allows for an effective evaluation of a detector's accuracy without requiring labeled testing data or an overly-specific definition of an anomaly. The second contribution is to investigate the use of the Sparse Matrix Transform (SMT) to model the local covariance structure of hyperspectral images, used in local, sliding window methods. Our results suggest that RX-style detectors using the SMT covariance estimates perform favorably compared to other methods even (indeed, especially) in the regime of very small window sizes.

- In Chapter 5, we propose the *vector SMT*, a new decorrelating transform suitable for performing distributed anomaly detection in wireless sensor networks (WSN). Here, we assume that each sensor in the network performs vector measurements, instead of a scalar ones. The vector SMT decorrelates a sequence of pairs of vector sensor measurements, until the vectors from all sensors are completely decorrelated. We perform simulations with a network of cameras, where each camera records an image of the monitored environment from its particular viewpoint. Results show that the proposed transform effectively decorrelates image measurements from the multiple cameras in the network while maintaining the communication cost low. Because it enables joint processing of the multiple images, our method provides significant improvements to anomaly detection accuracy when compared to the baseline case where we process the camera views independently.

2. FAST SIGNAL ANALYSIS AND DECOMPOSITION ON GRAPHS USING THE SPARSE MATRIX TRANSFORM

2.1 Introduction

Decorrelation and analysis of high dimensional signals are of great importance in a wide variety of applications [9]. For example, whitening filters and block transforms such as the DCT are widely used to approximately decorrelate stationary signals in time and space for applications such as image and audio source coding. If the signal being processed is not stationary, then techniques such as eigen-image and eigen-signal analysis can be used to decompose high-dimensional signals into approximately decorrelated components for applications such as anomaly detection [31] or face recognition [32]. However, one disadvantage of eigen-signal analysis or equivalently Karhunen-Loève decomposition methods is that they require a knowledge of the high dimensional signal's covariance. Since a signal of dimension p has an associated covariance matrix of dimension p^2, the amount of data required to estimate this covariance tends to grow as p grows. More specifically, in order to form a good unbiased estimate of the covariance, one needs the number of observed vectors, n, to be substantially larger then their dimension, p [9]. In practice, one often does not have this luxury.

Recently, methods have been proposed which allow high dimensional eigen-signal analysis even when the number of observations is much less than the dimension of the signal. These approaches estimate the eigen-decomponsition (and associated covariance) by imposing some type of regularizing constraint [11, 17, 18, 33].

In particular, the method of [23] estimates the eigen-decomposition of a high dimensional signal by assuming that the eigen-transformation can be represented as a sparse matrix transform (SMT). The SMT is formed by a finite product of Givens rotations, so it decomposes the eigen-decomposition into a product of very sparse transformations.

The SMT eigen-decomposition assumption has two major advantages. First, the approach can improve the accuracy of the estimated transform for a fixed quantity of data [26]. Second, the eigen-decomposition then has the form of an SMT, which is very fast to apply, i.e. is $O(p)$. However, one limitation of this approach is that it requires $O(p^3)$ to design the SMT from the observed training data. While applying the SMT is always fast, designing it can therefore be burdensome when p is very large.

In this chapter, we present an SMT eigen-decomposition method suited for application to signals that live on graphs, i.e. signals y_i where $i \in \{1, \cdots, p\}$ indexes nodes in a graph. This SMT eigen-decomposition method has two major advantages over the more generic method presented in [23]. First, the resulting SMT can be more accurately estimated due to the graphical constraint. Second, the computation required to design the SMT from training data is dramatically reduced from an average observed complexity of p^3 to $p \log p$.

In practice, many forms of data have a natural graphical structure which can be exploited to make the SMT design fast. For example, in images we can assume that neighboring pixels are connected in a graph structure to make SMT design practical for large images. We show that the resulting sparse transform can be used for eigen-image analysis in applications such as face recognition, and demonstrate that it results in a more accurate fit (as measured via the cross-validated log-likelihood) for real face image data.

2.2 SMT Estimation and Design

The SMT design consists of estimating the full set of eigenvectors and associated eigenvalues for a general p-dimensional signal. More specifically, the objective is to estimate the orthonormal matrix E and diagonal matrix Λ such that the signal covariance can be decomposed as $R = E\Lambda E^t$, and to compute this estimate from n independent training vectors, $\mathrm{Y} = [\mathrm{y}_1, \cdots, \mathrm{y}_n]$. This is done by assuming the samples are i.i.d. Gaussian random vectors and computing the constrained maximum log-likelihood (ML) estimates of E and Λ. In [23], we show that these constrained ML estimates are given by

$$\begin{aligned} \hat{E} &= \arg\min_{E\in\Omega_K} \left\{ \left| \operatorname{diag}(E^t S E) \right| \right\} && (2.1) \\ \hat{\Lambda} &= \operatorname{diag}(\hat{E}^t S \hat{E}) \ , && (2.2) \end{aligned}$$

where $S = \frac{1}{n}\mathrm{YY}^t$ is the sample covariance matrix, and Ω_K is the set of allowed orthonormal transforms.

If $n > p$ and Ω_K is the set of all orthonormal transforms, then the solution to (2.1) and (2.2) is the diagonalization of the sample covariance, i.e. $\hat{E}\hat{\Lambda}\hat{E}^t = S$. However, the sample covariance is a poor estimate of the covariance when $n < p$.

In order to improve the accuracy of the covariance estimate, we will impose the constraint that Ω_K be the set of sparse matrix transforms (SMT) of order K. More specifically, we will assume that the eigen-transformation has the form

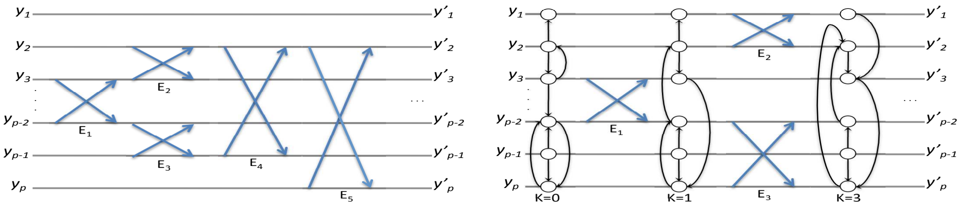
$$E = \prod_{k=1}^{K} E_k = E_1 \cdots E_K \ , \tag{2.3}$$

where each E_k is an orthonormal transform known as a Givens rotation, and K is the model order parameter. Each Givens rotation operates on just two coordinates, (i_k, j_k), so

$$E_k = I + \Theta(i_k, j_k, \theta_k) \text{ , where}$$

$$[\Theta]_{ij} = \begin{cases} \cos(\theta_k) - 1 & \text{if } i = j = i_k \text{ or } i = j = j_k \\ \sin(\theta_k) & \text{if } i = i_k \text{ and } j = j_k \\ -\sin(\theta_k) & \text{if } i = j_k \text{ and } j = i_k \\ 0 & \text{otherwise} \end{cases} \quad . \tag{2.4}$$

Figure 2.1(a) illustrates the structure of the SMT. Intuitively, each Givens rotation, E_k, plays the same role as the butterflies of a fast Fourier transform (FFT). In fact, the SMT is a generalization of both the FFT and the orthonormal wavelet transform. However, since both the ordering of the coordinate pairs, (i_k, j_k), and the values of the rotation angles, θ_k, are unconstrained, the SMT can model a much wider range of transformations. It is often useful to express the order of the SMT as $K = rp$, where r is the average number of



(a) (b)

Fig. 2.1. Diagram of the (a) SMT transform (b) SMT transform with graphical structure.

rotations per coordinate. Typically r is small (< 5), so that the computation to apply the SMT to a vector of data is very low, i.e. $2r + 1$ floating-point operations per coordinate.

The optimization of (2.1) is non-convex, so we use a greedy optimization approach in which we select each rotation, E_k, in sequence to minimize the cost. The greedy SMT design approach leads to the following very intuitive algorithm.

Initialize $S \leftarrow \frac{1}{n}\mathrm{Y}\mathrm{Y}^t$ and $\hat{E} \leftarrow I$.

For $k = 1$ to K,

1. Search for the two most correlated coordinates,
$$(i_k, j_k) \leftarrow \arg\max_{(i,j)} \left(\frac{S_{ij}^2}{S_{ii}S_{jj}} \right) \ . \tag{2.5}$$
2. Compute the Givens rotation, E_k, that decorrelates the coordinates with the rotation angle
$$\theta_k = \frac{1}{2}\tan^{-1}\left(-2S_{i_kj_k}, S_{i_ki_k} - S_{j_kj_k}\right) \ . \tag{2.6}$$
3. Perform the updates
$$S \quad \leftarrow \quad E_k^t S E_k \tag{2.7}$$
$$\hat{E} \quad \leftarrow \quad \hat{E} \cdot E_k \tag{2.8}$$

The computation of the SMT design algorithm is dominated by the time required to search for the most correlated coordinate pairs in (2.5). This search requires that the correlation between all $p(p-1)/2$ combinations of coordinates be checked. Assuming that

r and n are constant, this means the the SMT design algorithm requires $O(p^3)$ operations, and can be computationally expensive when p is large.

2.3 Fast SMT Design for Graphical Data

The general SMT design algorithm of Section 2.2 does not constrain the search for the coordinates (i_k, j_k). However, in many applications, such as social networks and images, the signal data have a natural graphical structure in which neighboring coordinates (i.e, those connected by an edge) are the ones expected to be correlated. We constrain the search for the pair (i_k, j_k) described above to these neighboring coordinates. This constraint to neighboring coordinates has two potential advantages: it can both reduce the computation of the SMT design and also improve the estimate of E.

Figure 2.1(b) illustrates this approach to SMT design for graphical data. For each value of k, the figure shows that the coordinates have a graph structure, with each node i pointing to a set of nodes ∂i. [1] Then Givens rotations are constrained to be between nodes (i, j) such that either $i \in \partial j$ or $j \in \partial i$. Of course, once a rotation is applied to a pair of coordinates, the neighborhood relations between nodes of the graph must be updated while maintaining the constraint that the maximum fan-out of the graph is bounded by $|\partial i| < M$, where generally $M \ll p$. [2]

Using this graphical constraint, we have the following algorithm for greedy SMT design.

For all i, initialize the list of neighbors ∂i, and for $\forall j \in \partial i$ compute the correlation C_{ij}.
For $k = 1$ to K,

1. Search for the two most correlated neighboring coordinates,

$$(i_k, j_k) \leftarrow \arg \max_{1 \leq i \leq p} \left(\max_{j \in \partial i} C_{i,j} \right) \ . \tag{2.9}$$

[1] Here we do not assume that the neighborhood relationship is symmetric, so this is a directed graph.

[2] However, the fan-in may exceed this bound, i.e. the set $\{j : i \in \partial j\}$ may be large for some nodes in the graph.

2. Compute the Givens rotation, E_k, that decorrelates the coordinates with the rotation angle
$$\theta_k = \frac{1}{2}\tan^{-1}\left(-2\sum_{l=1}^{n} \mathrm{Y}_{i_k l}\mathrm{Y}_{j_k l}, \sum_{l=1}^{n} \mathrm{Y}_{i_k l}^2 - \sum_{l=1}^{n} \mathrm{Y}_{j_k l}^2\right) . \tag{2.10}$$

3. Merge the neighborhoods ∂i and ∂j
$$\partial i_k \ \leftarrow \ \partial i_k \cup \partial j_k \tag{2.11}$$
$$\partial j_k \ \leftarrow \ \partial i_k \cup \partial j_k \ . \tag{2.12}$$

4. Prune the neighborhoods such that
$$\partial i_k \ \leftarrow \ \left\{ \begin{array}{c} M \text{ most correlated coordinates} \\ \text{between } i_k \text{ and } j \in \partial i_k \end{array} \right\} \tag{2.13}$$
$$\partial j_k \ \leftarrow \ \left\{ \begin{array}{c} M \text{ most correlated coordinates} \\ \text{between } j_k \text{ and } i \in \partial j_k \end{array} \right\} . \tag{2.14}$$

5. Perform the updates
$$\mathrm{Y} \ \leftarrow \ E_k^t \cdot \mathrm{Y} \tag{2.15}$$
$$\hat{E} \ \leftarrow \ \hat{E} \cdot E_k \ . \tag{2.16}$$

6. Update any correlations, $C_{i,j}$ that can be affected by the rotation on coordinates (i_k, j_k).

The computation of this algorithm is now dominated by the search of step 1, and the correlation update of step 6. A naïve implementation, results in an SMT design algorithm with complexity $O(p^2)$ when r, n, and M are fixed. This is because the design of each rotation, E_k, requires a search over all edges in the graph, which is $O(Mp)$.

However, a careful implementation with a red-black search tree can make the search of step 1 order $\log p$ average complexity. In this case, step 6 only requires the update of any coordinates which are either in the fan-out of the nodes $\{i_k, j_k\}$ or in the fan-in of these nodes. In practice, the sum of the fan-out and fan-in to a coordinate is, on average, of order

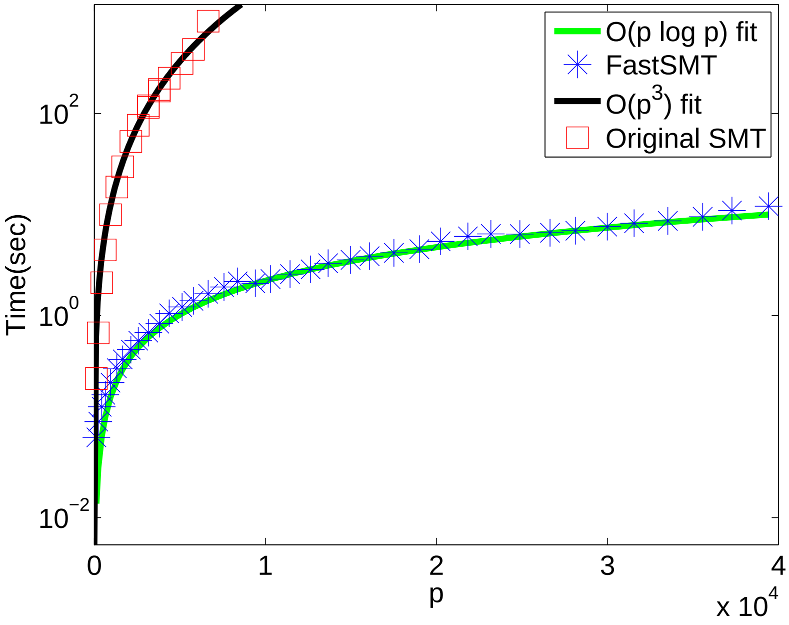


Fig. 2.2. Execution time of both the original and the fast algorithms as the dimensionality increases, and $r = 1.0$, $M = 8$, $n = 40$ remain fixed.

M. However, in a worst-case scenario the fan-in to any node of the graph is only bounded by p.

So in summary, when r, n, and M are fixed, the empirically observed complexity of the graph based SMT design is experimentally measured to be $p \log p$. However, the theoretical worse-case complexity is $O(p^2)$.

2.4 Experimental Results

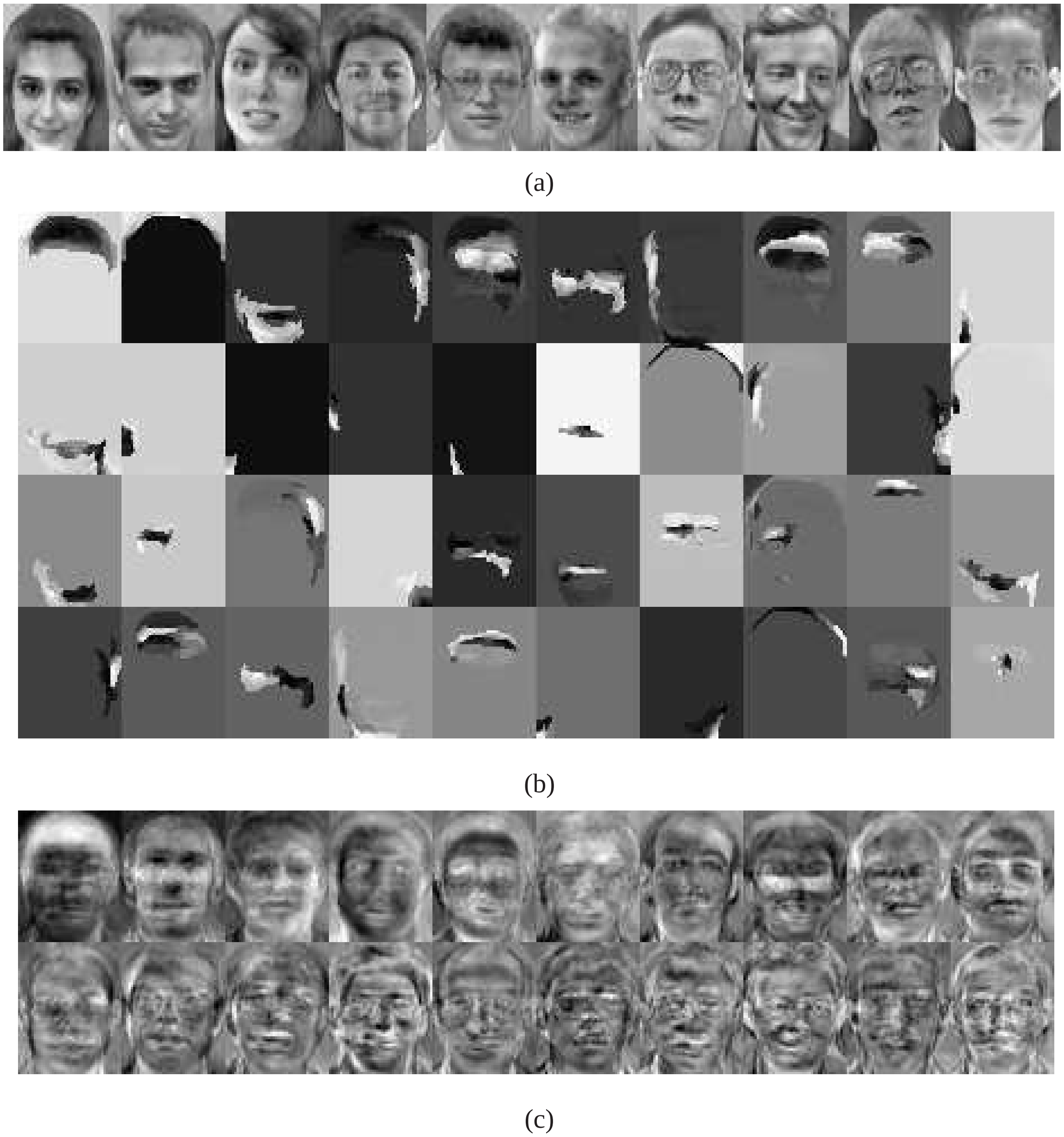
Figure 2.2 compares the computation of the fast SMT design for graphical data, with the one of the original SMT. Notice that the running time for the fast SMT design requires dramatically less computation than the original method as p becomes large, and that the two algorithms fit the proposed complexity of p^3 and $p \log p$ quite closely over a large range of p.

Table 2.1
Comparison of the maximum expected cross-validation log-likelihood values for all methods studied using face images (with p=644). The SMT with $M = 8$ has the largest log-likelihood, which is 92.37 greater than the value produced by the PCA+Shrinkage method.

method	log-likelihood	Δ	r_{max} (K_{max})
PCA+Shrinkage	-2885.71	0	-
SMT	-2805.81	79.88	1.41(910)
fast SMT($M = 8$)	-2793.33	92.37	1.57(1010)
fast SMT($M = 32$)	-2802.36	83.34	1.41(910)
diagonal	-3213.10	-327.40	-

We applied the fast SMT algorithm to a face image dataset from 40 different subjects from the *ORL Face Database* [34], with the images re-scaled to 28×23 pixels ($p = 644$). Table 2.1 shows the results of the cross-validated average log-likelihood values of the face dataset split in 10 ways(10 fold cross-validation) for both SMT algorithms and the PCA+Shrinkage method. It also shows the difference of the SMT log-likelihood values from the value obtained using PCA+Shrinkage, and the number of SMT rotations needed to obtain the stated log-likelihood value. As the results suggest, the fast SMT method produces the highest average log-likelihood value.

Figure 2.3 shows the results of the SMT eigen-decomposition applied to a small subset of 20 face images from the ORL database. Figure 2.3(a) shows examples of the images used in this experiment. Figure 2.3(b) and Figure 2.3(c) show the results of the estimated eigenvectors respectively estimated with the SMT and the PCA-based eigen-decompositions. Notice that the SMT decomposition produces a full set of eigen-faces as opposed to only 20 eigen-faces produced by PCA. Unlike PCA, the SMT eigen-images capture local spatial structure of the faces resembling their anatomical parts. We believe such a structure may yield to higher discriminative power than the PCA-based decomposition.



(a)

(b)

(c)

Fig. 2.3. Experimental results for eigen-faces: (a) Examples of faces in the dataset used in our experiments; (b) The first 40 eigen-faces from the SMT decomposition; (c) Eigen-faces from PCA decomposition.

2.5 Conclusions

We have introduced a fast algorithm for the design of SMT analysis transformations on graphical data. This approach has three major advantages: 1) It results in a more accurate estimate of the decorrelating transformation for some typical data cases, particularly when $n < p$; 2) The resulting decorrelating SMT transformation is computationally very efficient to implement, i.e. it requires only $2rp$ floating-point operations for a p dimensional vector; 3) When the observed data has a graphical structure, then the SMT design algorithm can be practically implemented with $p \log p$ computation.

3. HYPOTHESIS TESTING IN HIGH-DIMENSIONAL SPACE WITH THE SPARSE MATRIX TRANSFORM

3.1 Introduction

Statistical hypothesis testing is widely used in signal processing and machine learning. According to the seminal *Neyman-Pearson* lemma [35], when deciding between two alternative hypotheses, the test with most discrimination power depends on one's knowledge of the ratio between the likelihoods under both hypotheses, and therefore, the knowledge of the data covariance matrices under both hypotheses. In practice the true covariances are not known and we need to rely on estimates from available training sets.

However, when the data dimensionality p is large, the number of training samples, n available to estimate the covariances involved in the likelihood ratio test is small compared to p, making conventional covariance estimates to behave poorly. As argued in [9], this $n \ll p$ scenario is rather common. Nevertheless, even if one had enough samples to obtain accurate covariance matrix estimates, when p is large, the amount of computation required to compute their eigen-decomposition and the memory space required to store them would both be prohibitive, limiting the practical application of such tests.

The Sparse Matrix Transform (SMT) [23, 24] is capable of successfully modeling the covariance structure of high dimensional data in the scenario when $n \ll p$, and requiring low computational cost when applied. In this chapter we investigate the SMT deployment to estimate the covariance matrices involved in log-likelihood ratio for hypothesis testing. We look at three different flavors of hypothesis testing: matched filtering, power detection and classification.

Results in detection involving hyperspectral images and face recognition suggest that the accuracy of detectors and classifiers relying on SMT is better than of competing methods when few training samples are available, while the computation associated with its application is significantly lower. In the case when the true covariances are known, a sparse representation of the covariances by the SMT can reduce the computation required for the likelihood ratio test while yielding to similar accuracy to the exact method.

3.2 The Sparse Matrix Transform (SMT)

The essence of our method is to use SMTs to provide full-rank estimates of the $p \times p$ covariance matrices used in the detection and classification frameworks discussed in Section 3.3.

3.2.1 Design of the SMT transform

The SMT design consists of estimating the full set of eigenvectors and associated eigenvalues for a general p-dimensional signal. More specifically, the objective is to estimate the orthonormal matrix E and diagonal matrix Λ such that the signal covariance can be decomposed as $R = E\Lambda E^t$, and to compute this estimate from n independent training vectors, $\mathrm{Y} = [\mathrm{y}_1, \cdots, \mathrm{y}_n]$. This is done by assuming the samples are i.i.d. Gaussian random vectors and computing the constrained maximum log-likelihood (ML) estimates of E and Λ. In [23], we show that these constrained ML estimates are given by

$$\begin{aligned} \hat{E} &= \arg\min_{E \in \Omega_K} \left\{ \left| \mathrm{diag}(E^t S E) \right| \right\} & (3.1) \\ \hat{\Lambda} &= \mathrm{diag}(\hat{E}^t S \hat{E}) \,, & (3.2) \end{aligned}$$

where $S = \frac{1}{n}\mathrm{Y}\mathrm{Y}^t$ is the sample covariance matrix, and Ω_K is the set of allowed orthonormal transforms.

If $n > p$ and Ω_K is the set of all orthonormal transforms, then the solution to (3.1) and (3.2) is the diagonalization of the sample covariance, i.e, $\hat{E}\hat{\Lambda}\hat{E}^t = S$. However, the sample covariance is a poor estimate of the covariance when $n < p$.

In order to improve the accuracy of the covariance estimate, we will impose the constraint that Ω_K be the set of sparse matrix transforms (SMT) of order K. More specifically, we will assume that the eigen-transformation has the form

$$E = \prod_{k=1}^{K} E_k = E_1 \cdots E_K \ , \tag{3.3}$$

where each E_k is a planar rotation over some (i_k, j_k) coordinate pair by an angle θ_k, and K is the model order parameter.

Intuitively, each Givens rotation, E_k, plays the same role as the butterflies of a fast Fourier transform (FFT). In fact, the SMT is a generalization of both the FFT and the orthonormal wavelet transform. However, since both the ordering of the coordinate pairs, (i_k, j_k), and the values of the rotation angles, θ_k, are unconstrained, the SMT can model a much wider range of transformations. It is often useful to express the order of the SMT as $K = rp$, where r is the average number of rotations per coordinate, being typically very small: $r < 5$. The optimization of (3.1) is non-convex, so we use a greedy optimization approach in which we select each rotation, E_k, in sequence to minimize the cost. The greedy optimization can be done fast if a graphical constraint can be imposed to the data [24]. The parameter r can be estimated using cross-validation over the training set [23, 24] or using the minimum length description criterion proposed in [36].

3.2.2 Application of the SMT transform

Typically, r is small (< 5), so that the computation to apply the SMT to a vector of data is very low, i.e, $2r + 1$ floating-point operations per coordinate. Therefore, we can apply the SMT decorrelating transform to p-dimensional random vectors in only $(2r + 1)p$ steps.

3.3 Hypothesis Testing

Let x be a p-dimensional random vector drawn from a multivariate normal distribution. One seeks to decide between the hypotheses

$$\begin{aligned} \mathcal{H}_0 : &\quad \mathrm{x} \sim \mathcal{N}(\mu_A, R_A) \\ \mathcal{H}_1 : &\quad \mathrm{x} \sim \mathcal{N}(\mu_B, R_B) \ , \end{aligned} \tag{3.4}$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ are referred as the *null* and *alternative* hypotheses respectively. The *Neyman-Pearson* lemma [35] states that the log-likelihood ratio test

$$l(\mathrm{x}) = \log \left\{ \frac{p(\mathrm{x}; \mathcal{H}_1)}{p(\mathrm{x}; \mathcal{H}_0)} \right\} \gtrless \eta \tag{3.5}$$

maximizes the probability of detection $p(\mathcal{H}_1; \mathcal{H}_1)$ for a fixed probability of false alarm $p(\mathcal{H}_1; \mathcal{H}_0)$, which is controled by the threshold η.

Below, we discuss how the log-likelihood ratio test in (3.5) is used to test alternative hypotheses in the context of three common problems in signal processing, involving detection and classification of random signals.

3.3.1 Matched Filter

Let $\mathrm{t} \in \mathbb{R}^p$ be a deterministic signal buried in additive random clutter $\mathbf{w} \sim \mathcal{N}(0, R)$. The random vector X is measured and one wants to make a decision on whether the signal t is present (i.e, $\mathrm{x} = \mathrm{t} + \mathbf{w}$), or the measurement contains only clutter (i.e, $\mathrm{x} = \mathbf{w}$), by testing the hypotheses

$$\begin{aligned} \mathcal{H}_0 : &\quad \mathrm{x} \sim \mathcal{N}(0, R) \\ \mathcal{H}_1 : &\quad \mathrm{x} \sim \mathcal{N}(\mathrm{t}, R) \ . \end{aligned} \tag{3.6}$$

In this case, the log-likelihood ratio test in (3.5) has the form of an inner product: $l(\mathrm{x}) = \mathrm{q}^t\mathrm{x} \gtrless \eta'$, where the vector $\mathrm{q} \overset{\triangle}{=} R^{-1}\mathrm{t}$ is called a *matched filter*, and its detection capability is measured directly by the signal-to-clutter statistic [26]:

$$SCR = \frac{(\mathrm{q}^t\mathrm{t})^2}{\mathbb{E}\left\{(\mathrm{q}^t\mathrm{x})^2\right\}} = \frac{(\mathrm{q}^t\mathrm{t})^2}{\mathrm{q}^t\mathbb{E}\left\{\mathrm{X}\mathrm{X}^t\right\}\mathrm{q}} = \frac{(\mathrm{q}^t\mathrm{t})^2}{\mathrm{q}^t R \mathrm{q}} \ . \tag{3.7}$$

3.3.2 Power Detector

Let the p-dimensional random vector x be drawn from a multivariate normal distribution with the same mean under both hypotheses but different covariances. The general hypotheses in (3.4) become

$$\begin{aligned} \mathcal{H}_0 :&\quad \mathrm{x} \sim \mathcal{N}(0, R_A) \\ \mathcal{H}_1 :&\quad \mathrm{x} \sim \mathcal{N}(0, R_B) \ . \end{aligned} \tag{3.8}$$

For instance, the hypothesis test in (3.8) also corresponds to the problem of anomalous change detection in multispectral imagery modeled by Gaussian distributions [37].

We can compute the generalized eigen-decomposition [6] that diagonalizes both R_A and R_B simultaneously, allowing us to decorrelate the vector x under both hypotheses using

$$\widetilde{\mathrm{x}} = \widetilde{E}_B^t \Lambda_A^{-1/2} E_A^t \mathrm{x} \ , \tag{3.9}$$

where E_A and Λ_A are the eigenvectors and eigenvalues [1] given by

$$R_A = E_A \Lambda_A E_A^t \ ,$$

and $\widetilde{\Lambda}_B$ and $\widetilde{E}_B$ are the eigenvalues and eigenvectors of the matrix $\widetilde{R}_B$ given by

$$\widetilde{R}_B \triangleq \Lambda_A^{-1/2} E_A^t R_B E_A \Lambda_A^{-1/2} = \widetilde{E}_B \widetilde{\Lambda}_B \widetilde{E}_B^t \ .$$

The linear transformation of (3.9) is equivalent to the Fisher linear discriminant (FLD) that is used to maximize the ratio of the between class to within class scatter [6, 38, 39].

In this new space, the hypotheses in (3.8) are written in terms of $\widetilde{\mathrm{x}}$ and become

$$\begin{aligned} \mathcal{H}_0 :&\quad \widetilde{\mathrm{x}} \sim \mathcal{N}(0, I) \\ \mathcal{H}_1 :&\quad \widetilde{\mathrm{x}} \sim \mathcal{N}(0, \widetilde{\Lambda}_B) \ . \end{aligned} \tag{3.10}$$

Since x and $\widetilde{\mathrm{x}}$ are related by an invertible linear transformation, the log-likelihood ratio of (3.5) can be shown to be

$$l(\mathrm{x}) = \log \left\{ \frac{p(\mathrm{x}; \mathcal{H}_1)}{p(\mathrm{x}; \mathcal{H}_0)} \right\} \tag{3.11}$$

$$= -\sum_{i=1}^{p} \left(\frac{1}{\tilde{\lambda}_{Bi}} - 1 \right) \tilde{x}_i^2 + \sum_{i=1}^{p} \log \tilde{\lambda}_{Bi} \ , \tag{3.12}$$

[1] All eigenvalues in Λ_A are assumed here to be non-zero

where $\tilde{\lambda}_{Bi}$ is the ith diagonal element of $\widetilde{\Lambda}_B$ and $\widetilde{x}_i$ is the ith coordinate of the vector $\widetilde{\mathrm{x}}$.

3.3.3 Classification

Let y_0 and $\mathrm{y}_1, \cdots, \mathrm{y}_{\mathcal{K}}$ all be p-dimensional random vectors, and assume that each vector is formed by $\mathrm{y}_k = \mathrm{x}_k + \mathbf{w}_k$, where $\mathrm{x}_k \sim \mathcal{N}(0, R_x)$ is an unknown p-dimensional signal, and $\mathbf{w}_k \sim \mathcal{N}(0, R_w)$ is additive p-dimensional noise. Our objective is to classify the vector Y_0 as a member of the class $k \in \{1, \cdots, \mathcal{K}\}$ if the pair of vectors y_0 and y_k constitute a match, i.e, they both originated from the same signal: $\mathrm{x}_0 = \mathrm{x}_k$. Therefore, under the hypothesis of a match, the difference $\Delta\mathrm{y}_k = \mathrm{y}_k - \mathrm{y}_0 \sim \mathcal{N}(0, 2R_w)$. Alternatively, under the hypothesis that y_0 and y_k are *not* a match we have that $\Delta\mathrm{y}_k \sim \mathcal{N}(0, 2(R_x + R_w))$. In summary, the probability density of the random vector $\Delta\mathrm{y}_k$ is given by

$$\begin{array}{lll} \mathcal{H}_0: & \Delta\mathrm{y}_k \sim \mathcal{N}(0, 2(R_x + R_w)) & \text{if } \mathrm{x}_0 \neq \mathrm{x}_k \\ \mathcal{H}_1: & \Delta\mathrm{y}_k \sim \mathcal{N}(0, 2R_w) & \text{if } \mathrm{x}_0 = \mathrm{x}_k\ . \end{array} \tag{3.13}$$

The maximum likelihood selection of $\hat{k}$ is given by

$$\hat{k} = \arg\max_k \left\{ \log \left[\frac{p(\Delta\mathrm{y}_k; \mathcal{H}_1)}{p(\Delta\mathrm{y}_k; \mathcal{H}_0)} \right] \right\}. \tag{3.14}$$

Following the same lines of Section 3.3.2, we can compute the generalized eigen-decomposition of both R_x and R_w, thus allowing the computation of $\Delta\widetilde{\mathrm{y}}_k$ from $\Delta\mathrm{y}_k$, which is decorrelated under both hypotheses. As a result, the hypotheses in (3.13) are equivalent to

$$\begin{array}{lll} \mathcal{H}_0: & \Delta\widetilde{\mathrm{y}}_k \sim \mathcal{N}(0, 2(\widetilde{\Lambda}_x + I)) & \text{if } \mathrm{x}_0 \neq \mathrm{x}_k \\ \mathcal{H}_1: & \Delta\widetilde{\mathrm{y}}_k \sim \mathcal{N}(0, 2I) & \text{if } \mathrm{x}_0 = \mathrm{x}_k\ . \end{array} \tag{3.15}$$

The selection of $\hat{k}$ in (3.14) can be written in terms of the coordinates of $\Delta\widetilde{\mathrm{y}}_k$ and the diagonal elements of $\widetilde{\Lambda}_x$, resulting in the expression

$$\begin{aligned} \hat{k} &= \arg\max_k \left\{ \log \left[\frac{p(\Delta\mathrm{y}_k; \mathcal{H}_1)}{p(\Delta\mathrm{y}_k; \mathcal{H}_0)} \right] \right\} \\ &= \arg\min_k \left\{ \sum_{i=1}^{p} \left(\frac{\tilde{\lambda}_{xi}}{1 + \tilde{\lambda}_{xi}} \right) \Delta\tilde{y}_{ki}^2 \right\}, \end{aligned} \tag{3.16}$$

where $\tilde{\lambda}_{xi}$ is the ith diagonal element of $\widetilde{\Lambda}_x$ and $\Delta\widetilde{y}_{ki}$ is the ith coordinate of the vector $\Delta\widetilde{\mathrm{y}}_k$.

3.3.4 Hypothesis Testing using SMT

In Section 3.3.2, the generalized eigendecomposition of the covariance matrices R_A and $\widetilde{R}_B$ is a key step for the computation of the log-likelihood test (3.12). We use the SMT to perform the generalized eigendecomposition of $R_A = E_A \Lambda_A E_A^t$ and $\widetilde{R}_B = \widetilde{E}_B \widetilde{\Lambda}_B \widetilde{E}_B^t$, with r_1 and r_2 rotations per coordinate respectively. We apply the SMT for the computation of the following steps:

1. Compute $\mathrm{x}' = \Lambda_A^{-1/2} E_A^t \mathrm{x}$, requiring $(2r_1 + 1)p$ floating-point operations. At the end, we may choose to clip a fraction of the p dimensions and keep only αp of them, with $\alpha \in [0, 1]$.

2. Compute $\widetilde{\mathrm{x}} = \widetilde{E}_B^t \mathrm{x}'$, requiring $(2r_2 + 1)\alpha p$ operations.

3. Compute the sum in (3.12), equiring a total of $2\alpha p$ floating-point operations.

The steps above amount to a total of $[2(r_1 + \alpha r_2) + 3\alpha + 1]p$, i.e, $O(p)$ floating-point operations, where $\alpha \in [0, 1]$. These same steps are used to compute the generalized eigendecomposition of R_x and R_w in Section 3.3.3, and the log-likelihood ratio used in (3.16).

3.4 Experimental Results

3.4.1 Face Recognition

The SMT classification developed in Section 3.3.3 is applied to the task of *face recognition*. We evaluate the SMT-based face recognition with the FERET test protocol and dataset [40], and compare it against the LDA face recognition method [39], a conceptually similar method but that relies on dimensionality reduction to handle the high-dimensional face data. We also compare with a regularized version of LDA using shrinkage covariance estimation. The FERET protocol splits the data into two *disjoint* sets: the *training* set, with face images of 221 individuals/ three different frontal images per individual, and the *gallery* set, with face images of 160 individuals/ four different frontal images per individual. After training the classifier with images of the *training* set, we simulate the recognition process by randomly picking one image from the *gallery* set and searching it against the whole gallery. The system returns all candidates sorted by the likelihood of being a match. If the searched individual appears among the top ρ likely matches in a fraction f of all the searches, we say the rank-ρ recognition rate is f.

Figure 3.1 compares the recognition rates of several classifiers, each using a different method for covariance estimation. The SMT is used both as a standalone method for the covariance estimation, referred as SMT, as well as the shrinkage target, referred as S-SMT. Both SMT-based methods are compared with shrinkage toward identity (S-I) and the LDA face recognition method [39]. The Shrinkage/SMT (S-SMT) performs best among all compared methods. The SMT and Shrinkage/Identity (S-I) methods exhibit almost identical accuracies. Finally, all regularized methods compared are more accurate than the LDA.

As discussed in the Section 3.3.4, the computational cost associated with the application of the SMT is $O(p)$, compared to $O(p^2)$ required to apply the S-SMT and the S-I methods. Therefore, the SMT can be deployed in an environment with limited computational resources delivering competitive accuracy to the one of the computationally expensive shrinkage estimation.

3.4.2 Hyperspectral Image Processing

We use hyperspectral data to measure the performance of the matched filter and the power detector described in Sections 3.3.1 and 3.3.2 respectively.

Figure 3.2 shows the area under the ROC curve for the power detector presented in Section 3.3.2 using several methods. The true covariances R_A and R_B are known. In such scenario, the accuracy of the SMT-based method approaches the one of the exact generalized eigen-decomposition with only a small number of Givens rotations per coordinate.

Figure 3.3 shows the detection capability of the matched filter presented in Section 3.3.1 measured by the $SCRR = SCR/SCR_0$ statistic, where SCR_0 is the value of the ratio in (3.7) for the true covariance R. Therefore, normally we expect $SCRR < 1$. When $SCRR = 1$, the detection accuracy is equivalent of the one in the situation that the true

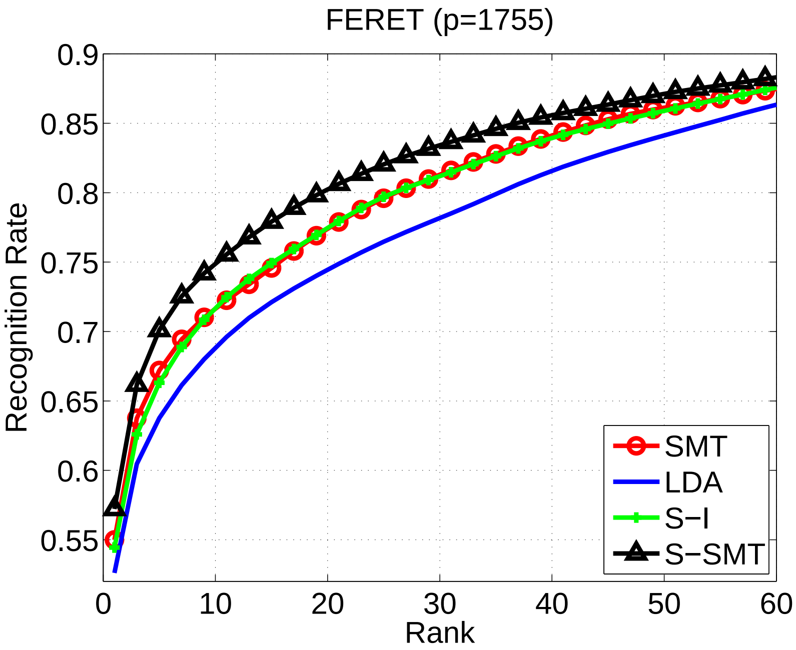


Fig. 3.1. Face recognition rates for ranks 1-60 using different classifiers, SMT, LDA, Shrinkage/Identity (S-I), and Shrinkage/SMT (S-SMT), trained with 221 individuals / 3 images per individual.

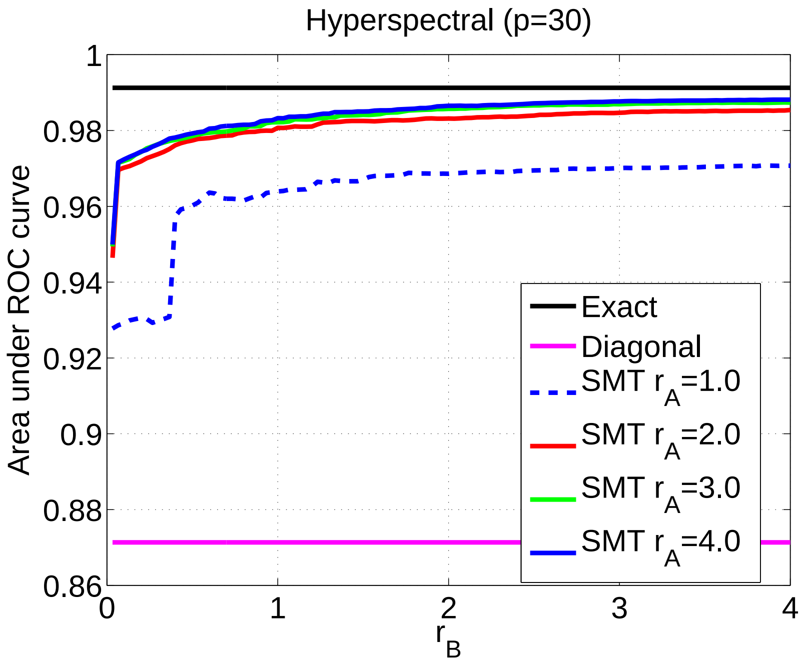


Fig. 3.2. Area under the ROC curve for the SMT as the number of Givens rotations varies. Only a few SMT's Givens rotations are necessary to get most of the detection accuracy given by the exact generalized eigen-decomposition of the true covariance matrices.

covariance R of the clutter is known. We varied the number of training samples n used to estimate $\hat{R}$. The results are averages over 10 trials, each using a different signal t and n different training samples. Notice that the SMT-based detectors perform substantially better than the ones using shrinkage and sample covariance estimates when the training set is small.

3.5 Conclusions

We presented a framework for hypothesis testing in high-dimensional space using the SMT to model the covariance structure of the high-dimensional data. Results show that the SMT methods for detection and classification can have advantages over other methods in the following important aspects. First, the log likelihood ratio test remains robust when few

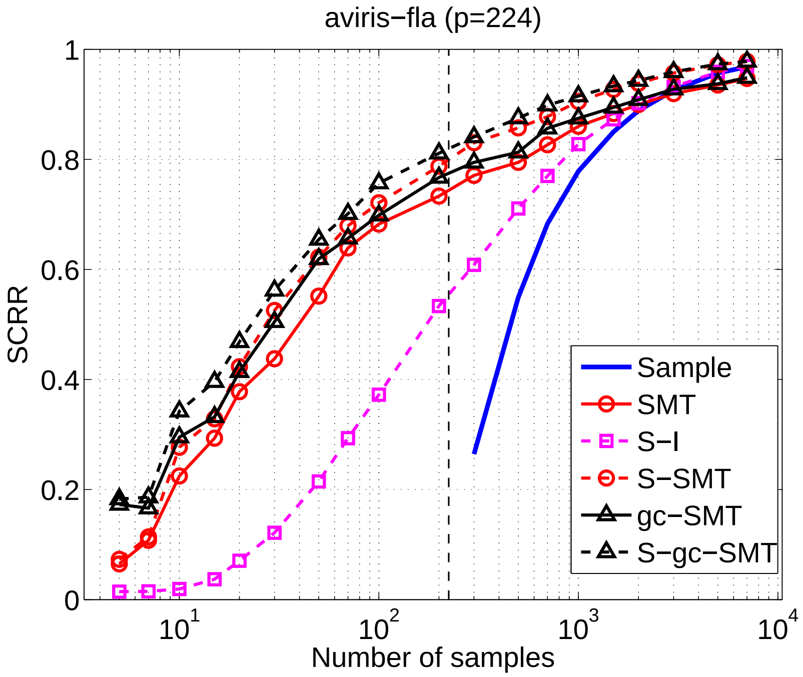


Fig. 3.3. SCRR for hyper-spectral image AVIRIS-FLA using several different estimators: Sample covariance, SMT, Shrinkage /Identity (S-I), Shrinkage/SMT (S-SMT), graphical-SMT (gc-SMT), and Shrinkage/graphical-SMT (S-gc-SMT). Average of 10 trials (each with different signal t and different set of n samples).

training samples are available to train the covariance matrices involved. Second, it operates directly in high-dimensional data at a low computational cost. Finally, the SMT can be used to improve the accuracy of shrinkage estimation when it is computationally feasible.

4. EVALUATING AND IMPROVING LOCAL HYPERSPECTRAL ANOMALY DETECTORS

4.1 Introduction

Anomaly detection promises the impossible: it is target detection without knowing anything about the target. In the context of hyperspectral imagery, the anomalous pixels are those that are unusual with respect to the other pixels in a local or global context. A number of anomaly detectors have been developed for hyperspectral datasets, many of which are surveyed by Stein *et. al.* [41], and more recently by Matteoli *et. al.* [42]

Local detectors form an important class of algorithms. They work using a statistical model of the background pixels in the local neighborhood of the pixel under test. In general, only the pixels within a sliding window are used to estimate properties of the local context. To the extent that the background statistical properties are non-stationary across the image, this local statistical characterization has the potential to improve the detection accuracy. One problem with these local methods is that the number of training samples (pixels), n, needed for a good estimate of the covariance must be at least as large as the data dimensionality (number of spectral bands), p, and preferably should be several times larger than p. [43, 44] This $n \gg p$ requirement rules out small window sizes. The potential increase in detection accuracy due to the local characterization of the background (in a small window) is compromised by the lack of adequate training samples needed to estimate the covariance.

Another way to address the covariance estimation problem is to use the Sparse Matrix Transform (SMT). The SMT provides full rank estimates of large covariance matrices even when the number of training samples n is smaller than the data dimensionality p. [45] We have recently shown that the SMT improves the accuracy of “global” anomaly detectors. [36] In this chapter, we suggest that RX-style detectors using the SMT covariance estimates perform favorably compared to other methods, even in the regime of very small window sizes.

The rest of this chapter is organized as follows: Section 4.2 formulates the anomaly detection task and reviews the most commonly used covariance estimation methods used in anomaly detection; Section 4.3 describes the SMT covariance estimation and how the SMT estimates yield highly accurate detectors even when small window sizes are used; Section 4.4 introduces the mean-log-volume as a measure of detection accuracy and show how it can be used to select the window size that maximizes the detection accuracy; Section 4.5 presents our main experimental results. Finally, Section 4.6 presents the main conclusions.

4.2 Hyperspectral Anomaly Detection

Hyperspectral anomaly detection consists in finding pixel regions (objects) in the hyperspectral image with pixels that differ substantially from the background, *i.e.*, the pixels in the regions surrounding these objects.

In general, there is no precise definition of what constitutes an anomaly. A common way of defining anomalies is to say that *anomalies are not concentrated*. [46] Here we assume that anomalous samples are drawn from a broad, uniform distribution with a much larger support than the distribution of typical (*i.e.,* not anomalous) samples. This assumption allows us to describe anomaly detection in terms of a binary classification problem.

4.2.1 Anomaly Detection as Binary Classification

Let x be a p-dimensional random vector. We want to classify x as *typical* if it is drawn from a multivariate Gaussian distribution $\mathcal{N}(\mu, R)$, or as *anomalous* if it is drawn from a uniform distribution $\mathcal{U}(x) = c$, where c is some constant. Formally, we have the following hypotheses:

$$\begin{aligned} \mathcal{H}_0 &: \quad \mathrm{x} \sim \mathcal{N}(\mu, R) \\ \mathcal{H}_1 &: \quad \mathrm{x} \sim \mathcal{U}, \end{aligned} \tag{4.1}$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ are referred as the *null* and *alternative* hypotheses respectively. According to the *Neyman-Pearson* lemma [35], optimal classifier has the form of a log-likelihood ratio test

$$l(\mathrm{x}) = \log \left\{ \frac{p(\mathrm{x}; \mathcal{H}_1)}{p(\mathrm{x}; \mathcal{H}_0)} \right\} \gtrless l_0, \tag{4.2}$$

that maximizes the probability of detection, $p(\mathcal{H}_1; \mathcal{H}_1)$ for a fixed probability of false alarm, $p(\mathcal{H}_1; \mathcal{H}_0)$, which is controlled by the threshold l_0.

The log-likelihood ratio test in (4.2) can be written as

$$\begin{aligned} l(\mathrm{x}) = \log \left\{ \frac{p(\mathrm{x}; \mathcal{H}_1)}{p(\mathrm{x}; \mathcal{H}_0)} \right\} &= \log c - \log p(\mathrm{x}; \mathcal{H}_0) \\ &= \log c + \frac{p}{2} \log 2\pi + \frac{1}{2} \log |R| \\ &\quad + \frac{1}{2} (\mathrm{x} - \mu)^t R^{-1} (\mathrm{x} - \mu) \gtrless l_0 \end{aligned} \tag{4.3}$$

We can incorporate the constant terms in (4.3) together with l_0 into a new threshold, η, such that the significance test in (4.3) is equivalent to the test

$$D_R(\mathrm{x}) = \sqrt{(\mathrm{x} - \mu)^t R^{-1} (\mathrm{x} - \mu)} \gtrless \eta. \tag{4.4}$$

The statistic $D_R(\mathrm{x})$ is interpreted as the Mahalanobis distance between the sample x and the mean μ of the background distribution. If such distance exceeds the threshold η, we label x as an *anomaly*.

In practice, one does not know the true parameters μ and R of the background pixel distribution $\mathcal{N}(\mu, R)$. In order to compute the statistic $D_R(\mathrm{x})$ in (4.4), the practitioner needs first to compute good estimates $\hat{\mu}$ and $\hat{R}$ of μ and R respectively, from the samples (pixels) available.

4.2.2 Sliding Window-based Detection

The RX detection algorithm [47, 48] uses a sliding window centered at the pixel x, as illustrated in Figure 4.1. The window pixels are used to compute the covariance estimate $\hat{R}$

of the background. As argued in [42] the pixels closest to x within the *Guard window* are left out of the estimation to avoid contaminating the estimate with potentially anomalous pixels. The dimension of the guard window is chosen according to the expected maximum size of an anomalous object. An interesting variation of the RX detector (not investigated here) uses a third window around x, larger than the guard window but smaller than the outer window, to estimate the mean μ. [42] The motivation is that a good estimate of the mean requires fewer pixels than a good estimate of the covariance.

The pixels within the outer window are used as the training pixels in the estimation of the covariance matrix R. The choice of the window size is a compromise between two factors: (i) The window should be small enough that it covers a homogeneous region of the background, therefore, being accurately modeled by the multivariate Gaussian $\mathcal{N}(\mu, R)$; (ii) The window should be large enough that the number of pixels within the outer window is enough to produce reliable estimates of the covariance R. At least $p + 1$ pixels are required for non-singular sample covariance estimates.

4.2.3 Covariance Estimation Methods

In this section, we discuss some of the methods used to estimate the covariance matrix R.

Sample Covariance

Let $\mathrm{X} = [\mathrm{x}_1, \cdots, \mathrm{x}_n]$ be the set of n i.i.d. p-dimensional Gaussian random vectors drawn from $\mathcal{N}(0, R)$. The sample covariance S is given by

$$S = \frac{1}{n}\mathrm{X}\mathrm{X}^t.$$

which is the unconstrained maximum likelihood estimate of R. [35]

When $n < p$, the sample covariance S is singular, with rank n and *overfits* the data. As argued in [42, 43], in the case of hyperspectral data, it is usually desirable to have $n \geq 10p$ so that S is a reliable estimate of R. But even when n is small and S is by

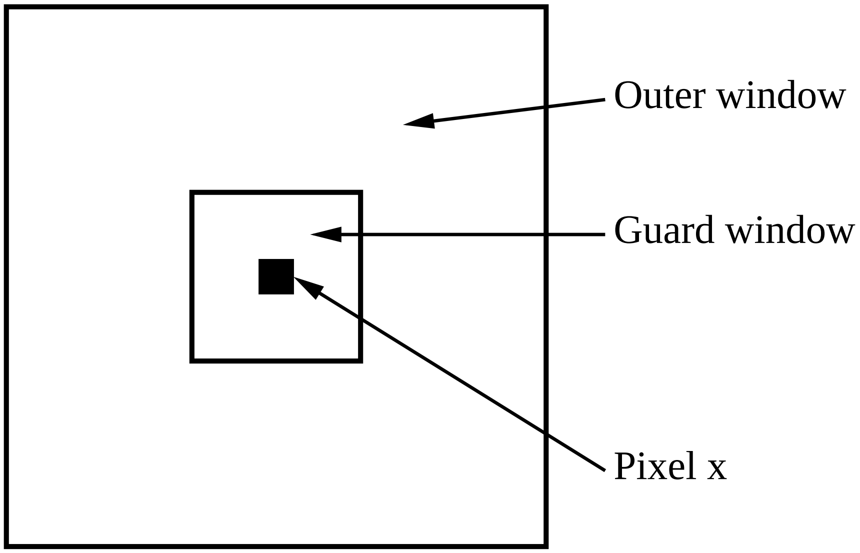


Fig. 4.1. Square sliding window used in the RX detection algorithm. The pixels in the outer window are used to compute the covariance estimate $\hat{R}$ of the background surrounding the pixel x. The pixels within the inner window (referred as the *guard window*) are not used in the covariance computation to avoid that potential anomalous pixels contaminate the estimate $\hat{R}$.

itself unreliable, the sample covariance is still useful as a starting point for the regularized shrinkage estimates reviewed below as well as the SMT introduced in Section 4.3.

Diagonal

Because it is the inverse of R that is used in (4.4), it is important that the estimate of R be full-rank. A simple way to obtain a full-rank estimate of R with a small number of samples n (especially when $n < p$) is to treat all the p dimensions as uncorrelated and simply estimate the variances for each of the p coordinates. This results in the estimator

$$D = diag(S),$$

which is generally of full-rank and can be well estimated even with small n. However, D tends to *underfit* the the data since the assumptions that the coordinates are uncorrelated is typically unrealistic.

Shrinkage

The shrinkage estimation is a very popular method of regularizing estimates of large covariance matrices. [14, 16, 49] It is based on the combination of the sample covariance matrix S that *overfits* the data with another estimator T (called the shrinkage target) that *underfits* the data:

$$\hat{R} = (1 - \alpha)S + \alpha T, \tag{4.5}$$

where $\alpha \in [0, 1]$. The choice of the value α that maximizes the likelihood of the estimate $\hat{R}$ is typically done through a cross-validation procedure.

The most common variation of the shrinkage method [14,16] uses $\sigma^2 I$ as the shrinkage target, where σ^2 is the average variance across all the p dimensions and I is the $p \times p$ identity matrix. The covariance estimator is given by

$$\hat{R} = (1 - \alpha)S + \alpha\sigma^2 I. \tag{4.6}$$

A variation of (4.5) proposed by Hoffbeck and Landgrebe [49] uses $D = diag(S)$ as the shrinkage target, resulting in the following shrinkage estimator

$$\hat{R} = (1-\alpha)S + \alpha D. \tag{4.7}$$

The authors in [49] also propose a computationally efficient leave-one-out cross-validation (LOOC) scheme to estimate α in (4.7). An even more computationally efficient approximation is described in [50].

Quasilocal Covariance

This method proposed by Caefer *et. al.* [51] considers the eigen-decomposition of the covariance matrix $R = E\Lambda E^t$, and makes the observation that the eigenvalues in the matrix Λ are more likely to change across different image locations while the eigenvectors in E remain mostly pointed to the same directions across the entire image.

The observation above suggests that one can obtain a global estimate of the eigenvector matrix E using all the pixels in the image, and then can adjust the eigenvalues in Λ locally by computing the variances independently in each direction using only pixels that are within the sliding window. Since the number of pixels in the entire image, we typically have $n \gg p$, and so the sample covariance S will provide a full-rank global estimate and its eigenvectors, $\hat{E}_{global}$ can be used as the estimates of E across all positions of the sliding window. Finally, the estimate of the matrix Λ is estimated locally at each position of the sliding window, by computing variances in each of the global eigenvector directions. This approach results in the *quasilocal* estimator of covariance:

$$\hat{R} = \hat{E}_{global}\hat{\Lambda}_{local}\hat{E}^t_{global}.$$

4.3 The Sparse Matrix Transform (SMT)

The Sparse Matrix Transform (SMT) [36,45] can be used to provide full-rank estimates of the covariance matrix R used in the detection framework in Section 4.2. The method decomposes the true covariance R into the product $R = E\Lambda E^t$, where E is the orthonormal matrix containing the eigenvectors of R and Λ is a diagonal matrix containing the eigenvalues of R. The SMT then provides the estimates $\hat{E}$ and $\hat{\Lambda}$ with the diagonal elements of $\hat{\Lambda}$ being strictly positive.

4.3.1 SMT Covariance Estimation

Given a training set with n independent p-dimensional i.i.d random vectors drawn from the multivariate Gaussian $\mathcal{N}(0, R)$, and organized into the data matrix $\mathrm{X} = [\mathrm{x}_1, \cdots, \mathrm{x}_n]$. The Gaussian likelihood of observing the data X is given by

$$l(\mathrm{X}; R) = \frac{|R|^{-n/2}}{(2\pi)^{np/2}} \exp\left\{-\frac{1}{2}\mathrm{trace}(R^{-1}S)\right\} \tag{4.8}$$

where $S = \frac{1}{n}\mathrm{X}\mathrm{X}^t$ is the sample covariance, a sufficient statistic for the likelihood of the data X. The joint maximization of (4.8) with respect to E and Λ results in the maximum likelihood (ML) estimates

$$\begin{aligned} \hat{E} &= \arg\min_{E \in \Omega_K} \left\{ \left| \mathrm{diag}(E^t S E) \right| \right\} && (4.9) \\ \hat{\Lambda} &= \mathrm{diag}(\hat{E}^t S \hat{E}) \ , && (4.10) \end{aligned}$$

where Ω_K is the set of allowed orthonormal transforms.

If $n > p$, and the set Ω_K includes all orthonormal transforms, then the solution to (4.9) and (4.10) is given by the sample covariance; *i.e*, $\hat{E}\hat{\Lambda}\hat{E}^t = S$. However, as discussed in Section 4.2, when $n < p$, the sample covariance, S overfits the data and is a poor estimate of the true covariance R.

In order to regularize the covariance estimate, we impose the constraint that Ω_K be the set of sparse matrix transforms (SMT) or order K. More specifically, we will assume that the eigen-transformation has the form

$$E_K = \prod_{k=1}^{K} E_k = E_1 \cdots E_K \in \Omega_K \ , \tag{4.11}$$

for a model order K. Each E_k is a *Givens rotation* [45] over some (i_k, j_k) coordinate pair by an angle θ_k,

$$E_k = I + \Theta(i_k, j_k, \theta_k),$$

where

$$[\Theta]_{ij} = \begin{cases} \cos(\theta_k) - 1 & \text{if } i = j = i_k \text{ or } i = j = j_k \\ \sin(\theta_k) & \text{if } i = i_k \text{ and } j = j_k \\ -\sin(\theta_k) & \text{if } i = j_k \text{ and } j = i_k \\ 0 & \text{otherwise} \end{cases} , \tag{4.12}$$

and K is the model order parameter.

The optimization of (4.9) is non-convex, so we use a greedy optimization approach to design each rotation, E_k, in sequence to minimize the cost [45]: Let $S_{k-1} = E_{k-1}^t S_{k-2} E_{k-1}$. At the kth step of the greedy optimization, we select the pair of coordinates (i_k, j_k) such that

$$(i_k, j_k) = \arg_{i,j} \max \left(\frac{(S_{k-1})_{ij}^2}{(S_{k-1})_{ii}(S_{k-1})_{jj}} \right) ,$$

i.e, the most correlated pair of coordinates, and choose the angle

$$\theta_k = \frac{1}{2} \tan^{-1} \left(\frac{-2(S_{k-1})_{i_k j_k}}{(S_{k-1})_{i_k i_k} - (S_{k-1})_{j_k j_k}} \right)$$

that completely decorrelates the i_k and j_k dimensions. This greedy optimization procedure can be done fast if a graphical constraint can be imposed to the data. [24]

Finally, for an SMT of order K, we have the estimates

$$\hat{E}_K = E_1 \cdots E_K \tag{4.13}$$

$$\hat{\Lambda}_K = \text{diag}(\hat{E}_K^t S \hat{E}_K) , \tag{4.14}$$

with the covariance estimate given by

$$\hat{R}_{SMT} = \hat{E}_K \hat{\Lambda}_K \hat{E}_K^t . \tag{4.15}$$

4.3.2 SMT Model Order

The model order parameter K can be estimated using cross-validation [24,45], a Wishart Criterion [36], or the minimum description length (MDL) approach derived in [36]. We used the MDL criterion for the experiments in this chapter. According to the MDL criterion, we select the smallest value of K such that the following inequality is satisfied:

$$\max_{ij} \left(\frac{[S_K]_{ij}^2}{[S_K]_{ii}[S_K]_{jj}} \right) \leq 1 - \exp\left(\frac{-\log n - 5 \log p}{n} \right) ,$$

where $S_K = \hat{E}_K^t S \hat{E}_K$.

It is often useful to express the order of the SMT as $K = rp$, where r is the average number of rotations per coordinate, being typically very small ($r < 5$) for several previously studied datasets.

4.3.3 Shrinkage SMT

The SMT covariance estimate in (4.15) can be used as a shrinkage target, alternative to the ones described in Section 4.2.3, resulting in the following Shrinkage-SMT estimate:

$$\hat{R} = (1 - \alpha)S + \alpha \hat{R}_{SMT} \; .$$

4.4 Ellipsoid Mean Log-Volume

In this section, we develop the *Ellipsoid Mean Log-Volume*, a novel metric to evaluate the accuracy of anomaly detection algorithms that make detection decisions based on a Mahalanobis statistic such as D_R in (4.4). Different versions of these detectors use different techniques to estimate the covariance yielding different detection accuracies depending on how well the covariance estimate $\hat{R}$ approximates the true background covariance R.

Traditionally, receiver operating characteristics (ROC) curves have been widely used to evaluate anomaly detectors. The ROC approach requires both samples labeled as *typical* and samples labeled as *anomalous* in order to estimate the both the *probability of detection* and the *probability of false alarm* used in the ROC analysis. Unfortunately, anomalies are rare events and it is often difficult to have enough data labeled as *anomalous* in order to estimate the probability of detection required in the ROC analysis.

The approach developed here seeks to characterize how well the estimates of the background model (*i.e.*, $\hat{\mu}$ and $\hat{R}$) fit the training (typical) pixel data, overcoming the limitation of the ROC analysis described above. More specifically, we evaluate the volume of the hyper-ellipsoid within the region

$$(\mathrm{x} - \hat{\mu})^t \hat{R}^{-1} (\mathrm{x} - \hat{\mu}) \leq \eta^2, \tag{4.16}$$

where η controls the probability of false alarm, as described previously. Such a volume is evaluated by the following expression:

$$V(R,\eta) = \frac{\pi^{p/2}\sqrt{|R|}}{\Gamma(1+p/2)}\,\eta^p. \tag{4.17}$$

Smaller values of $V(R,\eta)$ indicate smaller probabilities that an anomalous data point would fall within the hyper-ellipsoid region of (4.16). Based on this observation, the core idea in our approach is to use the value of $V(R,\eta)$ as a proxy for the probability of missed detection. Therefore, for a fixed probability of false alarm, smaller values of $V(R,\eta)$ indicate more accurate detection. Because the direct computation of $V(R,\eta)$ tends to be numerically unstable, often leading to numerical overflow for large values of p, in practice we work with $\log V(R,\eta)$ as our measure of accuracy.

This approach has been used before in global anomaly detection [36, 52, 53], but we are extending it here to local sliding window-based anomaly detection. These detectors produce a different local estimate of the background covariance at each location of the sliding window across the image. We suggest measuring detection accuracy in terms of the expected log-volume of the hyper-ellipsoid, $\mathbb{E}[\log V(\hat{R},\eta)]$ across the whole hyperspectral image, where each different estimate $\hat{R}$ is computed for each position of the sliding window using local training data pixels.

4.5 Experiments

All experiments in this section were performed using the *Blindrad* hyperspectral dataset, a HyMap image of Cooke City, MT of 800×280 pixels, [54] each with 126 hyperspectral bands. Figure 4.2 displays a RGB rendering of this dataset.

In all experiments, a sliding window like the one described in Figure 4.1 moves across the image and, at each position it estimates the covariance R from the samples of the outer window using several covariance estimation methods previously discussed. Such covariance is used to compute D_R in (4.4) for each pixel within the guard window. The radius η is adjusted globally so that a fraction of the points corresponding to a fixed probability of false alarm is left out of the ellipsoid region. Finally we compute the expected value

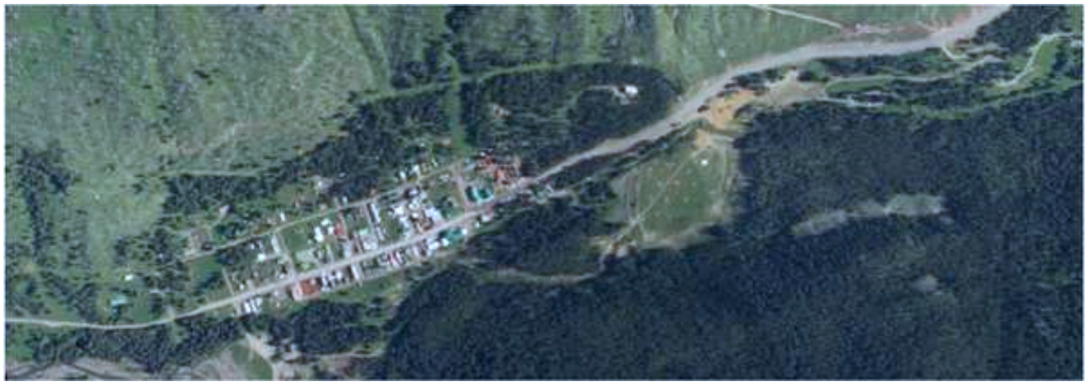


Fig. 4.2. RGB rendering of the 800×280 pixel *Blindrad* hyperspectral dataset, captured using a HyMap sensor with 126 channels.

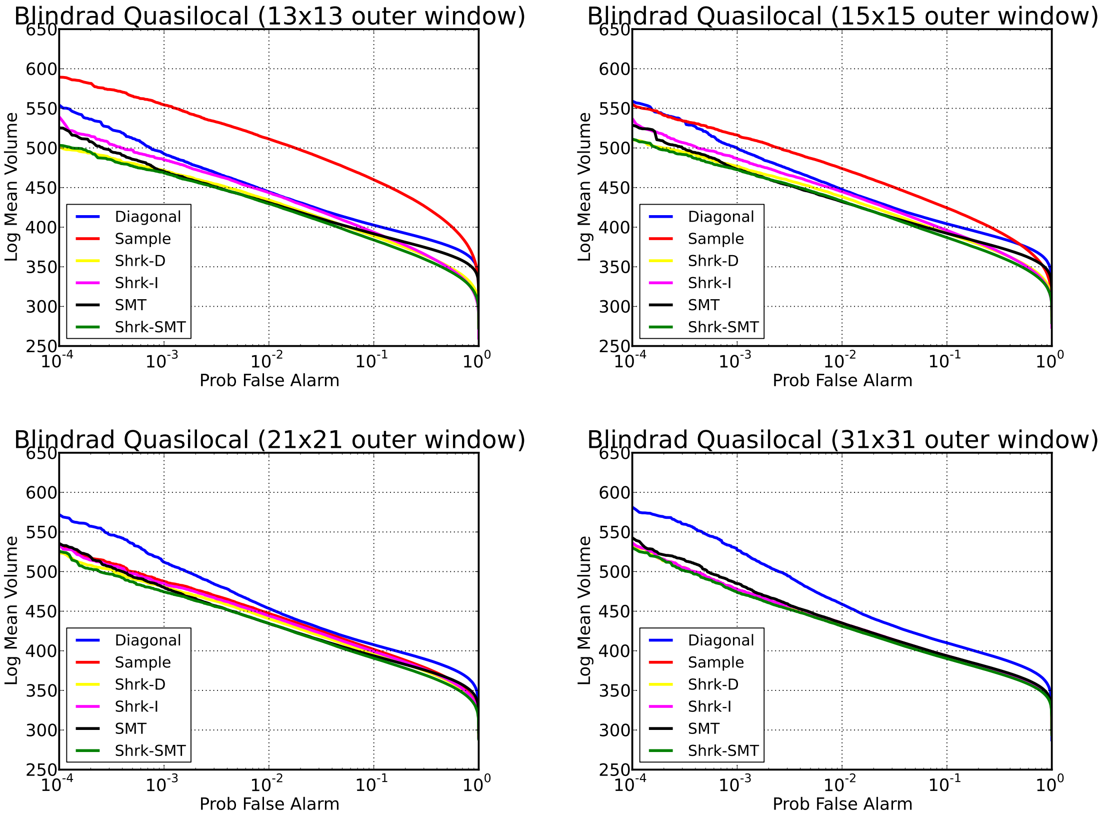
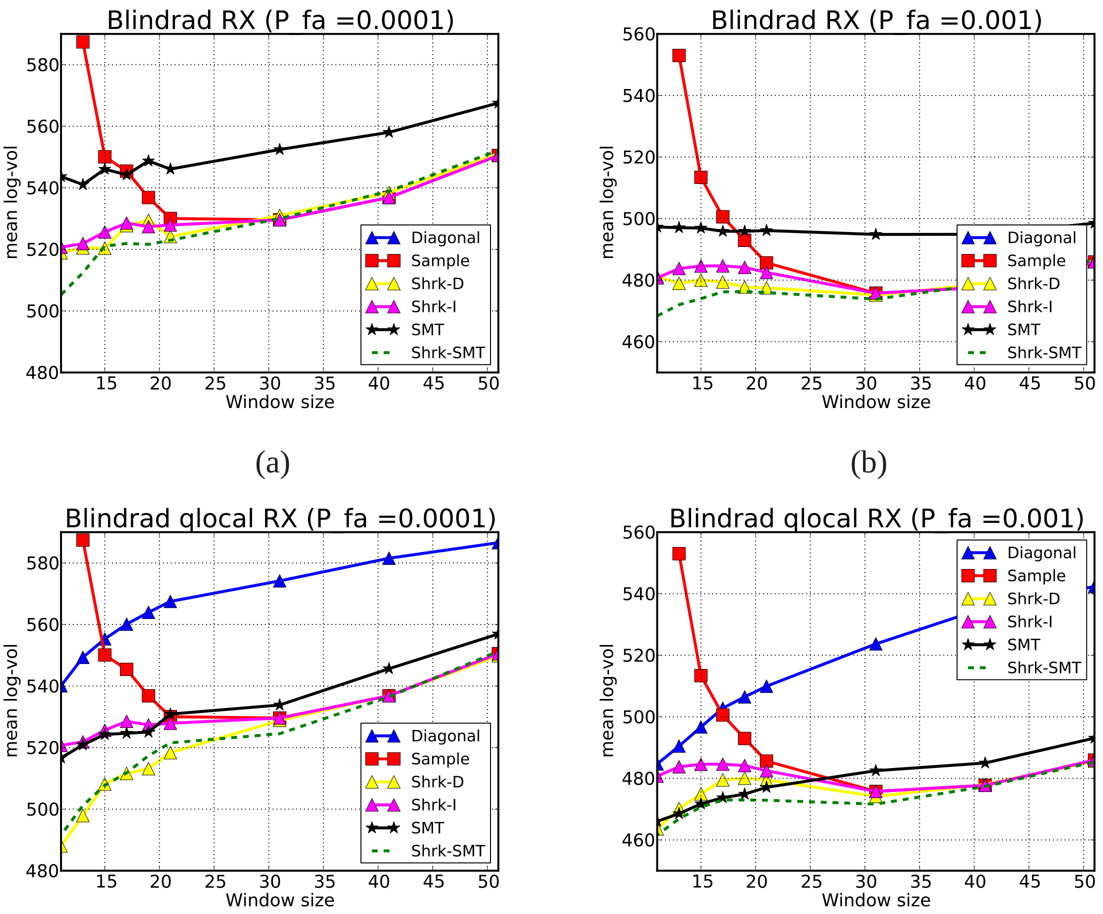


Fig. 4.3. Coverage plots with the expected ellipsoid log-volume *vs.* probability of false alarm for various outer window sizes.

$\mathbb{E}[\log V(\hat{R}, \eta)]$ over all window positions and take that as the measure of anomaly detection performance.

Figure 4.3 shows the *coverage plots* with the expected log-volume of ellipsoid *vs.* the probability of false alarm for different window sizes. The hyperspectral bands of the dataset were rotated to the *Quasilocal* coordinate system by the matrix $\hat{E}^t_{global}$ (see Sec. 4.2.3). These “ROC-like” curves suggest that the regularized methods are more accurate, especially when small window sizes are used. When large window sizes are used, the unregularized sample covariance has its performance similar to the regularized methods.

Figure 4.4 compares the performance of several detectors in both the original and the quasilocal coordinate systems at two different fixed false alarm rates. The diagonal co-



(a)

(b)

(c)

(d)

Fig. 4.4. Expected ellipsoid log-volume *vs.* the dimension of the sliding window fixed probabilities of false alarm in both the original, (a) and (b), and the quasilocal, (c) and (d), coordinate systems.

variance estimate performs poorly in the original coordinates (Figures 4.4(a) and 4.4(b)), but remains a competitive method in the quasilocal coordinates (Figures 4.4(c) and 4.4(d)); in fact, the diagonal estimator in quasilocal coordinates is just the quasilocal covariance estimator suggested by Caefer *et. al.* [51]. The Shrinkage-SMT estimates are among the best methods in both spaces, though in the quasilocal space, Shrinkage-Diagonal detectors perform just as well. When the window size used to estimate the covariance matrix grows large, we observe the increase in the expected ellipsoid log-volume; *i.e.*, the degradation of the detection accuracy for all the methods. This degradation is due to the distribution of the background pixels being non-stationary across the image. Therefore, the estimate of the covariance using large windows tends to yield poor estimates. When small window sizes are used, the training pixels are more likely to come from a homogeneous region with Gaussian distribution. Nevertheless, this is a regime where poor estimates of the covariance are due to the limited number of training samples, as observed in the curves for detectors using the sample covariance. On the other hand, the results suggest that the regularized methods perform best with smaller window sizes. Finally, the practitioner can use the curves in Figure 4.4 as a criterion to select the window size that produces the most accurate detector for a chosen covariance estimation method.

4.6 Conclusions

In this chapter we have shown how to use the expected log-volume of ellipsoid to measure local detector accuracy. This measure was used to compare different detectors as well as a to provide a criterion for selecting the optimal size of the sliding window. We have also shown how to use the SMT to produce regularized covariance estimates to be used in detection. While Shrinkage-SMT often produces good results, our results show that Shrinkage-Diagonal performs just as well when combined with the quasilocal method proposed in [51]. In the future, we plan to address how to push the covariance methods to work with even smaller window sizes.

5. DISTRIBUTED SIGNAL DECORRELATION AND DETECTION IN WIRELESS SENSOR NETWORKS USING THE SPARSE MATRIX TRANSFORM

5.1 Introduction

In recent years, there has been significant interest in the use of sensor networks for distributed monitoring in many applications [1, 4]. In particular, networks with camera sensors have gained significant popularity [2, 3]. Consider the scenario where all cameras collectively monitor the same environment. Each camera registers an image of the environment from its specific viewpoint and encodes it into a vector output. As the number of deployed cameras grows, so does the combined data generated from all cameras. Because these cameras usually operate under limited battery power and narrow communication bandwidth, this data deluge created in large networks imposes serious challenges to the way data is communicated and processed.

Event detection and more specifically anomaly detection are important applications for many sensor networks [55]. In general, the vector outputs from all sensors in a network can be concatenated to form a single p-dimensional vector x, and then the goal of anomaly detection is to determine if x corresponds to a typical or anomalous event. Figure 5.1 illustrates this scenario for a network of cameras. The vector outputs from different cameras in the network are likely to be correlated, particularly when the cameras capture overlapping portions of the scene; so for best detection accuracy, vector x should be decorrelated as part of the detection process.

One possible approach to decorrelate x is to have all cameras send their vector outputs to a single sink node. This approach has several problems because it puts a disproportional and unscalable burden on the sink and on the communication links leading to it. One possible solution is to design a more powerful sink node. Unfortunately, having a powerful sink node is not a suitable solution for the many applications that require nodes to operate in an ad hoc manner [56, 57], re-arranging themselves dynamically.

Alternatively, each sensor can compute the likelihood of its vector measurement independently and send a single (scalar) likelihood value to the sink, which then combines the likelihoods computed by each sensors and makes a detection decision. While requiring minimal communication energy, this approach does not model correlations between camera outputs, potentially leading to poor detection accuracy.

Because of the limitations above, there is a need for distributed algorithms which can decorrelate vector camera outputs without use of a centralized sink, while keeping the communication among sensors low. Several methods to compute distributed Karhunen-Loéve transform (KLT) and principal components analysis (PCA) in sensor networks have been proposed. Distributed PCA algorithms are proposed in [58] and [59]. Both methods operate on scalar sensor outputs, and in order to constrain communication in the network, they assume that sensor outputs are conditionally independent given the outputs of neighboring sensors. A distributed KLT algorithm is proposed in [60–63] to compress/encode vector sensor outputs with the subsequent goal of reconstructing the aggregated output at the sink node with minimum mean-square error. Distributed decorrelation using a wavelet transform with lifting has been studied for sensor networks with a linear topology [64], two-dimensional networks [65], and networks with tree topology [66]. While assuming specific network topologies and correlation models for scalar sensor outputs, these methods focus mainly on efficient data gathering and routing when sensor measurements are correlated. Also, these methods do not take into consideration that sensors far apart in the network can generate highly correlated outputs, as in the case when two cameras pointing to the same event, and therefore producing correlated outputs, can be several hops apart from each other, as argued in [67].

Multiple efforts have been made in distributed detection since the early 1980s (see [68] for a survey). Most of the approaches rely on encoding scalar sensor outputs efficiently to cope with low communication bandwidth and transmitting encoded outputs to a fusion center in charge of making final detection decisions. More recently, detection of volume anomalies in networks have been studied in [69–71]. These approaches focus on scalar measurements in network links and rely on centralized data processing for anomaly detection. Several methods for video anomaly detection have been proposed (see [72] for a survey). The method in [71] uses multi-view images of a highway system to detect traffic anomalies, with each view monitoring a different road segment or intersection. The processing of the multiple views is non-distributed and the method does not model any correlations between views.

Accurate anomaly detection requires decorrelation of the background signal. In order to decorrelate the background, we need an accurate estimate of its covariance matrix. Several methods to estimate covariances of high-dimensional signals have been proposed recently [**?**, 11, 12, 17, 45, 49]. Among these methods, the Sparse Matrix Transform (SMT) [45], here referred as scalar SMT, has been shown to be effective, providing full-rank covariance estimates of high-dimensional signals even when the number n of training samples used to compute the estimates is much smaller than the dimension p of a data sample, i.e, $n \ll p$. Furthermore, the decorrelating transform designed by the SMT algorithm consists of a product of $O(p)$ Givens rotations, and therefore, it is computationally inexpensive to apply. The scalar SMT has been used in detection and classification of high-dimensional signals [29, 30, 36]. Because it involves only pairwise operations between coordinate pairs, it is well suited to distributed decorrelation [73]. However, this existing method is only well suited for decorrelation of scalar sensor outputs.

In this chapter, we propose the vector sparse matrix transform (vector SMT), a novel algorithm suited for distributed signal decorrelation in sensor networks where each sensor outputs a vector. It generalizes the concept of the scalar sparse matrix transform in [45] to decorrelation of vectors. This novel algorithm operates on pairs of sensor outputs, and it has the interpretation of maximizing the constrained log likelihood of x. In particular, the vector SMT decorrelating transform is defined as an orthonormal transformation constrained to be formed by a product of pairwise transforms between pairs of vector sensor outputs. We design this transform using a greedy optimization of the likelihood function of x. Once this transform is designed, the associated pairwise transforms are applied to sensor outputs distributed over the network, without the need of a powerful central sink node. The total number of pairwise transforms is a model order parameter. By constraining the value of this model order parameter to be small, our method imposes a sparsity constraint to the data. When this sparsity constraint holds for the data being processed, the vector SMT can substantially improve the accuracy of the resulting decorrelating transform even when a limited number of training samples is available.

Being able to perform distributed decorrelation while consuming limited communication energy is an important characteristic of our method. Our primary way of limiting energy consumption is to select the model order parameter value such that the total energy required for distributed decorrelation is less than a specified budget. Another approach to limit energy consumption is based on constrained likelihood optimization using Lagrange multipliers. Because sensor pairs that are far apart can be highly correlated, the unconstrained greedy optimization of the likelihood of x may result in pairwise transforms between sensors that are far apart, requiring prohibitive amounts of energy. To limit energy consumption in such a scenario, we constrain the greedy optimization of the likelihood function by adding to it a linear penalization term that models the energy required by the associated decorrelating transform. As a result, during the design of decorrelating transformation, our method selects sensor pairs based on the correlation between their outputs while penalizing the ones that are several hops apart and require high energy consumption for their pairwise transforms.

We introduce the new concept of a correlation score, a measure of correlation between two vectors. This correlation score generalizes the concept of correlation coefficient to pairs of random vectors. In fact, we show that the correlation score between two scalar random variables is the absolute value of their correlation coefficient. We use this correlation score to select pairs of most correlated sensor outputs during the design of the vector SMT decorrelating transform, as part of the greedy optimization of the likelihood of x. We remark that this concept is closely related to the concepts of mutual information between two random vectors [27], and their total correlation [28].

To validate our method, we describe experiments using simulated data, artificially generated multi-camera image data of 3D spheres, and real multi-camera data of a courtyard. We use the vector SMT to decorrelate the data from multiple cameras in a simulated network for the purpose of anomaly detection. We compare our method against centralized and independent approaches for processing the sensor outputs. The centralized approach relies on a sink node to decorrelate all sensor outputs and requires a large amount of energy to communicate all sensor data. The independent approach relies on each sensor to compute its partial likelihood of its output independently from the others and communicate the resulting value to the sink that makes the final detection decision. While minimizing communication energy, this independent approach leads to poor detection accuracy since it does not take into account correlations between sensor outputs. Our results show that the vector SMT decorrelation enables consistently more accurate anomaly detection across the experiments while keeping the communication energy required for distributed decorrelation low.

The rest of this chapter is organized as follows: Section 5.2 describes the main concepts of the scalar SMT. Section 5.3 introduces the vector SMT algorithm, designed to perform distributed decorrelation of vector sensor outputs in a sensor network. Section 5.4 shows how to use the vector SMT to enable distributed detection in a sensor network. Section 5.5 shows simulation results of detection using multi-camera views of objects. Finally, the main conclusions and future work are discussed in Section 5.6.

5.2 The Scalar Sparse Matrix Transform

Let x be a p-dimensional random vector from a multivariate, Gaussian distribution, $\mathcal{N}(0, R)$. Moreover, the covariance matrix, R can be decomposed into $R = E\Lambda E^t$, where Λ is a diagonal matrix and E is orthonormal. The Sparse Matrix Transform (SMT) [45] models the orthonormal matrix E as the product of K sparse matrices, E_K, so that

$$E = \prod_{k=1}^{K} E_k = E_1 \cdots E_K \; . \tag{5.1}$$

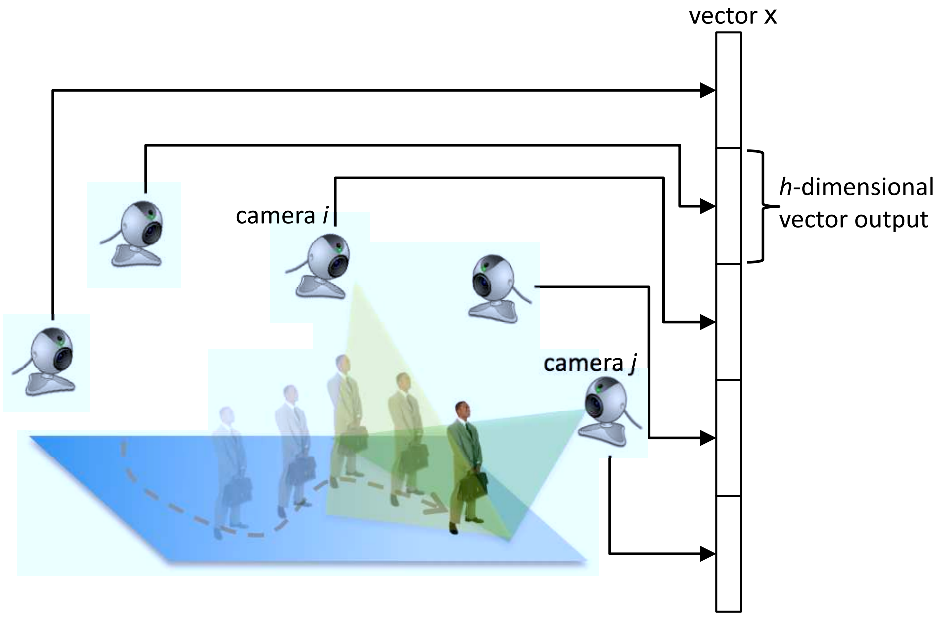


Fig. 5.1. A camera network where each camera captures an image of the environment from one viewpoint and encodes the image into a vector output. The aggregated outputs from all cameras form the high-dimensional vector, x. Cameras i and j have overlapping views. Because outputs from cameras with overlapping views tend to be correlated, so does the aggregated vector x.

In (5.1), each sparse matrix E_k, known as a Givens rotation, is a planar rotation over a coordinate pair (i_k, j_k) parametrized by an angle θ_k, i.e,

$$E_k = I + \Theta(i_k, j_k, \theta_k) \ , \tag{5.2}$$

where

$$[\Theta]_{ij} = \begin{cases} \cos(\theta_k) - 1 & \text{if } i = j = i_k \text{ or } i = j = j_k \\ \sin(\theta_k) & \text{if } i = i_k \text{ and } j = j_k \\ -\sin(\theta_k) & \text{if } i = j_k \text{ and } j = i_k \\ 0 & \text{otherwise} \end{cases} \ . \tag{5.3}$$

This SMT model assumes that K Givens rotations in (5.1) are sufficient to decorrelate the vector x. Each matrix, E_k operates on a single coordinate pair of x, playing a role analogous to the decorrelating "butterfly" in the fast Fourier Transform (FFT). Because both the ordering of coordinate pairs (i_k, j_k), and the values of rotation angles θ_k are unconstrained, the SMT can model a larger class of signal covariances than the FFT. In fact, the scalar SMT is a generalization of both the FFT and the orthonormal wavelet transform. Figures 5.2(b) and (c) make a visual comparison of both the FFT and the Scalar SMT. The SMT rotations can operate on pairs of coordinates in any order, while in the FFT case, the butterflies are constrained to a well defined sequence with specific rotation angles.

The scalar SMT design consists in learning the product in (5.1) from a set of n independent and identically distributed training vectors, $\mathrm{X} = [\mathrm{x}_1, \cdots, \mathrm{x}_n]$, from $\mathcal{N}(0, R)$. Assuming that $R = E\Lambda E^t$, the maximum likelihood estimates of E and Λ are given by

$$\begin{aligned} \hat{E} &= \arg\min_{E \in \Omega_K} \left\{ \left| \mathrm{diag}(E^t S E) \right| \right\} && (5.4) \\ \hat{\Lambda} &= \mathrm{diag}(\hat{E}^t S \hat{E}) \ , && (5.5) \end{aligned}$$

where $S = \frac{1}{n}\mathrm{X}\mathrm{X}^t$, and Ω_K is the set of allowed orthonormal transforms. With the SMT model assumption, the orthonormal transforms in Ω_K are in the form of (5.1), and the total number of planar rotations, K is the model order parameter.

When performing an unconstrained minimization of (5.4) by allowing the set Ω_K to contain all orthonormal transforms, when $n > p$, the minimizer is the orthonormal matrix that diagonalizes of the sample covariance, i.e., $\hat{E}\hat{\Lambda}\hat{E}^t = S$. However, S is a poor estimate of R when $n < p$. As shown in [45], the greedy optimization of (5.5) under the constraint that the allowed transforms are in the form of (5.1) yields accurate estimates even when $n \ll p$.

The constraint in (5.1) is non-convex with no obvious closed form solution. In [45], we use a greedy optimization approach in which we select each Givens rotation, E_k, independently, in sequence to minimize the cost in (5.4). The model order parameter K can be estimated using cross-validation over the training set [23, 24] or using the minimum description length (MDL) [36].

Typically, the average number of rotations per coordinate, K/p is small (< 5), so that the computation to apply the SMT to a vector of data is very low, i.e, $2(K/p) + 1$ floating-point operations per coordinate. Finally, when $K = \binom{p}{2}$, the SMT factorization of R is equal to its exact diagonalization, a process known as Givens QR.

5.3 Distributed Decorrelation with the Vector Sparse Matrix Transform

The vector Sparse Matrix Transform (vector SMT) is the core of our approach for distributed decorrelation of vector sensor outputs in sensor networks. Our goal is to decorrelate the p-dimensional vector x aggregated from outputs of all sensors, where each sensor outputs a sub-vector of x after sensing the environment. The vector SMT operates on x by decorrelating a sequence of pairs of its sub-vectors. This vector SMT generalizes the concept of the scalar SMT in Section 5.2 to the decorrelation of pairs of vectors instead of pairs of coordinates.

5.3.1 The Vector SMT Model

Let the p-dimensional vector x be partitioned into L sub-vectors,

$$\mathrm{x} = \left[\begin{array}{c} \mathrm{x}^{(1)} \\ \hline \vdots \\ \hline \mathrm{x}^{(L)} \end{array} \right] ,$$

where each sub-vector, $\mathrm{x}^{(i)}$ is an h-dimensional vector output from a sensor $i = 1, \cdots, L$ in a sensor network. A vector SMT is an orthonormal $p \times p$ transform, T, written as the product of M orthonormal, sparse matrices,

$$T = \prod_{m=1}^{M} T_m \ , \tag{5.6}$$

where each pairwise transform, $T_m \in \mathbb{R}^{p \times p}$, is a block-wise sparse, orthonormal matrix that operates exclusively on the $2h$-dimensional subspace of the sub-vector pair $\mathrm{x}^{(i_m)}$, $\mathrm{x}^{(j_m)}$, as illustrated in Figure 5.2(a). The decorrelating transform is then formed by the product of the M pairwise transforms, where M is a model order parameter.

Each T_m is a generalization of a Givens rotation in (5.2) to a transform that operates on pairs of sub-vectors instead of coordinates. Similarly, the vector SMT in (5.6) generalizes the concept of the scalar SMT in Section 5.2: it decorrelates a high-dimensional vector by decorrelating its pairs of sub-vectors instead of pairs of coordinates. Figures 5.2(b) and (d) compare both the vector and the scalar SMTs approaches graphically. In the scalar SMT, each Givens rotation E_k plays the role of a "decorrelating butterfly" (Figure 5.2(b)) that together decorrelate x. In the vector SMT, each orthonormal matrix T_m corresponds to series of decorrelating butterflies that operate exclusively on coordinates of a single pair of sub-vectors of x. Finally, the sequence in (5.6), illustrated in Figure 5.2(d), decorrelates M pairs of sub-vectors of x, until the decorrelated vector $\widetilde{\mathrm{x}}$ is obtained.

In a sensor network, we compute the distributed decorrelation of x by distributing the application of transforms T_m from the product (5.6) across multiple sensors. Before the decorrelation, each sub-vector $\mathrm{x}^{(i)}$ of x is the output of a sensor i and is stored locally in that sensor. Applying each T_m to sub-vectors $\mathrm{x}^{(i_m)}$, $\mathrm{x}^{(j_m)}$ requires point-to-point communication of one h-dimensional sub-vector between sensors i_m and j_m, consuming an amount of energy, $\mathcal{E}(h, i_m, j_m)$, proportional to some measure of the distance between these sensors. After applying T_m, the resulting decorrelated sub-vectors $\tilde{\mathrm{x}}^{(i_m)}$ and $\tilde{\mathrm{x}}^{(j_m)}$ are cached at the sensor used to compute this pairwise decorrelation, avoiding communicating one

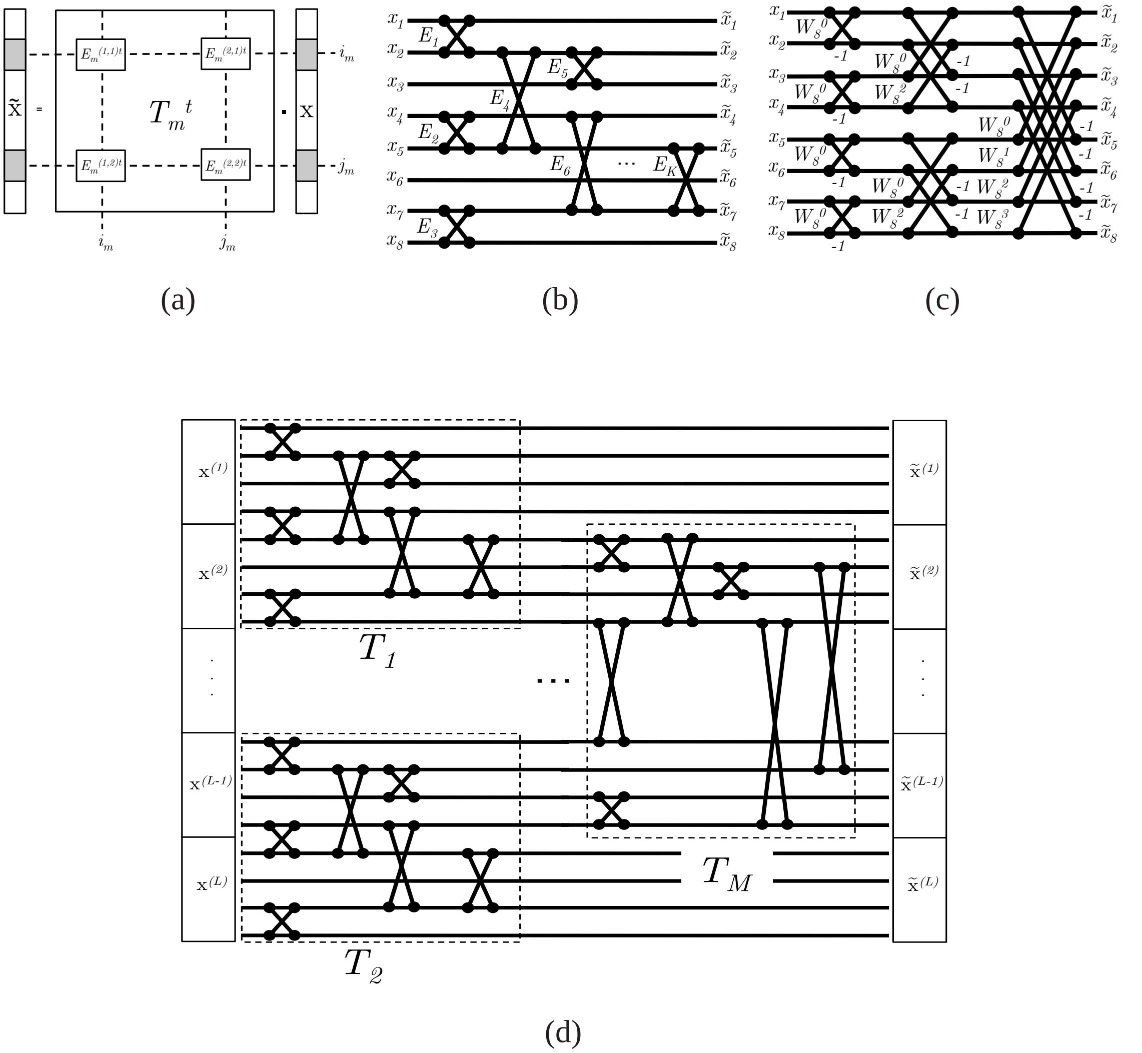

x̃
=
$E_m^{(1,1)t}$
$E_m^{(2,1)t}$
$E_m^{(1,2)t}$
$E_m^{(2,2)t}$
$T_m{}^t$
x
i_m
j_m
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8
E_1 E_2 E_3 E_4 E_5 E_6 ⋯ E_K
$\tilde{x}_1$ $\tilde{x}_2$ $\tilde{x}_3$ $\tilde{x}_4$ $\tilde{x}_5$ $\tilde{x}_6$ $\tilde{x}_7$ $\tilde{x}_8$
W_8^0 W_8^1 W_8^2 W_8^3
-1
(a)
(b)
(c)
$x^{(1)}$ $x^{(2)}$ $x^{(L-1)}$ $x^{(L)}$
$\tilde{x}^{(1)}$ $\tilde{x}^{(2)}$ $\tilde{x}^{(L-1)}$ $\tilde{x}^{(L)}$
T_1
T_2
T_M
⋯
(d)

Fig. 5.2. (a) In the product $\tilde{x} = T_m^t x$, the transform T_m operates over the p-dimensional vector x, changing only the components associated with the sub-vectors $x^{(i_m)}, x^{(j_m)}$ (shaded). (b) scalar SMT decorrelation, $\tilde{x} = E^t x$. Each E_k plays the role of a decorrelating "butterfly", operating on a single pair of coordinates. (c) 8-point FFT, seen as a particular case of the scalar SMT where the butterflies are constrained in their ordering and rotation angles. (d) Vector SMT decorrelation, $\tilde{x} = T^t x$, with each T_m decorrelating a sub-vector pair of x instead of a single coordinate pair. T_m is an instance of the scalar SMT with decorrelating butterflies operating only on coordinates of a single pair of sub-vectors.

sub-vector back to its originating sensor. Finally, the total communication energy required for the entire decorrelation is given by

$$\mathcal{E}(h, i_1, \cdots, i_M, j_1, \cdots, j_M) = \sum_{m=1}^{M} \mathcal{E}(h, i_m, j_m). \tag{5.7}$$

5.3.2 The Design of the Vector SMT

We design the vector SMT decorrelating transform from training data, using the maximum likelihood estimation of the data covariance matrix. Let $\mathrm{X} = [\mathrm{x}_1, \cdots, \mathrm{x}_n] \in \mathbb{R}^{p \times n}$, be a $p \times n$ matrix where each column, x_i is a p-dimensional zero mean Gaussian random vector with covariance R. In general, a covariance can decomposed as $R = T\Lambda T^t$, where Λ is the diagonal eigenvalue matrix and T is an orthonormal matrix. In this case, the log likelihood of X given the T and Λ is given by

$$\log p_{(T,\Lambda)}(\mathrm{X}) = -\frac{n}{2}\mathrm{trace}[\mathrm{diag}(T^t S T)\Lambda^{-1}] - \frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\Lambda| \ , \tag{5.8}$$

where

$$S = \frac{1}{n}\mathrm{X}\mathrm{X}^t \ . \tag{5.9}$$

When constraining T to be of the product form of (5.6), the joint maximum likelihood estimates $\widehat{\Lambda}$ and $\widehat{T}$ are given by

$$\begin{aligned} \hat{T} &= \arg \min_{T=\prod_{m=1}^{M} T_m} \left\{ \left| \mathrm{diag}(T^t S T) \right| \right\} && (5.10) \\ \hat{\Lambda} &= \mathrm{diag}(\hat{T}^t S \hat{T}) \ . && (5.11) \end{aligned}$$

Because the minimization in (5.10) has a non-convex constraint, its global minimizer is difficult to find. Therefore, we use a greedy procedure that designs each new T_m, $m = 1, \cdots, M$ independently while keeping the others fixed. We start by setting $S_1 = S$ and $\mathrm{X}_1 = \mathrm{X}$, and iterate over the following steps:

$$\begin{aligned} \hat{T}_m &= \arg \min_{T_m \in \Omega} \left\{ \left| \mathrm{diag}(T_m^t S_m T_m) \right| \right\} && (5.12) \\ S_{m+1} &= \hat{T}_m^t S_m \hat{T}_m && (5.13) \\ \mathrm{X}_{m+1} &= \hat{T}_m^t \mathrm{X}_m \ , && (5.14) \end{aligned}$$

where Ω is the set of all allowed pairwise transforms. Because T_m operates exclusively on $\mathrm{x}^{(i_m)}$ and $\mathrm{x}^{(j_m)}$, once the pair (i_m, j_m) is selected, the design of T_m involves only the components of X_m associated with these sub-vectors. Let $\mathrm{X}_m^{(i_m)}$ and $\mathrm{X}_m^{(j_m)}$ be $h \times n$ sub-matrices of X_m associated with the sub-vector pair (i_m, j_m). Their associated $2h \times 2h$ sample covariance is then given by

$$S_m^{(i_m,j_m)} = \frac{1}{n} \left[\begin{array}{c} \mathrm{X}_m^{(i_m)} \\ \hline \mathrm{X}_m^{(j_m)} \end{array} \right] \left[\mathrm{X}_m^{(i_m)t} | \mathrm{X}_m^{(j_m)t} \right] \ . \tag{5.15}$$

The minimization in (5.12) for a fixed subvector pair (i_m, j_m) can be recast in terms of $S^{(i_m,j_m)}$, and the $2h \times 2h$ orthonormal matrix E,

$$E_m = \arg \min_{E \in \Omega_{2h\times 2h}} \left\{ |\mathrm{diag}(E^t S_m^{(i_m,j_m)} E)| \right\} \ , \tag{5.16}$$

where $\Omega_{2h\times 2h}$ is the set of all valid $2h \times 2h$ orthonormal transforms. In practice, the optimization of E is precisely the same problem as the scalar SMT design presented in Section 5.2. Once E_m is selected, we partition it into four $h \times h$ blocks,

$$E_m = \left[\begin{array}{c|c} E_m^{(1,1)} & E_m^{(1,2)} \\ \hline E_m^{(2,1)} & E_m^{(2,2)} \end{array} \right] \ ,$$

and then we obtain the transform T_m using Kronecker product $\otimes$ as

$$\begin{aligned} T_m \ &= \ J^{(i_m,i_m)} \otimes E_m^{(1,1)} + J^{(i_m,j_m)} \otimes E_m^{(1,2)} \\ &+ \ J^{(j_m,i_m)} \otimes E_m^{(2,2)} + J^{(j_m,j_m)} \otimes E_m^{(2,1)} \ , \\ &+ \ I_{p\times p} - \left(J^{(i_m,i_m)} + J^{(j_m,j_m)} \right) \otimes I_{h\times h} \end{aligned} \tag{5.17}$$

where $J^{(i,j)}$ is a $L \times L$ matrix given by

$$\left[J^{(i,j)} \right]_{i'j'} = \begin{cases} 1 & \text{if } i' = i \text{ and } j' = j \\ 0 & \text{otherwise} \end{cases} \ . \tag{5.18}$$

Figure 5.3(a) illustrates the relationship between the $2h \times 2h$ orthonormal transform E_m, and the block sparse, $p \times p$ orthonormal transform T_m. The four blocks of E_m are inserted in the appropriate block locations to form the larger, block sparse matrix T_m. The overall change in the log likelihood in (5.8) due to applying T_m to X_m and maximized with respect to $\hat{\Lambda}(T_m)$ is given by (see Appendix A)

$$\begin{aligned}\Delta \log p_{(T_m,\hat{\Lambda}(T_m))}(\mathrm{X}_m) &= \log p_{(T_m,\hat{\Lambda}(T_m))}(\mathrm{X}_m) - \log p_{(I,\hat{\Lambda}(I))}(\mathrm{X}_m) \\ &= -\frac{n}{2}\log\frac{|\mathrm{diag}(T_m^t S_m T_m)|}{|\mathrm{diag}(S_m)|} \\ &= -\frac{n}{2}\log\frac{|\mathrm{diag}(E_m^t S_m^{(i_m,j_m)} E_m)|}{|\mathrm{diag}(S_m^{(i_m,j_m)})|} \\ &= -\frac{n}{2}\log\left(1 - F_{i_m j_m}^2\right) \ ,\end{aligned} \tag{5.19}$$

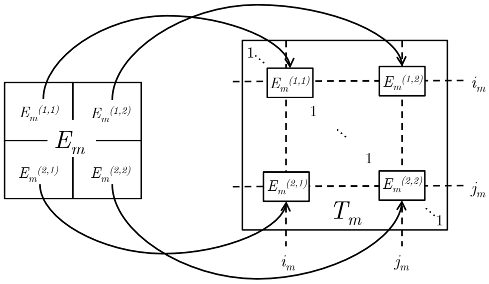
where we introduce the concept of a “correlation score”, F_{i_m,j_m}, defined by

$$F_{i_m,j_m} = \sqrt{1 - \frac{|\mathrm{diag}(E_m^t S_m^{(i_m,j_m)} E_m)|}{|\mathrm{diag}(S_m^{(i_m,j_m)})|}} \ .$$

In Appendix B, we show that the correlation score generalizes the concept of the correlation coefficient to pairs of random vectors and derive its main properties. The pair of sub-vectors with the largest value of $F_{i_m j_m}$ produces the largest increase in the log likelihood in (5.19). Therefore, we use the maximum value of $F_{i_m j_m}$ as the criterion for selecting the pair (i_m, j_m) during the design of $\hat{T}_m$ in (5.12). Finally, the algorithm in Figure 5.3(b) summarizes this greedy procedure to design the vector SMT.

5.3.3 The Vector SMT Design with Communication Energy Constraints

We extend the vector SMT design in Section 5.3.2 to account for the communication energy required for distributed decorrelation in a sensor network. When each T_m operates on $\mathrm{x}^{(i_m)}$ and $\mathrm{x}^{(j_m)}$ in a sensor network, it requires an amount, $\mathcal{E}(h, i_m, j_m)$ of energy for communication. In a scenario with a constrained energy budget, selecting sensors i_m and j_m based on the largest $F_{i_m j_m}$ can be prohibitive if these sensors are several hops apart in the network. Our approach to this problem is to perform a constrained optimization of (5.8) based on Lagrange multipliers. We augment the likelihood in (5.8) with a linear



(a)

```
//Initialization
forall $1 \leq i \leq L$ and $1 \leq j \leq L$ do
    $S^{(i,j)} \leftarrow \frac{1}{n}\begin{bmatrix} \mathrm{X}^{(i)} \\ \hline \mathrm{X}^{(j)} \end{bmatrix}\left[\mathrm{X}^{(i)t}|\mathrm{X}^{(j)t}\right]$
    $E \leftarrow ComputeScalarSMT(S^{(i,j)})$
    $F_{ij} \leftarrow \left(1 - \frac{|\mathrm{diag}(E^t S^{(i,j)} E)|}{|\mathrm{diag}(S^{(i,j)})|}\right)^{\frac{1}{2}}$
end
//Main Loop
for $m = 1, \cdots, M$ do
    $(i_m, j_m) \leftarrow \arg\max F_{ij}$
    $E_m \leftarrow ComputeScalarSMT(S^{(i_m,j_m)})$
    $T_m \leftarrow$
    $MapToPairwiseTransform(E_m, i_m, j_m)$
    Update matrix $F_{ij}$
    $S_m^{(i,j)} \leftarrow E_m^t S^{(i_m,j_m)} E_m$
end
```

(b)

Fig. 5.3. (a) Mapping from the $2h \times 2h$ matrix E, result of the optimization in (5.16), to the $p \times p$ block sparse matrix T_m associated with the (i_m, j_m) sub-vector pair. (b) The vector SMT design algorithm.

penalization term associated with the total communication energy required for distributed decorrelation. The augmented log likelihood is given by

$$\mathcal{L}_{(T,\Lambda)}(\mathrm{X}) = \log p_{(T,\Lambda)}(\mathrm{X}) - \mu \sum_{m=1}^{M} \mathcal{E}(h, i_m, j_m) \ . \tag{5.20}$$

The parameter μ has units of log likelihood/energy, and controls the weight given to the communication energy when maximizing the likelihood. When $\mu = 0$, the design becomes the unconstrained vector SMT design in Section 5.3.2. When we apply T_m to X_m and maximize (5.20) with respect to $\hat{\Lambda}(T_m)$, the overall change in the augmented likelihood is given by

$$\begin{aligned} \Delta\mathcal{L}_{(T_m,\hat{\Lambda}(T_m))}(\mathrm{X}_m) &= \mathcal{L}_{(T_m,\hat{\Lambda}(T_m))}(\mathrm{X}_m) - \mathcal{L}_{(I,\hat{\Lambda}(I))}(\mathrm{X}_m) \\ &= -\frac{n}{2}\log\left\{\frac{|\mathrm{diag}(T_m^t S_m T_m)|}{|\mathrm{diag}(S_m)|}\right\} - \mu\mathcal{E}(h, i_m, j_m) \\ &= -\frac{n}{2}\log\left(1 - F_{i_m j_m}^2\right) - \mu\mathcal{E}(h, i_m, j_m) \end{aligned} \tag{5.21}$$

Therefore, when designing $\hat{T}_m$ with energy constraints, we select the pair of sub-vectors (i_m, j_m) with the smallest value of $(1 - F_{i_m,j_m}^2)e^{2\mu\mathcal{E}(h,i_m,j_m)/n}$, i.e., the pair (i_m, j_m) that simultaneously maximizes the correlation coefficient, $F_{i_m j_m}$ and minimizes the communication energy penalty, $\mu\mathcal{E}(h, i_m, j_m)$ in order to increase the augmented log likelihood in (5.21) by the largest amount.

5.3.4 Model Order Identification

Let $\mathcal{M}_M$ be a vector SMT model with decorrelating transform $T = \prod_{m=1}^{M} T_m$. Here, we discuss three alternatives for selecting the model order parameter, M.

Fixed Maximum Energy

We select M such that the total energy required for the distributed decorrelation, $T^t\mathrm{x}$ does not exceed some fixed threshold $\mathcal{E}_0$, i.e., $\sum_{m=1}^{M} \mathcal{E}(h, i_m, j_m) \leq \mathcal{E}_0$. This threshold, $\mathcal{E}_0$ is fixed based on a pre-established maximum energy budget allowed for the distributed decorrelation.

Cross-Validation

We partition the $p \times n$ data sample matrix X into $\mathcal{K}$, $p \times n_k$ matrices $\mathrm{X}_{(k)}$, $\mathrm{X} = [\mathrm{X}_{(1)}|\cdots|\mathrm{X}_{(\mathcal{K})}]$, and define $\bar{\mathrm{X}}_{(k)}$ as a matrix containing the samples in X that are not in $\mathrm{X}_{(k)}$. For each $k = 1, \cdots, \mathcal{K}$, we design $\mathcal{M}_M$ from $\bar{\mathrm{X}}_{(k)}$, and compute its log likelihood over X_k, i.e., $\log p_{\mathcal{M}_M}(\mathrm{X}_{(k)}|\bar{\mathrm{X}}_{(k)})$. We select M so that it maximizes the average cross-validated log likelihood [7],

$$L(\mathcal{M}_M) = \frac{1}{\mathcal{K}} \sum_{i=1}^{\mathcal{K}} \log p_{\mathcal{M}_M}(\mathrm{X}_{(k)}|\bar{\mathrm{X}}_{(k)}) \ . \tag{5.22}$$

Minimum Description Length (MDL) Criterion

Based on the MDL principle [74–76], we select M such that the model $\mathcal{M}_M$ has the shortest encoding, among all models, of both its parameters and the sample matrix, X. The total description length of $\mathcal{M}_M$ in nats is given by

$$\ell_M = -\log p_{\mathcal{M}_M}(\mathrm{X}) + \frac{1}{2} MK \log(pn) + 2MK\log(2h) + 2M\log(L) \ , \tag{5.23}$$

where $-\log p_{\mathcal{M}_M}(\mathrm{X})$ nats are used to encode X, $\frac{1}{2}MK\log(pn)$ nats are used to encode the MK real-valued angles of the Givens rotations across all M pairwise transforms, $2MK\log(2h)$ nats are used for the MK rotation coordinate pairs, and finally, $2M\log(L)$ nats are used for the indices of sub-vector pairs of the M pairwise transforms. Our goal is then to select M such that it minimizes ℓ_M in (5.23). Initially, ℓ_M decreases with M because it is dominated by the likelihood term, $\log p_{\mathcal{M}_M}(\mathrm{X})$. However, when M is large, the other terms dominate ℓ_M causing it to increase as M increases. Therefore, we select M that minimizes ℓ_M by picking the first value of M such that

$$\begin{aligned} \ell_{M+1} - \ell_M &= -\log \frac{p_{\mathcal{M}_{M+1}}(\mathrm{X})}{p_{\mathcal{M}_M}(\mathrm{X})} + \frac{1}{2}K\log(pn) + 2K\log(2h) + 2\log(L) \\ &= -\frac{n}{2}\log(1 - F^2_{i_m,j_m}) + \frac{1}{2}K\log(pn) + 2K\log(2h) + 2\log(L) \geq 0 \ . \end{aligned}$$

This condition leads to this new stop condition for the main loop of the algorithm in Figure 5.3(b),

$$F^2_{i_m,j_m} \geq 1 - \exp\left\{ \frac{K\log(pn) + 4K\log(2h) + 4\log(L)}{n} \right\} \ . \tag{5.24}$$

It is easy to generalize ℓ_M in (5.23) to the case where each pairwise transform, T_m has a different number of Givens rotations, K_m, resulting in

$$\ell_M^{(general)} = -\log p_{\mathcal{M}_M}(\mathrm{X}) + \frac{1}{2}\sum_{m=1}^{M} K_m \log(pn) + 2\sum_{m=1}^{M} K_m \log(2h) + 2M\log(L) \ . \tag{5.25}$$

Finally, when $\ell_{M+1}^{(general)} - \ell_M^{(general)} \geq 0$ is satisfied, the new stop condition for the loop in Figure 5.3(b) is given by

$$F^2_{i_m,j_m} \geq 1 - \exp\left\{ \frac{K_{m+1}\log(pn) + 4K_{m+1}\log(2h) + 4\log(L)}{n} \right\} \ . \tag{5.26}$$

5.4 Anomaly Detection

We use the vector SMT to compute the covariance estimate, $\hat{R}$ of the p-dimensional vector, x for the purpose of performing anomaly detection using the Neyman-Pearson framework [35]. Here, we first formulate the anomaly detection problem, and then describe the ellipsoid volume measure of detection accuracy [52] used in the experimental section.

5.4.1 Problem Formulation

Let the p-dimensional vector x be an aggregated measurement from all L sensors in the network. We presume that x is typical (non-anomalous) if it is sampled from a multivariate Gaussian distribution, $\mathcal{N}(0, R)$ or anomalous if it is sampled from a uniform distribution $\mathcal{U}(\mathrm{x}) = c$, for some constant c [46, 77]. Formally, we have the following hypotheses,

$$\begin{aligned} \mathcal{H}_0 &: \quad \mathrm{x} \sim \mathcal{N}(0, R) \\ \mathcal{H}_1 &: \quad \mathrm{x} \sim \mathcal{U}, \end{aligned} \tag{5.27}$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ are the null and alternative hypotheses respectively. According to the Neyman-Pearson lemma [35], the optimal classifier has the form of the log likelihood ratio test,

$$\begin{aligned}\Gamma(\mathrm{x}) &= \log\left\{\frac{p(\mathrm{x};\mathcal{H}_1)}{p(\mathrm{x};\mathcal{H}_0)}\right\} = \log c - \log p(\mathrm{x};\mathcal{H}_0)\\ &= \log c + \frac{p}{2}\log 2\pi + \frac{1}{2}\log|R| + \frac{1}{2}\mathrm{x}^t R^{-1}\mathrm{x} \gtrless \Gamma_0\ . \end{aligned} \tag{5.28}$$

This likelihood ratio test maximizes the probability of detection, $p(\mathcal{H}_1;\mathcal{H}_1)$ for a fixed probability of false alarm, $p(\mathcal{H}_1;\mathcal{H}_0)$, which controlled by the threshold Γ_0. We incorporate all the constant terms into a new threshold, η^2, such that the test in (5.28) becomes

$$D_R(\mathrm{x}) = \mathrm{x}^t R^{-1}\mathrm{x} \gtrless \eta^2. \tag{5.29}$$

If we further assume that $R = T\Lambda T^t$, where T and Λ are orthonormal and diagonal matrices respectively, the test in (5.28) can be written as a weighted sum of p uncorrelated coordinates,

$$\widetilde{D}_\Lambda(\tilde{x}) = \sum_{i=1}^{p}\frac{\tilde{x}_i^2}{\lambda_i} \gtrless \eta^2 \tag{5.30}$$

where $\tilde{\mathrm{x}} = T^t\mathrm{x}$, and $\lambda_i \equiv [\Lambda]_{ii}$ $(1 \leq i \leq p)$. Finally, because the sum in (5.30) involves only independent terms, it can be evaluated distributedly across a sensor network while requiring minimum communication.

5.4.2 Ellipsoid Volume as a Measure of Detection Accuracy

The ellipsoid volume approach [36, 52, 53] measures anomaly detection accuracy without requiring labeled anomalous samples. Because anomalies are rare and loosely defined events, we often lack enough test samples labeled as anomalous to estimate the probability of detection, $p(\mathcal{H}_1;\mathcal{H}_0)$ required for ROC analysis [35]. Instead of relying on anomalous samples, the ellipsoid volume approach seeks to measure detection accuracy by characterizing how well a covariance estimate, $\hat{R}$ models the typical data samples. It evaluates the volume of the region within the ellipsoid, $\mathrm{x}^t\hat{R}^{-1}\mathrm{x} \leq \eta^2$ for a certain probability of false alarm controlled by η. Such a volume is evaluated by

$$V(\hat{R},\eta) = \frac{\pi^{p/2}}{\Gamma(1+p/2)}\eta^p\sqrt{|\hat{R}|}\ . \tag{5.31}$$

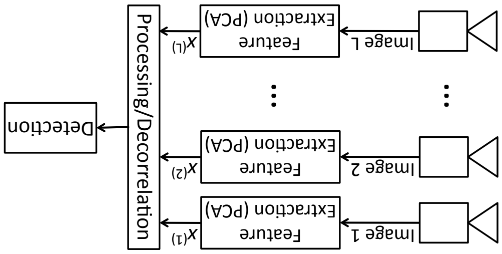
We use $V(\hat{R},\eta)$ as a proxy for the probability of missed detection. Smaller values of $V(\hat{R},\eta)$ indicate smaller chances of an anomalous sample lying within this ellipsoid, and therefore being wrongly classified as typical. Therefore, for a fixed probability of false alarm, smaller values of $V(\hat{R},\eta)$ indicate higher detection accuracy.

5.5 Experimental Results

We provide experimental results using simulated and real data to quantify the effectiveness of our proposed method. In all experiments, we assume communications occur between sensors connected in a hierarchical network with binary tree topology, and that communication of one scalar value between adjacent sensors uses one unit of energy. We compare the vector SMT decorrelation with two other approaches for processing the sensor outputs, a centralized and an independent one. In the centralized approach, all sensors communicate their h-dimensional vector outputs to the root of the tree. This approach is very communication intensive, but once all the data is centrally located, any decorrelation algorithm can be used to decorrelate x. We choose the scalar SMT algorithm because it has been shown to provide accurate decorrelation from limited training data since it approximates the maximum likelihood estimate. In the independent approach, each sensor computes a partial likelihood of its output independently and communicates it to the root of the tree. The root sensor adds the partial likelihoods from all sensors and makes a detection decision without decorrelating the sensor outputs. This requires the least communication among all approaches compared. Figure 5.4(a) summarizes these approaches in terms of their main computation and communication characteristics. Finally, Figure 5.4(b) shows the event detection simulation steps by a camera network in several of our experiments. Each camera sensor records an image and encodes its h-dimensional vector output using

Processing/Decorrelation Methods

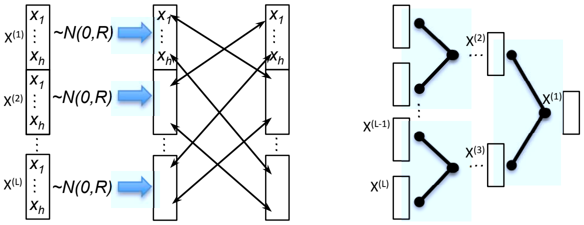
Method	Algorithm	Communication	Decorrelation
Vector SMT (distributed)	Vector SMT	Between pairs of nodes / caching	sub-vector pairs in network
Centralized	Scalar SMT	Vector outputs to centralized node	coordinate pairs at single node
Independent	None	Partial likelihoods to centralized node	–



(b)

(a)

Fig. 5.4. The experimental setup: (a) Summary of the several approaches to sensor output decorrelation compared and their main properties. (b) Steps for decorrelation and anomaly detection used in our experimental results. Each sensor encodes its output as an h-dimensional vector using PCA. Experiments with artificial data replace the sensor vector outputs with artificially generated random vector data. The outputs are processed in the network before a detection decision is made.



(a) (b)

Fig. 5.5. Generation of a data sample, x aggregated from correlated h-dimensional sensor outputs $\mathrm{x}^{(i)}$, $i = 1, \cdots, L$. (a) First we draw each $\mathrm{x}^{(i)}$ independently from the $\mathcal{N}(0, R)$ distribution, with $[R]_{rs} = \rho^{|r-s|}$. Then, we permute individual coordinates of x across all $x^{(i)}$, $i = 1, \cdots, L$ to spread correlations among all sensor outputs. (b) Each $x^{(i)}$ is the output of a sensor i connected to other sensors in a hierarchical network with binary tree topology.

principal component analysis (PCA). We process the outputs using one of the approaches in Figure 5.4(a) before making a detection decision.

5.5.1 Simulation experiments using artificial model data

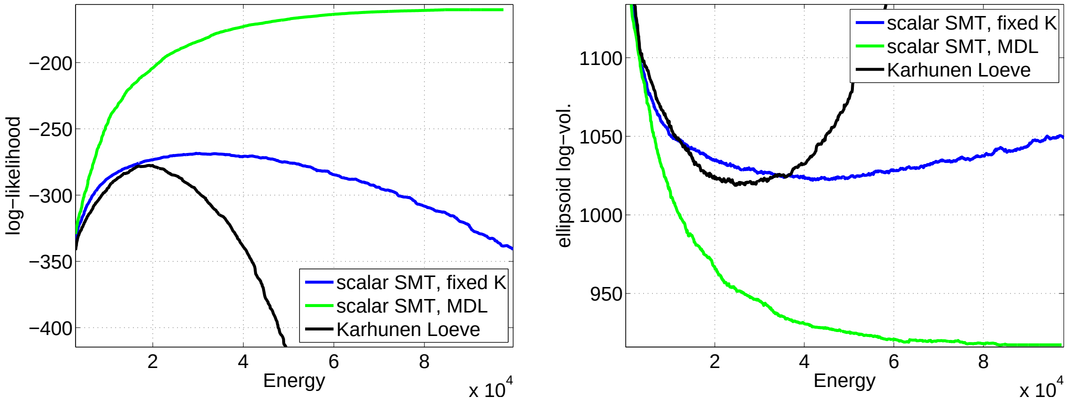
In these experiments, we study how the vector SMT model accuracy changes with (i) different choices of decorrelating transforms used as the pairwise transform between two sensor outputs, and (ii) different values of the energy constraint parameter, μ used in the constrained design in Section 5.3.3. We simulate a network with $L = 31$ sensors, where each sensor i outputs a vector, $\mathrm{x}^{(i)}$ with $h = 25$ dimensions. These sensor vector outputs are correlated. Figure 5.5 shows how we generate a data sample x, aggregated from correlated sensor outputs $\mathrm{x}^{(i)}$, $i = 1, \cdots, 31$. First, we draw each $\mathrm{x}^{(i)}$ independently, from the $\mathcal{N}(0, R)$ distribution, with the $h \times h$ covariance matrix, $[R]_{rs} = \rho^{|r-s|}$, where $\rho = 0.7$. Then we permute individual coordinates of x across all $\mathrm{x}^{(i)}$, $i = 1, \cdots, 31$, to spread correlations among all sensor outputs. Finally, each $x^{(i)}$ is the output of a sensor i interconnected in a hierarchical network with binary tree topology.

Figure 5.6 shows the vector SMT model accuracy *vs.* communication energy required for decorrelation for three different choices of pairwise transforms: scalar SMT with fixed number of Givens rotations, scalar SMT with MDL criterion, and PCA (eigenvector matrix from the exact diagonalization of the pairwise sample covariance). We measure accuracy by the average log-likelihood of the vector SMT model over $n = 300$ testing samples (Figure 5.6(a)), and the ellipsoid log-volume covering 99% of the testing samples, i.e., for 1% false alarm rate (Figure 5.6(b)). In general the model accuracy improves to an optimal level and then starts to decrease as more energy is spent with pairwise transforms. This decrease in accuracy happens because vector SMT models with a large number of pairwise transforms tend to overfit the training data. For scalar SMT-MDL pairwise transforms, the MDL criterion adjusts the number of Givens rotations for each new pairwise transform according to an estimate of the correlation still present in the data [45], helping to prevent overfitting. Because it is overall the most accurate, the scalar SMT-MDL is our pairwise transform of choice during all other experiments in this chapter.

Figure 5.7 shows model accuracy *vs.* communication energy for three choices of the energy constraint parameter μ. The accuracy is measured by average model log-likelihood (Figure 5.7(a)) and ellipsoid log-volume covering 99% of the testing samples (Figure 5.7(b)). The parameter μ selects the trade-off between model accuracy and energy consumption. For a small fixed energy value, the vector SMT with largest μ value produces the most accurate model. For large values of energy, the constrained vector SMT accuracy tends to level out at sub-optimal values while the unconstrained vector SMT has the highest accuracy.

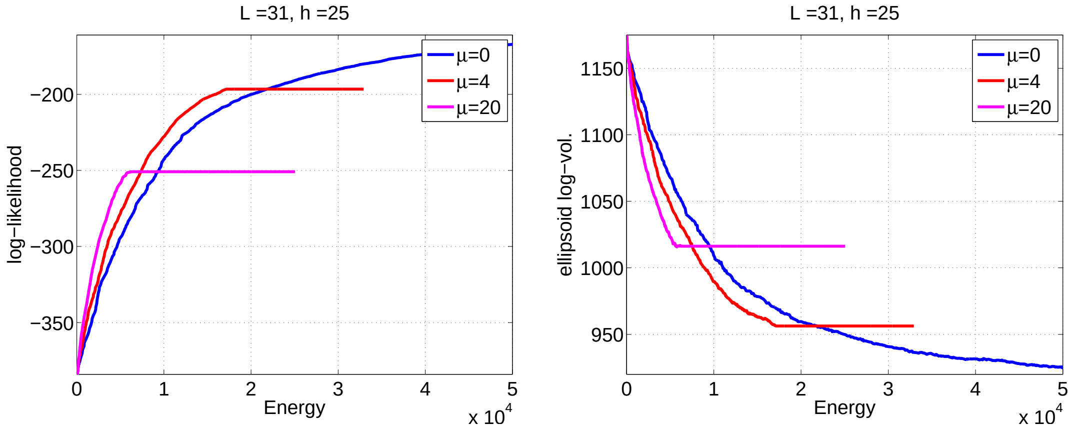
5.5.2 Simulation experiments using artificial moving sphere images

In this experiment, we apply the vector SMT to decorrelate two simultaneous camera views for anomaly detection. We generate artificial images of a 3D sphere placed at random positions along two straight diagonal lines over a plane, as illustrated in Figures 5.8(a) and (b). We refer to sphere positions along the line in Figure 5.8(a) as typical ones, while



(a) (b)

Fig. 5.6. Vector SMT model accuracy *vs.* communication energy consumption using 100 training data samples from AR(1) model. Comparison of different vector SMT pairwise transforms for a range of communication energies using 100 training data samples from AR(1) model: (a) average log-likelihood over 300 test samples; (b) ellipsoid log-volume covering 99% of the test samples (1% false alarm rate). The choice of scalar SMT MDL produces the best increase in accuracy, measured by both metrics.



(a) (b)

Fig. 5.7. Comparison of vector SMT energy constraint parameter values for a range of communication energies using 100 training data samples from AR(1) model: (a) average log-likelihood over 300 test samples; (b) ellipsoid log-volume covering 99% of the test samples (1% false alarm rate). Vector SMT models with larger μ are the most accurate for fixed small energy values. For large energy values, the constrained models tend to exhibit sub-optimal accuracies compared to the unconstrained vector SMT.

referring to positions along the mirrored diagonal line in Figure 5.8(b) as anomalous ones. Two cameras ($L = 2$) monitor the sphere locations in the 3D region. Figure 5.8(c) shows the top (X-Y) view captured by camera 1, while Figure 5.8(d) shows the side (X-Z) view captured by camera 2. Note that it is impossible to tell anomalous from typical sphere positions by looking at the views in Figures 5.8(c) and (d) separately. Instead, one needs to process both views together to extract useful discriminant information. Each camera outputs a vector of $h = 10$ dimensions with its largest PCA components. The joint output from both cameras form a sample. We use 100 typical samples to train the detectors using vector SMT decorrelation and independent processing of the views. During testing, we use 200 samples, disjoint from the training set, with 100 typical, and another 100 anomalous samples.

Figures 5.8(e) and (f) compare the detection accuracy using both independent processing and vector SMT to decorrelate the joint camera outputs. Both the ROC analysis (Figure 5.8(e)) and ellipsoid log-volume coverage plot (Figure 5.8(f)) suggest that when the two views are processed independently, the detector cannot distinguish anomalous from typical samples. However, when the vector SMT decorrelates both views, the anomaly detection is very accurate.

Figure 5.9 shows sets with five eigen-images associated with the largest eigenvalues for both the independent (Figure 5.9(a)) and the vector SMT (Figure 5.9(b)) processing approaches. In the independent processing case, each eigen-image is associated with a single camera view. On the other hand, the vector SMT processing produces eigen-images, each modeling both camera views jointly.

5.5.3 Simulation experiments using artificial 3D sphere cloud images

In this experiment, we monitor clouds of spheres using twelve simultaneous camera views for the purpose of anomaly detection. We artificially generate sphere clouds randomly positioned in the 3D space, each containing 30 spheres. There are two types of clouds according to the sphere position distribution: (i) typical: the sphere positions are

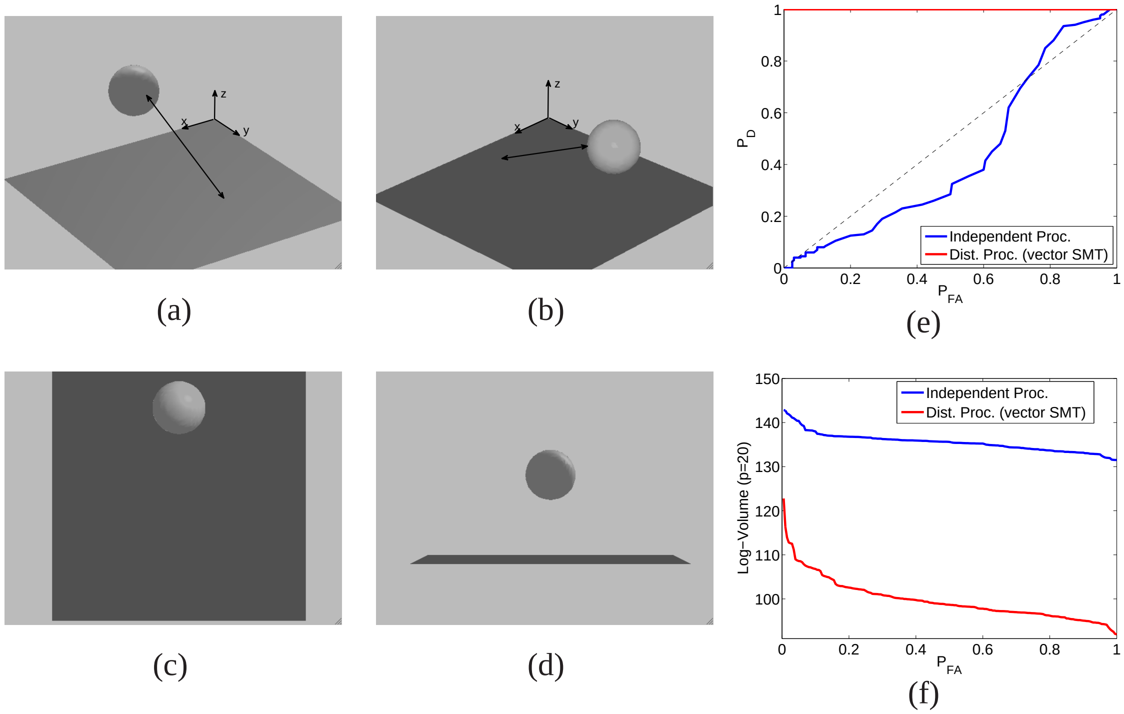
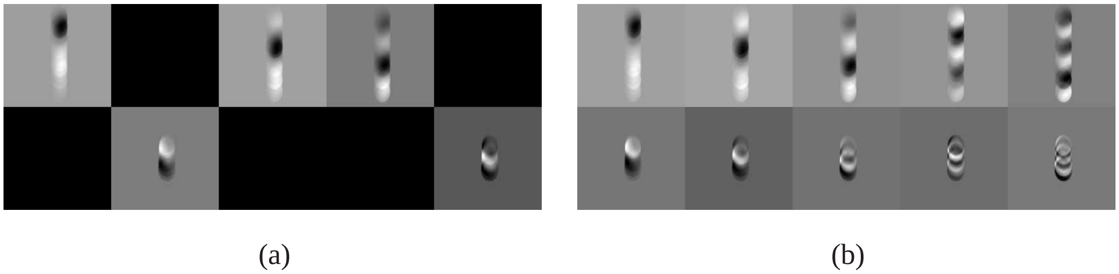


Fig. 5.8. Simulated 3D space with bouncing sphere: the sphere takes random positions along the line indicated by the double arrow (a) typical behavior; (b) anomalous behavior. The camera views: (c) top (X-Y dimensions); (d) side (X-Z dimensions). The detection accuracies using independent processing and vector SMT joint processing: (e) ROC curve; (f) "coverage plot" with log-volume of ellipsoid *vs.* probability of false alarm. Because there are only two camera views, centralized and vector SMT processing methods are equivalent.



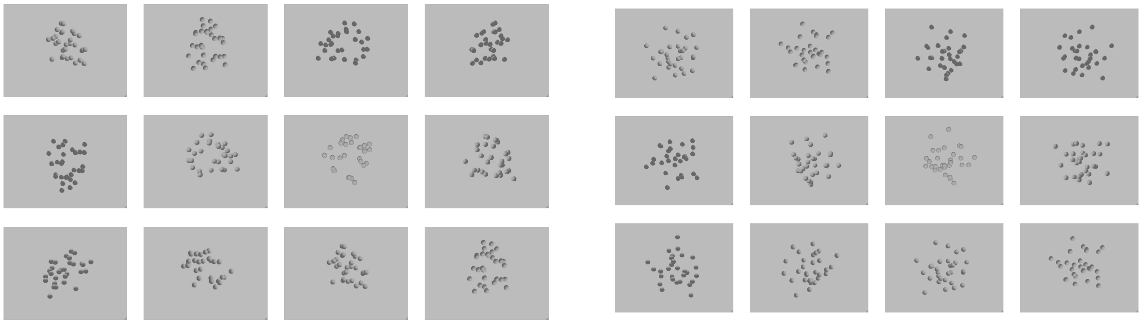
(a) (b)

Fig. 5.9. Eigen-image pairs of the moving sphere experiment sorted according to their corresponding eigenvalues in decreasing order (left-to-right): (a) when the camera views are processed independently, each eigenvector models a single view; (b) when the camera views are processed jointly using the vector SMT, each eigenvector models both views together.

generated from the $\mathcal{N}(0, I_{3\times 3})$ distribution, but only positions with distance from the origin exceeding a fixed threshold are selected, so that the resulting cloud is hollow; and (ii) anomalous: the random positions for the spheres are drawn from the $\mathcal{N}(0, I_{3\times 3})$ distribution without further selection so that the resulting cloud is dense. We monitor the same 3D cloud using $L = 12$ different cameras from different viewpoints, and each camera encodes its output using PCA to a vector of $h = 10$ dimensions. Figure 5.10 shows the twelve camera views for both a typical cloud sample (Figure 5.10(a)), and for an anomalous one (Figure 5.10(b)). Each data sample is formed by aggregating the twelve camera outputs. We generate 100 typical samples to train the detectors, and another 200 test samples, with 100 typical, and 100 anomalous.

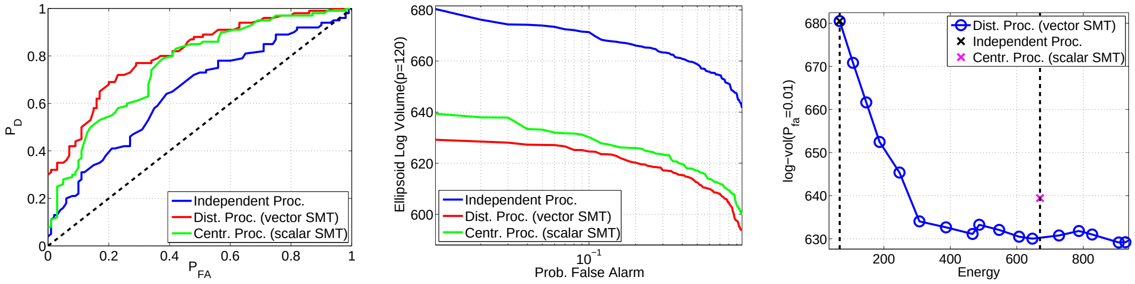
Figure 5.11 shows anomaly detection accuracy based on ROC analysis (Figure 5.11(a)), and log-volume of ellipsoid (Figure 5.11(b)). Among all methods compared, detection using independent processing is the least accurate, while both the centralized processing using scalar SMT and the distributed processing using vector SMT lead to high detection accuracies. Intuitively, as the views in Figure 5.10 suggest, it is difficult to distinguish between typical and anomalous samples by processing each view independently. Instead, the information that helps distinguishing an anomalous cloud from the typical ones is contained in the joint view of the camera images.

Figure 5.11(c) shows the ellipsoid log-volume for 1% false alarm rate *vs.* the communication energy for the different approaches compared. Independent processing is the least accurate while requiring the minimum energy among all approaches. The centralized approach very accurate, but it requires significant communication energy. In the vector SMT decorrelation, each pairwise decorrelation increases the detection accuracy while consuming more energy. There is a trade-off between detection accuracy and energy consumption, and one can choose the number of pairwise transforms to apply based on the desired accuracy and available energy budget. Finally, detection is more accurate when using vector SMT decorrelation compared to the scalar SMT one for the same energy consumption. This difference in accuracy is due to the inherent constraint of the vector SMT decorrelat-



(a) (b)

Fig. 5.10. The twelve camera views of a 3D sphere cloud sample: (a) a typical sample (hollow cloud); (b) an anomalous sample (dense cloud). The images suggest the difficulty to discriminate anomalous from typical samples by looking at each view independently. Instead, the discriminant information is contained in the joint camera views.



(a) (b) (c)

Fig. 5.11. Anomaly detection accuracy using the sphere cloud data: (a) ROC analysis; (b) log-volume of ellipsoid *vs.* probability of false alarm. Vector SMT decorrelation yields to the most accurate detection results for all false alarm rates. (c) log-volume of ellipsoid for 1% false alarm rate, i.e., 99% coverage *vs.* communication energy.

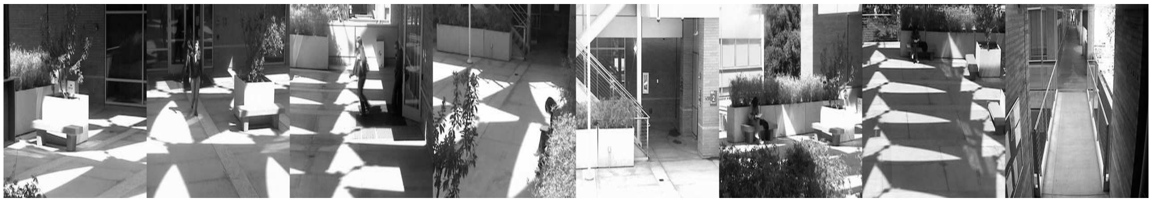


Fig. 5.12. The courtyard dataset from the UCR Videoweb Activities Dataset [78]: eight cameras, with ids 1 to 8 from left to right, monitor a courtyard from different viewpoints. Several activities in the courtyard are captured simultaneously by several cameras.

ing pairs of vectors, which tends to produce better models of a distribution when a limited number of training samples is available.

5.5.4 Simulation experiments using real multi-camera images

Figure 5.12 shows $L = 8$ camera views of a courtyard, constructed from video sequences from the UCR Videoweb Activities Dataset [78]. Each camera records a video sequence of approximately 4.2 min, with 30 frames/sec, generating a total of 7600 frames. The sequences are synchronized, so that multiple cameras capture events simultaneously. We subsample 1 in 3 frames from the 7600-frame sequence, and use 800 of the selected samples to to compute the encoding PCA transforms for each camera view. The final courtyard dataset has 1734 samples of $p = 160$ dimensions, with each view encoded in a sub-vector of $h = 20$ dimensions.

Table 5.1 shows correlation score values for all view pairs. Pairs of highly correlated views, capturing mostly the same events (as with cameras 1 and 6), receive higher score values than weakly correlated view pairs. The events captured by camera 8 are unrelated, and therefore uncorrelated, to the events captured by the other cameras, resulting in small correlation score values.

Figure 5.13 shows the eigen-images associated with the four largest eigenvalues for both the independent and vector SMT approaches. In the independent processing case

Table 5.1
Correlation score values for all pairs of views in the courtyard dataset. The correlation score measures the correlation of camera outputs between pairs of camera views. Pairs of cameras capturing the same events simultaneously have the highest correlation scores.

	1	2	3	4	5	6	7	8
1	1.0000	0.7206	0.5941	0.6637	0.6103	0.7373	0.7246	0.0000
2	-	1.0000	0.5999	0.6646	0.5897	0.7032	0.7559	0.0000
3	-	-	1.0000	0.6054	0.4909	0.5855	0.6216	0.0000
4	-	-	-	1.0000	0.5697	0.6601	0.6837	0.0000
5	-	-	-	-	1.0000	0.5982	0.6028	0.0000
6	-	-	-	-	-	1.0000	0.7237	0.0000
7	-	-	-	-	-	-	1.0000	0.0000
8	-	-	-	-	-	-	-	1.0000

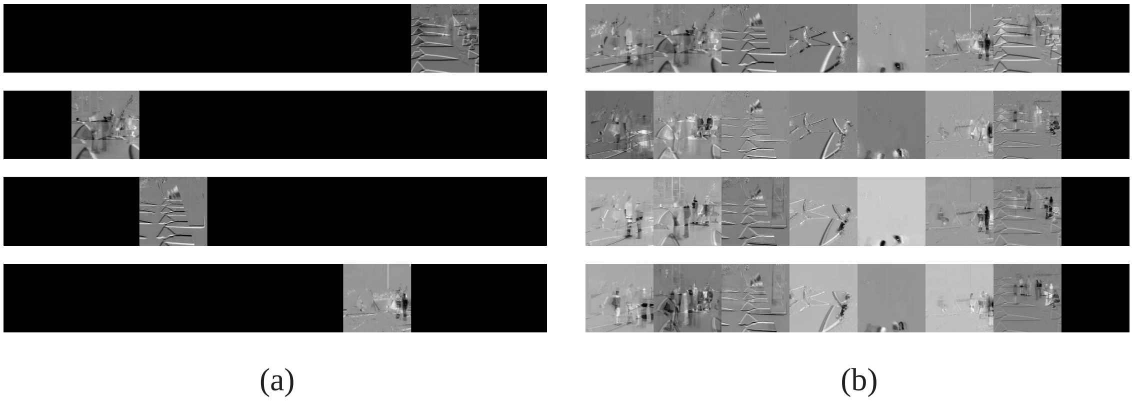
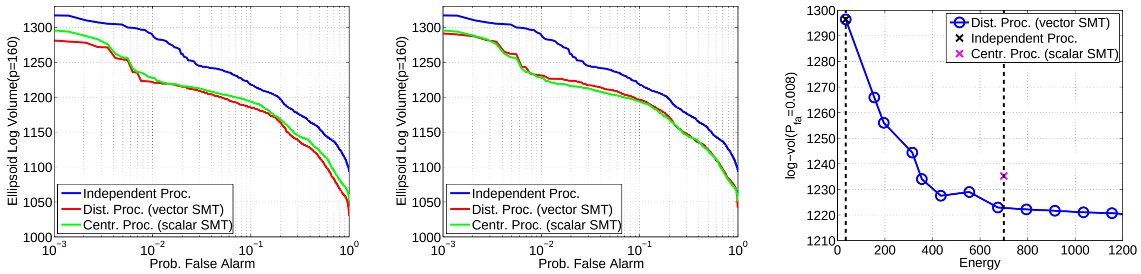


Fig. 5.13. Eigen-images from camera views of the courtyard dataset. (a) independent processing of camera views: each eigen-image corresponds to a single view and does not contain correlation information among multiple views; (b) joint processing modeled by the vector SMT: each eigen-image contains joint information of all correlated views.

(Figure 5.13(a)), each eigen-image corresponds to a single camera view, containing no information regarding the relationship between different views. On the other hand, the vector SMT eigen-images (Figure 5.13(b)) contain joint information of the correlated views. Since camera view 8 is not correlated with any other view, it does not appear together with others in the same eigen-image.

Figure 5.14 compares the accuracy of all approaches measured by the log-volume of the ellipsoid covering test samples. We split the samples into a training set, with 300 samples, and a test set, with 1434 samples. Figure 5.14(a) shows the ellipsoid log-volume computed for all false alarm rates. The vector SMT is the most accurate approach, with its volumes being the smallest across all false alarm rates. The vector SMT volumes are also smaller than the scalar SMT volumes. As discussed in Section 5.5.3, the vector SMT is more accurate than the scalar SMT because of the nature of its constrained decorrelating transform when trained with a small training set. Figure 5.14(b) shows results of the same experiment as in Figure 5.14(a) with the vector SMT model order selected so that the distributed decorrelation consumes only 50% of the energy required for the centralized approach. Figure 5.14(c) shows the ellipsoid log-volume for a fixed false alarm rate

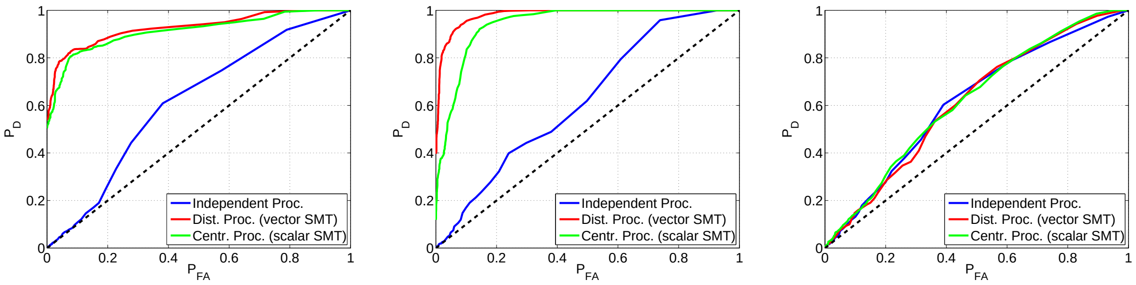


(a) (b) (c)

Fig. 5.14. Detection accuracy measured by the ellipsoid log-volume for the courtyard data set. Coverage plots showing the log-volume *vs.* probability of false alarm: (a) model order, $M = 7$, matching the energy of centralized processing, (b) model order, $M = 4$, matching 50% of the energy consumed for the centralized processing; (c) log-volume *vs.* communication energy for fixed probability of false alarm, $P_{FA} = 0.008$. When the communication energy is equal to the level required to execute the scalar SMT at a centralized node, the vector SMT has better detection accuracy. When the energy level is 50% of the level required by the centralized approach, the vector SMT has similar accuracy.

(0.8%) *vs.* communication energy. We observe the same trends observed in the sphere cloud experiment in Section 5.5.3. The independent approach has low accuracy while requiring low communication energy. The centralized decorrelation is highly accurate, but it requires large amounts of communication energy. The vector SMT increases the detection accuracy after each pairwise transform. Finally, the vector SMT approach has similar accuracy to the centralized approach for all false alarm rates while requiring significantly less communication energy.

Figure 5.15 shows ROC curves for detection of anomalous samples generated by 4-fold increase in the largest component of the vector output of a single camera view. We use 200 typical samples to learn the decorrelating transform, and using the remaining samples for testing. Figures 5.15(a), (b), and (c) show the ROC curves for the cases with the anomaly generated in camera views 2, 6, and 8 respectively. Because views 2 and 6 are correlated with other views (see Table 5.1), detection of anomalies in these views is accurate when



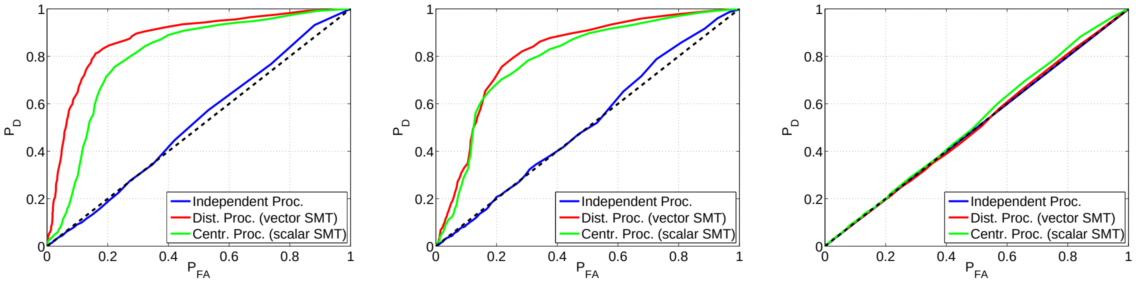
(a) (b) (c)

Fig. 5.15. Detection accuracy of anomalies artificially generated by a 4-fold increase of the largest eigenvalue of a single view: (a) view 2; (b) view 6; (c) view 8. The detection accuracy increases with decorrelation when the anomalies are in camera views that are highly correlated with other views. When the anomaly is inserted in a uncorrelated view, decorrelation methods do not increase the detection accuracy.

we decorrelate the views using the vector and scalar SMT approaches, and very inaccurate when we process the views independently. Because view 8 is uncorrelated with other views, decorrelation does not help improve detection accuracy and all approaches are inaccurate.

Figure 5.16 shows the ROC curves for detection of what we call the “Ocean’s Eleven” anomaly. This anomaly is generated by swapping images of a single views between two samples captured at different instants. We refer to it as the Ocean’s Eleven anomaly because of the resemblance with the anomaly created to trick the surveillance cameras during the casino robbery in the Ocean’s Eleven film [79]. Figures 5.16(a), (b), and (c) show the ROC curves for detection of anomalies in views 2, 6, and 8 respectively. Because views 2 and 6 are correlated with other views, detection is accurate when we decorrelate the views with scalar and vector SMTs, and very inaccurate when we process the views independently. Because view 8 is uncorrelated with the other views, decorrelation does not help improve detection accuracy and all approaches are inaccurate.

Figure 5.17 shows the typical and anomalous samples used in an experiment to detect suspicious (anomalous) human activity captured simultaneously by multiple cameras. We select 200 samples where a group of people coalesce at the center of the courtyard and



(a) (b) (c)

Fig. 5.16. ROC analysis of the Ocean's Eleven anomaly, generated by swapping images of a single camera view between samples: (a) camera view 2, and (b) camera view 6, which are highly correlated with other views; (c) camera view 8, which is uncorrelated with other views.

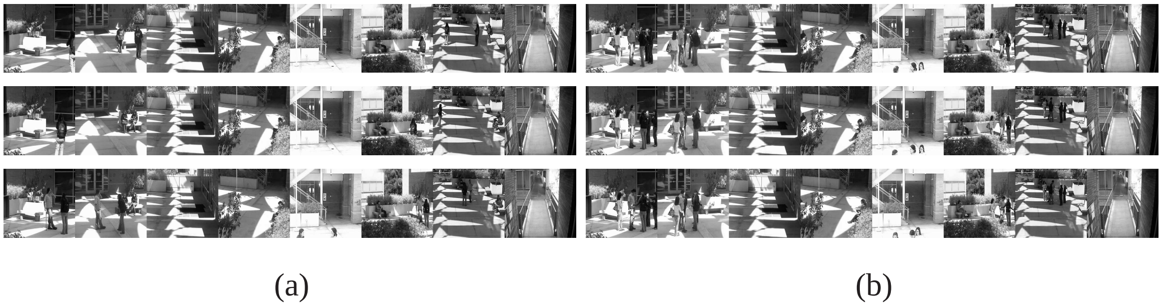


Fig. 5.17. Samples used in the experiment detecting people coalescing in the middle of the courtyard: (a) Typical samples; (b) Anomalous samples, with images of people coalescing.

label them as anomalous, while selecting another 200 samples where the group do not coalesce and label them as typical. We use another 300 typical samples to train the vector SMT. Figure 5.18 shows the ROC curves for detection of people coalescing in the middle of the courtyard. The vector SMT decorrelation in this experiment consumes 60% of the communication energy required for the scalar SMT. Detection is very accurate when using vector and scalar SMTs for view decorrelation, and inaccurate when processing the views independently, specially for low probabilities of false alarm. Similarly to the detection of dense clouds (see Section 5.5.3), it is difficult to detect people coalescing when processing camera views independently. Instead, one needs to to consider the views jointly for good detection accuracy.

5.6 Conclusions

We have proposed a novel method for decorrelation of vector measurements distributed across sensor networks. The new method is based on the constrained maximum likelihood estimation of the joint covariance of the measurements. It generalizes the concept of the previously proposed sparse matrix transform to the decorrelation of vectors. We have demonstrated the effectiveness of the new approach using both artificial and real data sets. In addition to providing accurate decorrelating transforms and enabling accurate anomaly

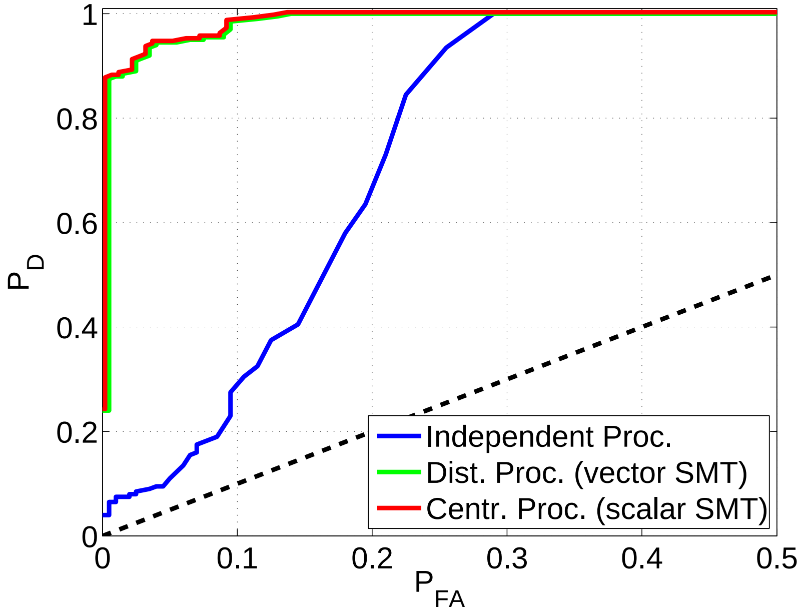


Fig. 5.18. ROC analysis comparing the detection accuracy when detecting people coalescing in the middle of the courtyard. Detection using vector and scalar SMTs are highly accurate for small probabilities of false alarm. In this experiment the vector SMT consumes approximately 60% of the communication energy required for the scalar SMT.

detection, our method offers advantages in terms operating distributedly, under communication energy constraints. In future work, we plan to provide a distributed algorithm to design the decorrelating transform in network.

LIST OF REFERENCES

LIST OF REFERENCES

[1] I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, pp. 102–114, Aug. 2002.

[2] S. Soro and W. Heinzelman, “A Survey of Visual Sensor Networks,” *Advances in Multimedia*, vol. 2009, pp. 1–22, 2009.

[3] A. K. R. Chowdhury and B. Song, *Camera Networks: The Acquisition and Analysis of Videos over Wide Areas*. Synthesis Lectures on Computer Vision, Morgan & Claypool Publishers, 2012.

[4] J. B. Predd, S. R. Kulkarni, and H. V. Poor, “Distributed learning in wireless sensor networks,” *IEEE Signal Processing Magazine*, vol. 23, pp. 56–69, Jul. 2006.

[5] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Boston, MA: Academic Press, 1990. 2nd Ed.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2nd ed., Nov. 2000.

[7] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, New York, NY, USA: Springer New York Inc., 2 ed., 2009.

[8] A. K. Jain, R. P. Duin, and J. Mao, “Statistical pattern recognition: A review,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[9] D. L. Donoho, “High-dimensional data analysis: The curses and blessings of dimensionality,” in *Math Challenges of the 21st Century*, (Los Angeles, CA), American Mathematical Society, Aug. 2000.

[10] J. Schafer and K. Strimmer, “A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics,” *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, 2005.

[11] M. J. Daniels and R. E. Kass, “Shrinkage estimators for covariance matrices,” *Biometrics*, vol. 57, no. 4, pp. 1173–1184, 2001.

[12] O. Ledoit and M. Wolf, “Improved estimation of the covariance matrix of stock returns with an application to portfolio selection,” *Journal of Empirical Finance*, vol. 10, Dec. 2003.

[13] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. Hoboken, NJ: John Wiley & Sons, 2003.

[14] O. Ledoit and M. Wolf, “A well-conditioned estimator for large dimensional covariance matrices,” *Journal of Multivariate Analysis*, vol. 88, pp. 365–411, Feb. 2004.

[15] N. M. Nasrabadi, "Regularization for spectral matched filter and rx anomaly detector," in *Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery XIV* (S. S. Shen and P. Lewis, eds.), 2008.

[16] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, 1989.

[17] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, pp. 432–441, Jul. 2008.

[18] C. Chennubhotla and A. Jepson, "Sparse PCA: Extracting multi-scale structure from data," *Proc. 8th IEEE Int. Conf. Computer Vision, 2001 (ICCV 2001)*.

[19] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[20] A. Rothman, P. Bickel, E. Levina, and J. Zhu, "Sparse permutation invariant covariance estimation," *Electronic Journal of Statistics*, vol. 2, pp. 494–515, 2008.

[21] P. J. Bickel and E. Levina, "Covariance regularization by thresholding," Technical Report 744, Department of Statistics, UC Berkeley, 2007.

[22] P. J. Bickel and E. Levina, "Regularized estimation of large covariance matrices," *Annals of Statistics*, vol. 36, no. 1, pp. 199–227, 2008.

[23] G. Cao and C. A. Bouman, "Covariance estimation for high dimensional data vectors using the sparse matrix transform," in *Adv. Neural Information Processing Systems (NIPS)*, (Vancouver, BC, Canada), MIT Press, Dec. 2008.

[24] L. R. Bachega, G. Cao, and C. A. Bouman, "Fast signal analysis and decomposition on graphs using the sparse matrix transform," in *Proc. Int. Conf. Accustics, Speech and Signal Processing*, (Dallas, TX), Mar. 2010.

[25] W. Givens, "Computation of plane unitary rotations transforming a general matrix to triangular form," *Journal of the Society for Industrial and Applied Mathematics*, vol. 6, pp. 26–50, Mar. 1958.

[26] G. Cao, C. A. Bouman, and J. Theiler, "Weak signal detection in hyperspectral-imagery using sparse matrix transformation (SMT) covariance estimation," in *First Workshop on Hyperspectral Image and Signal Processing*, 2009.

[27] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ: John Wiley & Sons Inc., 2006.

[28] S. Watanabe, "Information theoretical analysis of multivariate correlation," *IBM J. Research and Development*, pp. 66–82, Jan. 1960.

[29] L. R. Bachega, C. A. Bouman, and J. Theiler, "Hypothesis testing in high-dimensional spase with the sparse matrix transform," in *The 6th IEEE Sensor Array and Multichannel Signal Processing Workshop*, (Israel), IEEE, Oct. 2010.

[30] L. R. Bachega, , and C. A. Bouman, "Classification of high-dimensional data using the sparse matrix transform," in *Proc. Int. Conf. Image Processing*, (Hong Kong, China), Sep.

[31] J. Theiler, “Subpixel anomalous change detection in remote sensing imagery,” (Santa Fe, NM), pp. 165–168, 2008 IEEE Southwest Symposium on Image Analysis and Interpretation, Mar. 2008.

[32] B. Moghaddam, T. Jebara, and A. Pentland, “Bayesian face recognition,” *Pattern Recognition*, vol. 33, pp. 1771–1782, 2000.

[33] A. B. Lee, B. Nadler, and L. Wasserman, “Treelets – an adaptive multi-scale basis for sparse unordered data,” *Annals of Applied Statistics*, vol. 2, no. 2, pp. 435–471, 2008.

[34] “The orl face database.” `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`.

[35] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol.2: Detection Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1998.

[36] J. Theiler, G. Cao, L. Bachega, and C. Bouman, “Sparse matrix transform for hyperspectral image processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, pp. 424–437, Jun. 2011.

[37] J. Theiler and S. Perkins, “Proposed framework for anomalous change detection,” *ICML Workshop on Machine Learning Algorithms for Surveillance and Event Detection*, pp. 7–14, 2006.

[38] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, pp. 711–720, 1997.

[39] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, and J. Weng, “Discriminant analysis of principal components for face recognition,” in *Proc. 3rd. IEEE Int. Conf. Automatic Face and Gesture Recognition*, pp. 336–341, Apr.

[40] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss, “The FERET evaluation methodology for face-recognition algorithms,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000.

[41] D. W. J. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, “Anomaly detection from hyperspectral imagery,” *IEEE Signal Processing Magazine*, vol. 19, pp. 58–69, Jan. 2002.

[42] S. Matteoli, M. Diani, and G. Corsini, “A tutorial overview of anomaly detection in hyperspectral images,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, pp. 5–28, Jul. 2010.

[43] I. S. Reed, J. D. Mallett, and L. E. Brennan, “Rapid convergence rate in adaptive arrays,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 10, pp. 853–863, 1974.

[44] A. Ben-David and C. E. Davidson, “Estimation of hyperspectral covariance matrices,” in *2011 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 4324–4327, Jul. 2011.

[45] G. Cao, L. Bachega, and C. Bouman, “The sparse matrix transform for covariance estimation and analysis of high dimensional signals,” *IEEE Trans. Image Processing*, vol. 20, pp. 625–640, Mar. 2011.

[46] I. Steinwart, D. Hush, and C. Scovel, “A classification framework for anomaly detection,” *Journal of Machine Learning Research*, vol. 6, pp. 211–232, 2005.

[47] J. Y. Chen and I. Reed, “A detection algorithm for optical targets in clutter,” *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-23, pp. 46–59, Jan. 1987.

[48] I. S. Reed and X. Yu, “Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution,” *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 38, pp. 1760–1770, 1990.

[49] J. P. Hoffbeck and D. A. Landgrebe, “Covariance matrix estimation and classification with limited training data,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 763–767, 1996.

[50] J. Theiler, “The incredible shrinking covariance estimator,” in *Proc. SPIE, Automatic Target Recognition XXII*, p. 83910P, May 2012.

[51] C. E. Caefer, J. Silverman, O. Orthal, D. Antonelli, Y. Sharoni, and S. R. Rotman, “Improved covariance matrices for point target detection in hyperspectral data,” *Optical Engineering*, vol. 7, p. 076402, 2008.

[52] J. Theiler and D. R. Hush, “Statistics for characterizing data on the periphery,” *Proc. IEEE Int. Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 4764–4767, Jul. 2010.

[53] J. Theiler, “Ellipsoid-simplex hybrid for hyperspectral anomaly detection,” *Proc. IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pp. 1–4, Jun. 2011.

[54] D. Snyder, J. Kerekes, I. Fairweather, R. Crabtree, J. Shive, and S. Hager, “Development of a web-based application to evaluate target finding algorithms,” *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, vol. 2, pp. 915–918, 2008.

[55] J.-F. Chamberland and V. V. Veeravalli, “Wireless sensors in distributed detection applications,” *IEEE Signal Processing Magazine*, vol. 25, pp. 16–25, May 2007.

[56] R. J. Radke, “A survey of distributed computer vision algorithms,” in *Aghajan (Eds.), Handbook of Ambient Intelligence and Smart Environments*, Springer, 2008.

[57] H. Medeiros, J. Park, and A. Kak, “Distributed object tracking using a cluster-based kalman filter in wireless camera networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, pp. 448–463, Aug. 2008.

[58] Y.-A. Le Borgne, S. Raybaud, and G. Bontempi, “Distributed principal component analysis for wireless sensor networks,” *Sensors*, vol. 8, no. 8, pp. 4821–4850, 2008.

[59] A. Wiesel and A. O. Hero, “Decomposable principal component analysis,” *IEEE Trans. Signal Processing*, vol. 57, pp. 4369–4377, Nov. 2009.

[60] M. Gastpar, P. Dragotti, and M. Vetterli, “The distributed karhunen loeve transform,” *IEEE Trans. Information Theory*, vol. 52, pp. 5177–5196, Dec. 2006.

[61] A. Amar, A. Leshem, and M. Gastpar, “A greedy approach to the distributed karhunen-loeve transform,” in *IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, (Dallas, TX), pp. 2970–2973, Mar. 2010.

[62] H. I. Nurdin, R. R. Mazumdar, and A. Bagchi, “On the estimation and compression of distributed correlated signals with incomplete observations,” in *Proc. Mathematical Theory of Networks and Systems (MTNS 2004)*, 2004.

[63] O. Roy and M. Vetterli, “Dimensionality reduction for distributed estimation in the infinite dimensional regime,” *IEEE Trans. information theory*, vol. 54, Apr. 2008.

[64] A. Ciancio and A. Ortega, “A distributed wavelet compression algorithm for wireless sensor networks using lifting,” in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, (Montreal, Quebec, Canada), May 2004.

[65] A. Ciancio, S. Pattem, A. Ortega, and B. Krishnamachari, “Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm,” in *Proc. 5th Int. Conf. Information Processing in Sensor Networks (IPSN)*, (Nashville, TN), Apr. 2006.

[66] G. Shen, S. Pattem, and A. Ortega, “Energy-efficient graph-based wavelets for distributed coding in wireless sensor networks,” in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, (Taipei, Taiwan), Apr. 2009.

[67] J. Yoder, H. Medeiros, J. Park, and A. Kak, “Cluster-based distributed face tracking in camera networks,” *IEEE Trans. Image Processing*, vol. 19, pp. 2551–2563, Oct. 2010.

[68] P. K. Varshney, *Distributed Detection and Data Fusion*. New York, NY: Springer-Verlag, 1997.

[69] A. Lakhina, M. Crovella, and C. Diot, “Diagnosing network-wide traffic anomalies,” in *Proc. 2004 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM’04)*, vol. 34, pp. 219–230, Oct. 2004.

[70] A. Lakhina, M. Crovella, and C. Diot, “Mining anomalies using traffic feature distributions,” in *Proc. 2005 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM’05)*, vol. 34, pp. 217–228, Oct. 2005.

[71] T. Ahmed, B. Oreshkin, and M. Coates, “Machine learning approaches to network anomaly detection,” in *Proc. Second Workshop on Tackling Computer Systems Problems with Machine Learning (SysML)*, (Cambridge, MA), Apr. 2007.

[72] V. Saligrama, J. Konrad, and P.-M. Jordoin, “Video anomaly identification,” *IEEE Signal Processing Magazine*, vol. 27, pp. 18–32, Sep. 2010.

[73] S. Hariharan, L. R. Bachega, N. Shroff, and C. A. Bouman, “Communication efficient signal detection in correlated clutter for wireless sensor networks,” in *Asilomar*, (Pacific Grove, CA), Nov. 2010.

[74] J. Rissanen, “Modeling by the shortest data description,” *Automation*, vol. 14, pp. 465–471, 1978.

[75] J. Rissanen, “A universal prior for integers and estimation by minimum description length,” *The Annals of Statistics*, vol. 11, no. 2, pp. 416–431, 1983.

[76] M. H. Hansen and B. Yu, “Model selection and the principle of minimum description length,” *Journal of the American Statistical Association*, vol. 96, pp. 746–774, 1998.

[77] L. R. Bachega, J. Theiler, and C. A. Bouman, “Evaluating and improving local hyperspectral anomaly detectors,” in *Proc. Applied Imagery Pattern Recognition Workshop (AIPR), 2011 IEEE*, (Washington, DC), pp. 1–8, Oct. 2011.

[78] G. Deninan, B. Bhanu, H. Nguyen, C. Ding, A. Kamal, C. Ravishankar, A. Roy-Chowdhury, A. Ivers, and B. Varda, “Videoweb dataset for multicamera activities and non-verbal communication,” in *Distributed Video Sensor Networks* (B. Bhanu, C. Ravishankar, A. Row-Chowdhury, H. Aghajan, and D. Terzopoulos, eds.), Springer, 2010.

[79] S. Soderbergh, “Ocean’s eleven (film),” *Warner Bros.*, Dec. 2001.

APPENDIX

A. CHANGE IN LIKELIHOOD DUE TO THE DECORRELATING TRANSFORM, T

Let X be a $p \times n$ matrix with n p-dimensional samples with covariance R. Assuming the covariance can be decomposed into $R = T\Lambda T^t$, where Λ is diagonal and T is orthonormal, the Gaussian log likelihood of X is given by

$$\log p_{(T,\Lambda)}(\mathrm{X}) = -\frac{n}{2}\mathrm{trace}[\mathrm{diag}(T^tST)\Lambda^{-1}] - \frac{n}{2}\log(2\pi)^p|\Lambda| \ , \qquad \text{(A.1)}$$

where $S = \frac{1}{n}\mathrm{X}\mathrm{X}^t$ is the sample covariance. The maximum likelihood estimate of Λ given T is

$$\hat{\Lambda}(T) = \mathrm{diag}(\hat{T}^tS\hat{T}) \ .$$

The log likelihood in (A.1) maximized with respect to Λ is given by

$$\log p_{(T,\hat{\Lambda}(T))}(\mathrm{X}) = -\frac{np}{2} - \frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\mathrm{diag}(T^tST)| \ . \qquad \text{(A.2)}$$

Similarly, for $T = I$, where I is the $p \times p$ identity,

$$\log p_{(T,\hat{\Lambda}(I))}(\mathrm{X}) = -\frac{np}{2} - \frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\mathrm{diag}(S)| \ . \qquad \text{(A.3)}$$

Therefore, the change in likelihood due to T is given by the difference between (A.2) and (A.3):

$$\begin{aligned} \Delta \log p_{(T,\hat{\Lambda}(T))}(\mathrm{X}) &= \log p_{(T,\hat{\Lambda}(T))}(\mathrm{X}) - \log p_{(I,\hat{\Lambda}(I))}(\mathrm{X}) \\ &= -\frac{n}{2}\log\frac{|\mathrm{diag}(T^tST)|}{|\mathrm{diag}(S)|} \ . \qquad \text{(A.4)} \end{aligned}$$

B. THE CORRELATION SCORE

The correlation score is a measure of correlation between two vectors. This correlation score is used in Section 5.3.2 to select the most correlated pair of sensor vector output for decorrelation.

Definition B.0.1 *Let* x *and* y *be two vectors with covariances* R_x *and* R_y *respectively, and joint covariance* R_{xy}*. The vector correlation coefficient between* x *and* y *is*

$$F_{xy} = \sqrt{1 - \frac{|R_{xy}|}{|R_x||R_y|}}.$$

Proposition B.0.1 *Let* x *and* y *be* p*-dimensional Gaussian random vectors. The mutual information* [1] $I(\mathrm{x}, \mathrm{y})$ *between* x *and* y *in terms of their vector correlation coefficient is*

$$I(\mathrm{x}; \mathrm{y}) = -\frac{1}{2} \log \left(1 - F_{xy}^2\right) \ .$$

Proof

$$\begin{aligned} I(\mathrm{x}; \mathrm{y}) &= h(\mathrm{x}) + h(\mathrm{y}) - h(\mathrm{x}, \mathrm{y}) && \text{(B.1)} \\ &= \frac{1}{2} \log[(2\pi e)^p |R_x|] + \frac{1}{2} \log[(2\pi e)^p |R_y|] \\ &\quad -\frac{1}{2} \log[(2\pi e)^{2p} |R_{xy}|] && \text{(B.2)} \\ &= \frac{1}{2} \log \left[\frac{|R_x||R_y|}{|R_{xy}|}\right] && \text{(B.3)} \\ &= -\frac{1}{2} \log[1 - F_{xy}^2] && \text{(B.4)} \end{aligned}$$

■

Proposition B.0.2 *Let* x *and* y *be both unidimensional (scalar) Guassian random variables with covariances* σ_x^2 *and* σ_y^2*, respectively, and correlation coefficient* ρ_{xy}*. Then,* $F_{xy} = |\rho_{xy}|$.

[1] *Total correlation* is a related concept [28], generalizing the concept of mutual information to multiple random variables.

Proof We have that $|R_x| = \sigma_x^2$ and $|R_y| = \sigma_y^2$.

The covariance of the joint distribution of x and y is $R_{xy} = \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y \\ \rho_{xy}\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}$.

$$\begin{aligned}
F_{xy} &= \sqrt{1 - \frac{|R_{xy}|}{|R_x||R_y|}} && \text{(B.5)}\\
&= \sqrt{1 - \frac{\sigma_x^2\sigma_y^2 - \rho_{xy}^2\sigma_x^2\sigma_y^2}{\sigma_x^2\sigma_y^2}} && \text{(B.6)}\\
&= \sqrt{1 - (1 - \rho_{xy}^2)} && \text{(B.7)}\\
&= \sqrt{\rho_{xy}^2} && \text{(B.8)}\\
&= |\rho_{xy}| && \text{(B.9)}
\end{aligned}$$

■

VITA

VITA

Leonardo Ruggiero Bachega was born and raised in São Carlos-SP, Brazil. At the age of eleven, he fell in love with computers after learning how to write his first BASIC program in his parents' old IBM PC clone. As a teenager, he spent most of his free time writing Clipper programs, and later teaching Microsoft Office products at a local private school. His passion for everything related to computers was soon to become his passion for mathematics and science. He graduated from University of São Paulo (USP), Brazil with a Bachelor's Degree in Computational Physics in December, 2000. After graduation, he worked as a software engineer at Scopus Tecnologia, a major Brazilian software company. In October, 2002, he relocated to Weschester, New York to join the Blue Gene/L software team at the IBM T. J. Watson Research Center. Blue Gene/L was announced to be the fastest supercomputer in the world in 2004. After joining Purdue, he worked on several high-performance computing and compiler projects before concentrating his research on sparse matrices, under prof. Charles Bouman. Mr. Bachega received a Bachelors degree in Physics, and a Masters in Computer Engineering from University of São Paulo in 2000 and 2004, respectively. He also received a Masters of Science degree in Electrical and Computer Engineering from Purdue University in 2010. His main interests include modeling of high-dimensional data, pattern recognition, signal processing, machine learning, high-performance computing and compilers.