Distributed Signal Decorrelation and Detection in Sensor Networks Using the Vector Sparse Matrix Transform

Leonardo R. Bachega*, *Student Member, IEEE,* Srikanth Hariharan, *Student Member, IEEE*
Charles A. Bouman, *Fellow, IEEE,* and Ness Shroff, *Fellow, IEEE*

***Abstract*—In this paper, we propose the *vector SMT*, a new decorrelating transform suitable for performing distributed processing of high dimensional signals in sensor networks. We assume that each sensor in the network encodes its measurements into vector outputs instead of scalar ones. The proposed transform decorrelates a sequence of pairs of vector sensor outputs, until these vectors are decorrelated. In our experiments, we simulate distributed anomaly detection by a network of cameras monitoring a spatial region. Each camera records an image of the monitored environment from its particular viewpoint and outputs a vector encoding the image. Our results, with both artificial and real data, show that the proposed vector SMT transform effectively decorrelates image measurements from the multiple cameras in the network while maintaining low overall communication energy consumption. Since it enables joint processing of the multiple vector outputs, our method provides significant improvements to anomaly detection accuracy when compared to the baseline case when the images are processed independently.**

***Index Terms*—Sparse Matrix Transform, Wireless Sensor Networks, Smart Camera Networks, Distributed Signal Processing, Distributed Anomaly Detection, Multiview Image Processing, Pattern Recognition**

I. Introduction

In recent years, there has been significant interest in the use of sensor networks for distributed monitoring in many applications [1], [2]. In particular, networks with camera sensors have gained significant popularity [3], [4]. 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. Since 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 [5]. 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. Fig. 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.

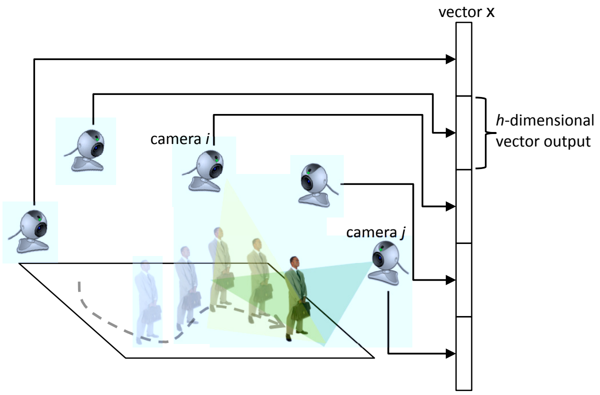


Fig. 1: A camera network where each camera captures an image of the environment from one viewpoint and encodes the image into a vector output. The aggregated outputs from all cameras form the high-dimensional vector, x. Cameras i and j have overlapping views. Since outputs from cameras with overlapping views tend to be correlated, so does the aggregated vector x.

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 [6], [7], 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

L.R. Bachega and C.A. Bouman are with the School of Electrical and Computer Engineering, Purdue University, 465 Northwestern Ave., West Lafayette, IN 47907-2035, USA. Tel: 765-494-6553, Fax: 765-494-3358, E-mail: {lbachega, bouman}@purdue.edu.

S. Hariharan is with AT&T Labs, San Ramon, CA 94583, USA. Email: srikanth.hariharan@gmail.com

N. B. Shroff is with the Departments of ECE and CSE at the Ohio State University, Columbus, OH 43210, USA. Email: shroff@ece.osu.edu

This material is based upon work supported by, or in part by, the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number 56541-CI, and by grant from the Army Research Office MURI Award W911NF-07-10376 (SA08-03).

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 [8] and [9]. 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 [10], [11], [12], [13] 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 [14], two-dimensional networks [15], and networks with tree topology [16]. 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 [17].

Multiple efforts have been made in distributed detection since the early 1980s (see [18] for a survey). Most 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 [19], [20], [21]. 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 [22] for a survey). The method in [21] 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 [23]. 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 [24], [25], [26], [27], [28], [29]. Among these methods, the Sparse Matrix Transform (SMT) [28], here referred to as the 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 [30], [31], [32] and Givens rotations have been used in ICA [33]. Since it involves only pairwise operations between coordinate pairs, it is well-suited to distributed decorrelation [34]. However, this existing method is only well suited for decorrelation of scalar sensor outputs.

In this paper, 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 [28] 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. Since 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 communication 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 the 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 [35], and their total correlation [36].

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 the 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 paper is organized as follows. Sec. II describes the main concepts of the scalar SMT. Sec. III introduces the vector SMT algorithm, designed to perform distributed decorrelation of vector sensor outputs in a sensor network. Sec. IV shows how to use the vector SMT to enable distributed detection in a sensor network. Sec. V shows experimental results of detection using data from multi-camera views of objects as well as simulated data. Finally, the main conclusions and future work are discussed in Sec. VI.

II. 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) [28] models the orthonormal matrix E as the product of K sparse matrices, E_K, so that

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

In (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,

$$E_k = I + \Theta(i_k, j_k, \theta_k) \ , \tag{2}$$

where

$$[\Theta]_{ij} = \begin{cases} \cos(\theta_k) - 1 & \text{if } i = j = i_k \text{ or } i = j = j_k \\ \sin(\theta_k) & \text{if } i = i_k \text{ and } j = j_k \\ -\sin(\theta_k) & \text{if } i = j_k \text{ and } j = i_k \\ 0 & \text{otherwise} \end{cases} \ . \tag{3}$$

This SMT model assumes that K Givens rotations in (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). Since 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 much 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. Figs. 2(b) and (c) make a visual comparison of the FFT and the Scalar SMT. The SMT rotations can operate on pairs of coordinates in any order, while in the FFT, the butterflies are constrained to a well-defined sequence with specific rotation angles.

The scalar SMT design consists in learning the product in (1) from a set of n independent and identically distributed training vectors, $X = [x_1, \cdots, x_n]$, from $\mathcal{N}(0,R)$. Assuming that $R = E\Lambda E^t$, the maximum likelihood estimates of E and Λ are given by

$$\hat{E} = \arg\min_{E \in \Omega_K} \left\{ \left| \mathrm{diag}(E^t S E) \right| \right\} \tag{4}$$

$$\hat{\Lambda} = \mathrm{diag}(\hat{E}^t S \hat{E}) \ , \tag{5}$$

where $S = \frac{1}{n} X X^t$, and Ω_K is the set of allowed orthonormal transforms. The functions $\mathrm{diag}(\cdot)$ and $|\cdot|$ are the diagonal and determinant, respectively, of a matrix argument. With the SMT model assumption, the orthonormal transforms in Ω_K are in the form of (1), and the total number of planar rotations, K is the model order parameter.

When performing an unconstrained minimization of (4) by allowing the set Ω_K to contain all orthonormal transforms, when $n > p$, the minimizer,$\hat{E}$ is the orthonormal matrix that diagonalizes 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 [28], the greedy optimization of (4) under the constraint that the allowed transforms are in the form of (1) yields accurate estimates even when $n \ll p$.

The constraint in (1) is non-convex with no obvious closed form solution. In [28], we use a greedy optimization approach in which we select each Givens rotation, E_k, independently, in sequence to minimize the cost in (4). The model order parameter K can be estimated using cross-validation over the training set [37], [38] or using the minimum description length (MDL) [32].

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.

III. Distributed Decorrelation with the Vector Sparse Matrix Transform

Our goal is to decorrelate the p-dimensional vector x aggregated from outputs of all sensors, where each of the L sensors outputs an h-dimensional sub-vector of x. 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 Sec. II to the decorrelation of pairs of vectors instead of pairs of coordinates.

A. The Vector SMT Model

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

$$x = \begin{bmatrix} x^{(1)} \\ \hline \vdots \\ \hline x^{(L)} \end{bmatrix} \ ,$$

where each sub-vector, $x^{(i)}$ 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,

$$T = \prod_{m=1}^{M} T_m \ , \tag{6}$$

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 $x^{(i_m)}$, $x^{(j_m)}$, as illustrated in Fig. 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 (2) to a transform that operates on pairs of sub-vectors instead of coordinates. Similarly, the vector SMT in (6) generalizes the concept of the scalar SMT in Sec. II: it decorrelates a high-dimensional vector by decorrelating its pairs of sub-vectors instead of pairs of coordinates. Figs. 2(b) and (d) compare the vector and the scalar SMTs approaches graphically. In the scalar SMT, each Givens rotation E_k plays the role of a "decorrelating butterfly" (Fig. 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 (6), illustrated in Fig. 2(d), decorrelates M pairs of sub-vectors of x, until the decorrelated vector $\widetilde{x}$ is obtained.

In a sensor network, we compute the distributed decorrelation of x by distributing the application of transforms T_m from the product (6) across multiple sensors. Before the decorrelation, each sub-vector $x^{(i)}$ of x is the output of a sensor i and is stored locally in that sensor. Applying each T_m to sub-vectors $x^{(i_m)}$, $x^{(j_m)}$ 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 $\tilde{x}^{(i_m)}$ and $\tilde{x}^{(j_m)}$ are cached at the sensor used to compute this pairwise decorrelation, avoiding communicating one sub-vector back to its originating sensor. Finally, the total communication energy required for the entire decorrelation is given by

$$\mathcal{E}(h, i_1, \cdots, i_M, j_1, \cdots, j_M) = \sum_{m=1}^{M} \mathcal{E}(h, i_m, j_m). \tag{7}$$

B. 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 $X = [x_1, \cdots, x_n] \in \mathbb{R}^{p \times n}$, 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 be 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 T and Λ is given by

$$\log p_{(T,\Lambda)}(X) = -\frac{n}{2} \mathrm{tr}[\mathrm{diag}(T^t S T) \Lambda^{-1}] - \frac{n}{2} \log(2\pi)^p |\Lambda| \ , \tag{8}$$

where $S = \frac{1}{n} X X^t$. When constraining T to be of the product form of (6), the joint maximum likelihood estimates $\widehat{\Lambda}$ and $\widehat{T}$ are given by

$$\hat{T} = \arg \min_{T = \prod_{m=1}^{M} T_m} \left\{ \left| \mathrm{diag}(T^t S T) \right| \right\} \tag{9}$$

$$\hat{\Lambda} = \mathrm{diag}(\hat{T}^t S \hat{T}) \ . \tag{10}$$

Since the minimization in (9) 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 $X_1 = X$, and iterate over the following steps:

$$\hat{T}_m = \arg \min_{T_m \in \Omega} \left\{ \left| \mathrm{diag}(T_m^t S_m T_m) \right| \right\} \tag{11}$$

$$S_{m+1} = \hat{T}_m^t S_m \hat{T}_m \tag{12}$$

$$X_{m+1} = \hat{T}_m^t X_m \ , \tag{13}$$

where Ω is the set of all allowed pairwise transforms. Since T_m operates exclusively on $x^{(i_m)}$ and $x^{(j_m)}$, 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 $X_m^{(i_m)}$ and $X_m^{(j_m)}$ 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

$$S_m^{(i_m,j_m)} = \frac{1}{n} \left[\frac{X_m^{(i_m)}}{X_m^{(j_m)}} \right] \left[X_m^{(i_m)t} | X_m^{(j_m)t} \right] \ . \tag{14}$$

The minimization in (11) 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,

$$E_m = \arg \min_{E \in \Omega_{2h \times 2h}} \left\{ \left| \mathrm{diag}(E^t S_m^{(i_m,j_m)} E) \right| \right\} \ , \tag{15}$$

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 Sec. II. Once E_m is selected, we partition it into four $h \times h$ blocks,

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

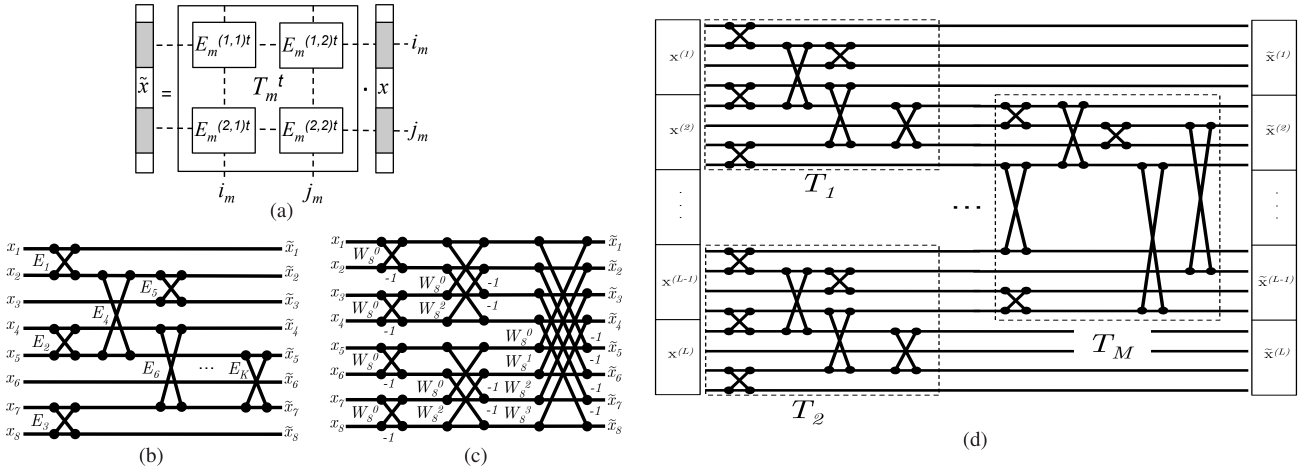
and then we obtain the transform T_m using Kronecker product $\otimes$ as

$$\begin{aligned} T_m = & J^{(i_m,i_m)} \otimes E_m^{(1,1)} + J^{(i_m,j_m)} \otimes E_m^{(1,2)} \\ & + J^{(j_m,i_m)} \otimes E_m^{(2,2)} + J^{(j_m,j_m)} \otimes E_m^{(2,1)} \ , \\ & + I_{p \times p} - \left(J^{(i_m,i_m)} + J^{(j_m,j_m)} \right) \otimes I_{h \times h} \end{aligned} \tag{16}$$

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

$$\left[J^{(i,j)} \right]_{i'j'} = \begin{cases} 1 \, \text{if } i' = i \text{ and } j' = j \\ 0 \, \text{otherwise} \end{cases} \ . \tag{17}$$

Fig. 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


$\tilde{x}$
=
$E_m^{(1,1)t}$
$E_m^{(1,2)t}$
$T_m^{\ t}$
$E_m^{(2,1)t}$
$E_m^{(2,2)t}$
·
x
i_m
j_m
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_8^2 W_8^1 W_8^3
-1
$x^{(1)}$ $x^{(2)}$ $x^{(L-1)}$ $x^{(L)}$
$\tilde{x}^{(1)}$ $\tilde{x}^{(2)}$ $\tilde{x}^{(L-1)}$ $\tilde{x}^{(L)}$
T_1
T_2
T_M
···

(a)

(b)

(c)

(d)

Fig. 2: (a) In the product $\tilde{x} = T_m^t x$, the $p \times p$ block-wise sparse transform T_m operates over the p-dimensional vector x, changing only the $2h$ components associated with the h-dimensional 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.

the log likelihood in (8) due to applying T_m to X_m and maximized with respect to $\hat{\Lambda}(T_m)$ is given by (see App. A)

$$\begin{aligned}\Delta \log p_{(T_m,\hat{\Lambda}(T_m))}(X_m) &= -\frac{n}{2}\log\frac{|\mathrm{diag}(T_m^t S_m T_m)|}{|\mathrm{diag}(S_m)|}\\ &= -\frac{n}{2}\log\frac{|\mathrm{diag}(E_m^t S_m^{(i_m,j_m)} E_m)|}{|\mathrm{diag}(S_m^{(i_m,j_m)})|}\\ &= -\frac{n}{2}\log\left(1 - F^2_{i_m j_m}\right) \ , \qquad (18)\end{aligned}$$

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 App. 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 (18). 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 (11). Finally, the algorithm in Fig. 3(b) summarizes this greedy procedure to design the vector SMT.

C. The Vector SMT Design with Communication Energy Constraints

We extend the vector SMT design in Sec. III-B to account for the communication energy required for distributed decorrelation in a sensor network. When each T_m operates on $x^{(i_m)}$ and $x^{(j_m)}$ 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. We augment the likelihood in (8) with a linear penalization term associated with the total communication energy required for distributed decorrelation. The augmented log likelihood is given by

$$\mathcal{L}_{(T,\Lambda)}(X) = \log p_{(T,\Lambda)}(X) - \mu \sum_{m=1}^{M} \mathcal{E}(h, i_m, j_m) \ . \quad (19)$$

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 Sec. III-B. When we apply T_m to X_m and maximize (19) with respect to $\hat{\Lambda}(T_m)$, the overall change in the augmented likelihood is given by

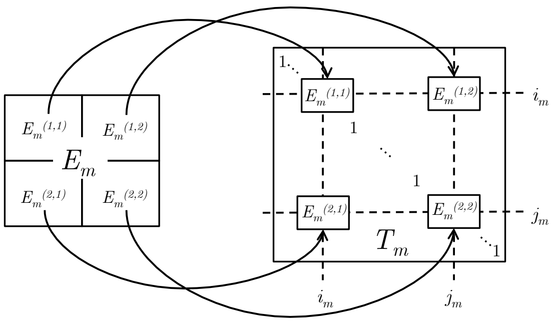
$$\begin{aligned}\Delta\mathcal{L}_{(T_m,\hat{\Lambda}(T_m))}(X_m) &= \mathcal{L}_{(T_m,\hat{\Lambda}(T_m))}(X_m) - \mathcal{L}_{(I,\hat{\Lambda}(I))}(X_m)\\ &= -\frac{n}{2}\log\left\{\frac{|\mathrm{diag}(T_m^t S_m T_m)|}{|\mathrm{diag}(S_m)|}\right\}\\ &\quad -\mu\mathcal{E}(h, i_m, j_m) \qquad (20)\\ &= -\frac{n}{2}\log\left(1 - F^2_{i_m j_m}\right) - \mu\mathcal{E}(h, i_m, j_m)\end{aligned}$$

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 $(1 - F^2_{i_m,j_m})e^{2\mu\mathcal{E}(h,i_m,j_m)/n}$, 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 (20) by the largest amount.

D. 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.

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



(a)

```
//Initialization
forall the $1 \leq i \leq L$ and $1 \leq j \leq L$ do
    $S^{(i,j)} \leftarrow \frac{1}{n}\left[\frac{X^{(i)}}{X^{(j)}}\right]\left[X^{(i)t}|X^{(j)t}\right]$
    $E \leftarrow ComputeScalarSMT(S^{(i,j)})$
    $F_{ij} \leftarrow \left(1 - \frac{|\mathrm{diag}(E^t S^{(i,j)} E)|}{|\mathrm{diag}(S^{(i,j)})|}\right)^{\frac{1}{2}}$
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. 3: (a) Mapping from the $2h \times 2h$ orthonormal matrix, E to the $p \times p$ block-wise sparse matrix T_m associated with the (i_m, j_m) sub-vector pair. (b) The vector SMT design algorithm.

2) Cross-Validation: We partition the $p \times n$ data sample matrix X into $\mathcal{K}$, $p \times n_k$ matrices $X_{(k)}$, $X = [X_{(1)}|\cdots|X_{(\mathcal{K})}]$, and define $\bar{X}_{(k)}$ as a matrix containing the samples in X that are not in $X_{(k)}$. For each $k = 1, \cdots, \mathcal{K}$, we design $\mathcal{M}_M$ from $\bar{X}_{(k)}$, and compute its log likelihood over X_k, i.e., $\log p_{\mathcal{M}_M}(X_{(k)}|\bar{X}_{(k)})$. We select M so that it maximizes the average cross-validated log likelihood [39],

$$L(\mathcal{M}_M) = \frac{1}{\mathcal{K}}\sum_{i=1}^{\mathcal{K}} \log p_{\mathcal{M}_M}(X_{(k)}|\bar{X}_{(k)}) \ . \qquad (21)$$

3) Minimum Description Length (MDL) Criterion: Based on the MDL principle [40], [41], [42], 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

$$\begin{aligned}\ell_M =& -\log p_{\mathcal{M}_M}(X) + \frac{1}{2}MK\log(pn) + 2MK\log(2h) \\ &+2M\log(L) \ , \end{aligned} \qquad (22)$$

where $-\log p_{\mathcal{M}_M}(X)$ 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 (22). Initially, ℓ_M decreases with M because it is dominated by the likelihood term, $\log p_{\mathcal{M}_M}(X)$. 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

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

This condition leads to this new stop condition for the main loop of the algorithm in Fig. 3(b),

$$F^2_{i_m,j_m} \geq 1 - e^{\frac{K\log(pn)+4K\log(2h)+4\log(L)}{n}} \ . \qquad (23)$$

It is easy to generalize ℓ_M in (22) to the case where each pairwise transform, T_m has a different number of Givens rotations, K_m, resulting in

$$\begin{aligned}\ell_M^{(general)} =& -\log p_{\mathcal{M}_M}(X) + \frac{1}{2}\sum_{m=1}^{M} K_m\log(pn) \\ &+2\sum_{m=1}^{M} K_m\log(2h) + 2M\log(L) \ . \end{aligned} \qquad (24)$$

Finally, when $\ell_{M+1}^{(general)} - \ell_M^{(general)} \geq 0$ is satisfied, the new stop condition for the loop in Fig. 3(b) is given by

$$F^2_{i_m,j_m} \geq 1 - e^{\frac{K_{m+1}\log(pn)+4K_{m+1}\log(2h)+4\log(L)}{n}} \ . \qquad (25)$$

IV. 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 [23]. Here, we first formulate the anomaly detection problem, and then describe the ellipsoid volume measure of detection accuracy [43] used in the experimental section.

A. 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 $\mathcal{U}(x) = c$, for some constant c [44], [45]. Formally, we have the following hypotheses,

$$\begin{aligned}\mathcal{H}_0 &: x \sim \mathcal{N}(0, R) \\ \mathcal{H}_1 &: x \sim \mathcal{U},\end{aligned} \qquad (26)$$

where $\mathcal{H}_0$ and $\mathcal{H}_1$ are the null and alternative hypotheses respectively. According to the Neyman-Pearson lemma [23], the optimal classifier has the form of the log likelihood ratio test,

$$\begin{aligned}\Gamma(x) =& \log\left\{\frac{p(x;\mathcal{H}_1)}{p(x;\mathcal{H}_0)}\right\} = \log c - \log p(x;\mathcal{H}_0) \\ =& \log c + \frac{p}{2}\log 2\pi + \frac{1}{2}\log|R| + \frac{1}{2}x^t R^{-1} x \gtrless \Gamma_0 \ . \end{aligned} \qquad (27)$$

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 is controlled by the threshold Γ_0. We incorporate all the constant terms into a new threshold, η^2, such that the test in (27) becomes

$$D_R(x) = x^t R^{-1} x \gtrless \eta^2 . \tag{28}$$

If we further assume that $R = T\Lambda T^t$, where T and Λ are orthonormal and diagonal matrices respectively, the test in (27) can be written as a weighted sum of p uncorrelated coordinates,

$$\widetilde{D}_\Lambda(\tilde{x}) = \sum_{i=1}^{p} \frac{\tilde{x}_i^2}{\lambda_i} \gtrless \eta^2 \tag{29}$$

where $\tilde{x} = T^t x$, and $\lambda_i \equiv [\Lambda]_{ii}$ $(1 \leq i \leq p)$. Finally, because the sum in (29) involves only independent terms, it can be evaluated distributedly across a sensor network while requiring minimum communication.

B. Ellipsoid Volume as a Measure of Detection Accuracy

The ellipsoid volume approach [32], [43], [46] 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, $p(\mathcal{H}_1;\mathcal{H}_1)$ required for ROC analysis [23]. 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, $x^t \hat{R}^{-1} x \leq \eta^2$ for a certain probability of false alarm controlled by η. Such a volume is evaluated by

$$V(\hat{R},\eta) = \frac{\pi^{p/2}}{\Gamma(1+p/2)} \eta^p \sqrt{|\hat{R}|} \ . \tag{30}$$

We use $V(\hat{R},\eta)$ as a proxy for the probability of missed detection, $1 - p(\mathcal{H}_1;\mathcal{H}_1)$. 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.

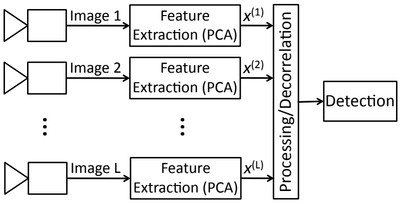
V. Experimental Results

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. Fig. 4(a) summarizes these approaches in terms of their main computation and communication characteristics. Finally, Fig. 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 principal component analysis (PCA). We process the outputs using one of the approaches in Fig. 4(a) before making a detection decision.

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	–

(a)



(b)

Fig. 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. 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 Sec. III-C. We simulate a network with $L = 31$ sensors, in which each sensor i outputs a vector, $x^{(i)}$ with $h = 25$ dimensions. These sensor vector outputs are correlated. Fig. 5 shows how we generate a data sample x, aggregated from correlated sensor outputs $x^{(i)}$, $i = 1, \cdots, 31$. First, we draw each $x^{(i)}$ 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 perform random permutations of the individual coordinates of x across all $x^{(i)}$, $i = 1, \cdots, 31$, 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.

Fig. 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 Karhunen-Loève (eigenvector matrix from

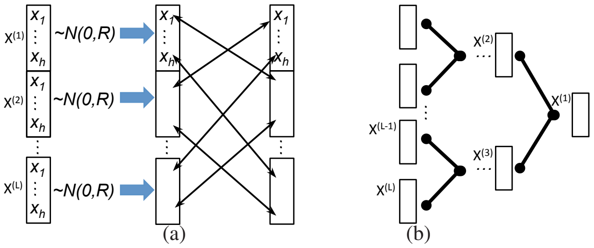


Fig. 5: Generation of a data sample, x aggregated from correlated h-dimensional sensor outputs $x^{(i)}$, $i = 1, \cdots, L$, using an artifical model. (a) First we draw each $x^{(i)}$ independently from the $\mathcal{N}(0, R)$ distribution, with $[R]_{rs} = \rho^{|r-s|}$. Then, we permute individual 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.

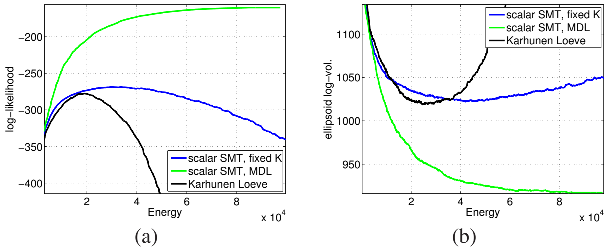


Fig. 6: Vector SMT model accuracy *vs.* communication energy consumption using 100 training data samples from an artificial model. Comparison of different vector SMT pairwise transforms for a range of communication energies: (a) average log-likelihood over 300 test samples; (b) ellipsoid log-volume covering 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.

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 (Fig. 6(a)), and the ellipsoid log-volume covering 99% of the testing samples, i.e., for 1% false alarm rate (Fig. 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 [28], helping to prevent overfitting. Since it is overall the most accurate, the scalar SMT-MDL is our pairwise transform of choice during all other experiments in this paper.

Fig. 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 (Fig. 7(a)) and ellipsoid log-volume covering 99% of the testing samples (Fig. 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.

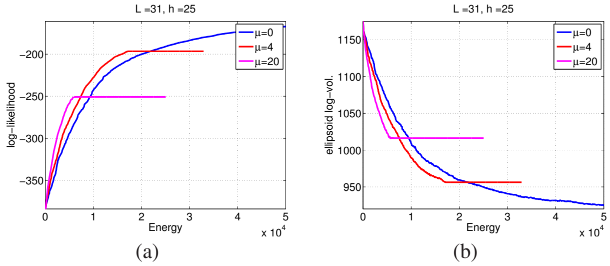


Fig. 7: Comparison of vector SMT energy constraint parameter values for a range of communication energies using 100 training data samples from an artificial model. (a) average log-likelihood over 300 test samples; (b) ellipsoid log-volume covering 99% of the test samples (1% false alarm rate). Vector 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.

B. 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 Figs. 8(a) and (b). We refer to sphere positions along the line in Fig. 8(a) as typical ones, while referring to positions along the mirrored diagonal line in Fig. 8(b) as anomalous ones. Two cameras ($L = 2$) monitor the sphere locations in the 3D region. Fig. 8(c) shows the top (X-Y) view captured by camera 1, while Fig. 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 Figs. 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.

Figs. 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 (Fig. 8(e)) and ellipsoid log-volume coverage plot (Fig. 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, anomaly detection is very accurate.

Fig. 9 shows sets with five eigen-images associated with the largest eigenvalues for both the independent (Fig. 9(a)) and the vector SMT (Fig. 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.

C. 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

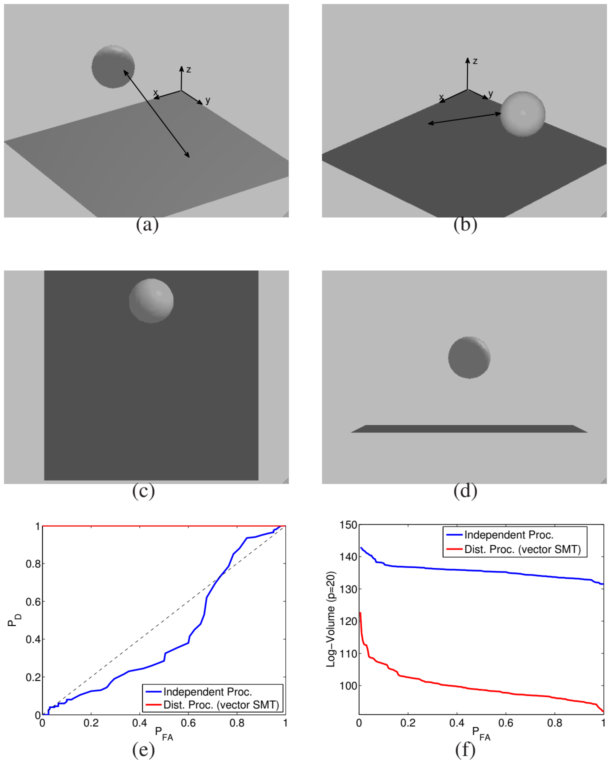


Fig. 8: Simulated 3D space with bouncing sphere: the sphere takes random positions along the line indicated by the double arrow (a) typical behavior; (b) anomalous behavior. The camera views: (c) top (X-Y dimensions); (d) side (X-Z dimensions). The detection accuracies using independent processing and vector SMT joint processing: (e) ROC curve; (f) "coverage plot" with log-volume of ellipsoid *vs.* probability of false alarm.

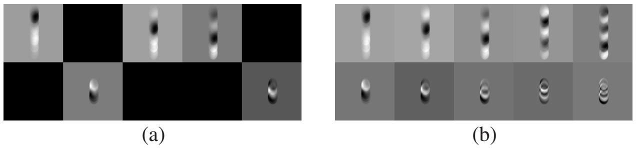


Fig. 9: Eigen-images of the moving sphere experiment. The five eigen-images (columns) are associated with the five largest eigenvalues in decreasing order (left-to-right). Each eigen-image has two views (top and bottom rows). (a) when the camera views are processed independently, each eigenvector models a single view; (b) when the camera views are processed jointly using the vector SMT, each eigenvector models both views together.

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 generated from the $\mathcal{N}(0, I_{3\times3})$ 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 $\mathcal{N}(0, I_{3\times3})$ 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. Fig. 10 shows the twelve camera views for both a typical cloud sample (Fig. 10(a)), and for an anomalous one (Fig. 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.

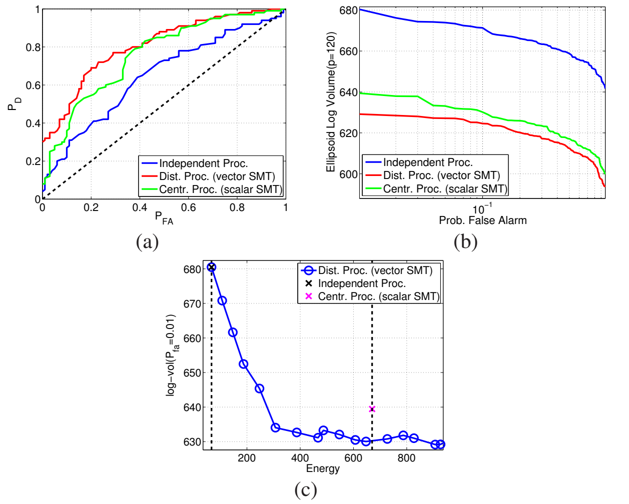


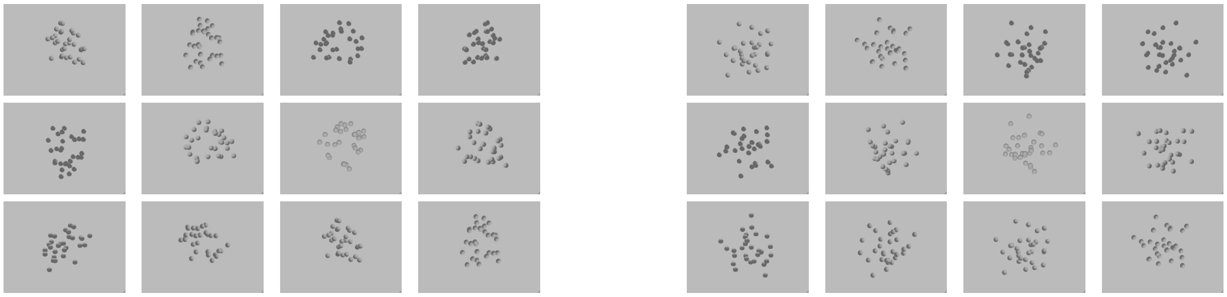
Fig. 11: Anomaly detection accuracy using the sphere cloud data: (a) ROC analysis; (b) log-volume of ellipsoid *vs.* probability of false alarm. Vector SMT decorrelation yields to the most accurate detection results for all false alarm rates. (c) log-volume of ellipsoid for 1% false alarm rate, i.e., 99% coverage *vs.* communication energy.

Fig. 11 shows anomaly detection accuracy based on ROC analysis (Fig. 11(a)), and log-volume of ellipsoid (Fig. 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 Fig. 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.

Fig. 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 is 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 for the same energy consumption. This difference in accuracy is due to the inherent constraint of the vector SMT decorrelating pairs of vectors, which tends to produce better models of a distribution when a limited number of training samples is available.

D. *Simulation experiments using real multi-camera images*

Fig. 12 shows $L = 8$ camera views of a courtyard, constructed from video sequences from the UCR Videoweb Activities Dataset [47]. Each camera records a video sequence of approximately 4.2 min, with 30 frames/sec, generating a total of 7600 frames. The sequences are



(a) (b)

Fig. 10: The twelve camera views of a 3D sphere cloud sample: (a) a typical sample (hollow cloud); (b) an anomalous sample (dense cloud). It is difficult to discriminate anomalous from typical samples by processing each view independently. Instead, the discriminant information is contained in the joint camera views.

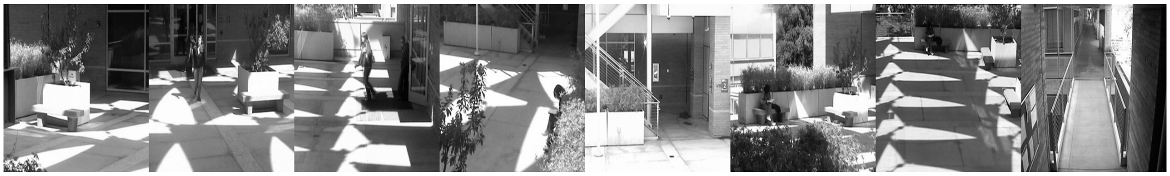


Fig. 12: The courtyard dataset from the UCR Videoweb Activities Dataset: eight cameras, with ids 1 to 8 from left to right, monitor a courtyard from different viewpoints. Several activities in the courtyard are captured simultaneously by several cameras.

TABLE I: 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	1.00	0.72	0.59	0.66	0.61	0.74	0.72	0.00
2	-	1.00	0.59	0.66	0.59	0.70	0.76	0.00
3	-	-	1.00	0.61	0.49	0.59	0.62	0.00
4	-	-	-	1.00	0.57	0.66	0.68	0.00
5	-	-	-	-	1.00	0.59	0.60	0.00
6	-	-	-	-	-	1.00	0.72	0.00
7	-	-	-	-	-	-	1.00	0.00
8	-	-	-	-	-	-	-	1.00

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 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 I 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 negligible correlation score values.

Fig. 13 shows two eigen-images associated with the two largest eigenvalues for both the independent and vector SMT approaches. In the independent processing case (Fig. 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 (Fig. 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.

Fig. 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. Fig. 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 Sec. V-C, 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. Fig. 14(b) shows results of the same experiment as in Fig. 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. Fig. 14(c) shows the ellipsoid log-volume for a fixed false alarm rate (0.8%) *vs.* communication energy. We observe the same trends observed in the sphere cloud experiment in Sec. V-C. 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.

Figs. 15(a)-(c) show ROC curves for detection of anomalous samples generated by an artificial 4-fold increase in the largest component of the vector output of a single camera view, and injected in views 2, 6, and 8, respectively. We use 200 typical samples to learn the decorrelating transform and the remaining samples for testing. Since views 2 and 6 are correlated with other views (see Table I), detection of anomalies in these views is accurate when 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.

Figs. 15(d)-(f) show the ROC curves for detection of what we call the “Ocean’s Eleven” anomaly, injected into

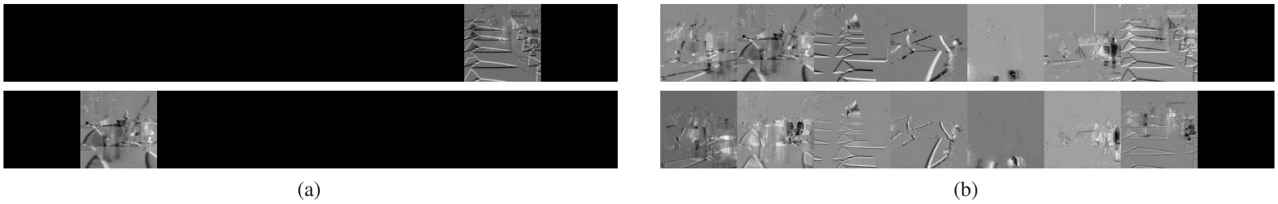


Fig. 13: Two eigen-images from the eight camera views of the courtyard dataset. Each eigen-image has eight views (columns) associated to it. (a) independent processing of camera views: each eigen-image corresponds to a single view and does not contain correlation information among multiple views; (b) joint processing modeled by the vector SMT: each eigen-image contains joint information of all correlated views.

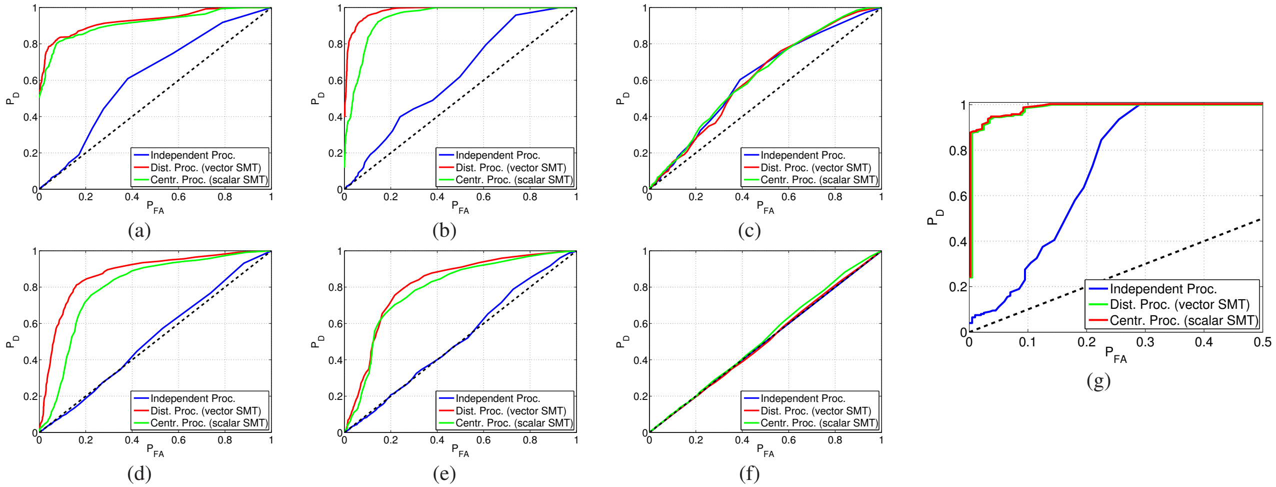


Fig. 15: ROC analysis of detection accuracy: (a)-(c) artificially generated anomalies by a 4-fold increase in the largest eigenvalue of a single view for views 2, 6 and 8, respectively. (d)-(f) Ocean's Eleven anomalies, generated by swapping images of a single camera view between samples for views 2, 6 and 8, respectively. Decorrelation improves detection accuracy when anomalies appear in correlated camera views (2 and 6). When the anomaly is inserted in a uncorrelative view (8), decorrelation methods do not improve the detection accuracy. (g) people coalescing in the middle of a courtyard: scalar and vector SMTs are highly accurate for small probabilities of false alarm with vector SMT consuming approximately 60% of communication energy required for the scalar SMT.

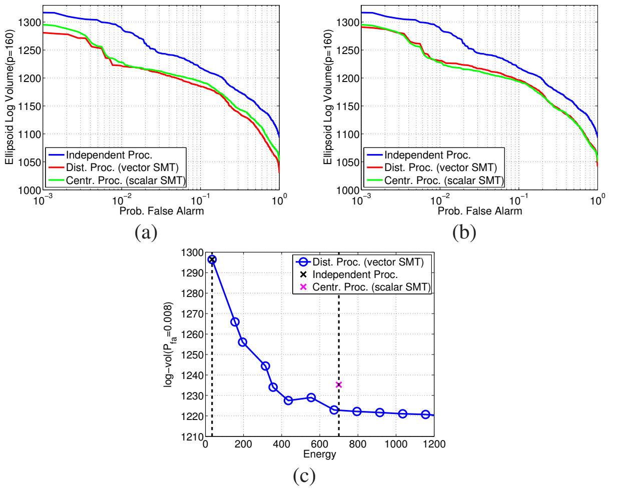


Fig. 14: Detection accuracy measured by the ellipsoid log-volume for the courtyard data set. Coverage plots showing the log-volume *vs.* probability of false alarm: (a) model order, $M = 7$, matching the energy of centralized processing, (b) model order, $M = 4$, matching 50% of the energy consumed for 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.

the camera views 2, 6, and 8, respectively. This anomaly is generated by swapping images of a single view 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 [48]. Since 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.

Fig. 15(g) shows the ROC curves for detection of a suspicious (anomalous) activity where people coalesce in at the center of the courtyard. Fig. 16 shows the typical and anomalous samples used in this experiment. We select 200 samples where a group of people coalesces at the center of the courtyard and label them as anomalous, while selecting another 200 samples where the group does not coalesce and label them as typical. We use another 300 typical samples to train the vector SMT. 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 Sec. V-C), 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.

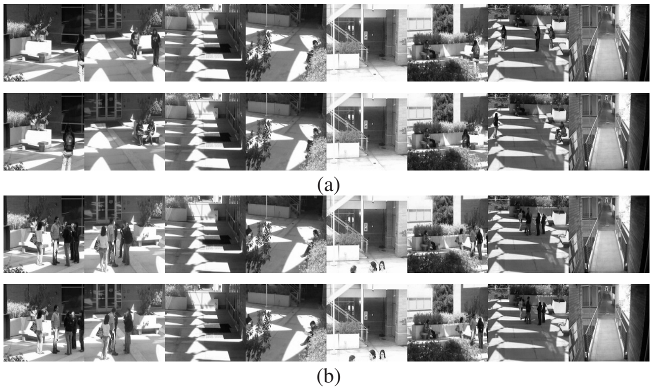


Fig. 16: Samples used in the experiment detecting people coalescing in the middle of the courtyard: (a) Typical samples; (b) Anomalous samples, with images of people coalescing.

VI. 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 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.

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

$$\log p_{(T,\Lambda)}(X) = -\frac{n}{2}\mathrm{tr}[\mathrm{diag}(T^tST)\Lambda^{-1}] - \frac{n}{2}\log(2\pi)^p|\Lambda| \,, \tag{31}$$

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

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

The log likelihood in (31) maximized with respect to Λ is given by

$$\log p_{(T,\hat{\Lambda}(T))}(X) = -\frac{np}{2} - \frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\mathrm{diag}(T^tST)| \ . \tag{32}$$

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

$$\log p_{(T,\hat{\Lambda}(I))}(X) = -\frac{np}{2} - \frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\mathrm{diag}(S)| \ . \tag{33}$$

Therefore, the change in likelihood due to T is given by the difference between (32) and (33):

$$\begin{aligned}\Delta \log p_{(T,\hat{\Lambda}(T))}(X) &= \log p_{(T,\hat{\Lambda}(T))}(X) - \log p_{(I,\hat{\Lambda}(I))}(X) \\ &= -\frac{n}{2}\log\frac{|\mathrm{diag}(T^tST)|}{|\mathrm{diag}(S)|} \ .\end{aligned} \tag{34}$$

B. The Correlation Score

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

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

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

Proof:

$$I(x;\mathrm{y}) = h(x) + h(\mathrm{y}) - h(x,\mathrm{y}) \tag{35}$$

$$\begin{aligned}&= \frac{1}{2}\log[(2\pi e)^p|R_x|] + \frac{1}{2}\log[(2\pi e)^p|R_y|] \\ &\quad -\frac{1}{2}\log[(2\pi e)^{2p}|R_{xy}|]\end{aligned} \tag{36}$$

$$= \frac{1}{2}\log\left[\frac{|R_x||R_y|}{|R_{xy}|}\right] \tag{37}$$

$$= -\frac{1}{2}\log[1 - F_{xy}^2] \tag{38}$$

■

Proposition A.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, $F_{xy} = |\rho_{xy}|$.

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 $R_{xy} = \begin{bmatrix} \sigma_x^2 & \rho_{xy}\sigma_x\sigma_y \\ \rho_{xy}\sigma_x\sigma_y & \sigma_y^2 \end{bmatrix}$.

$$F_{xy} = \sqrt{1 - \frac{|R_{xy}|}{|R_x||R_y|}} \tag{39}$$

$$= \sqrt{1 - \frac{\sigma_x^2\sigma_y^2 - \rho_{xy}^2\sigma_x^2\sigma_y^2}{\sigma_x^2\sigma_y^2}} \tag{40}$$

$$= \sqrt{1 - (1 - \rho_{xy}^2)} \tag{41}$$

$$= \sqrt{\rho_{xy}^2} \tag{42}$$

$$= |\rho_{xy}| \tag{43}$$

■

ACKNOWLEDGMENTS

We would like to thank Dr. Henry Medeiros for many valuable comments on the manuscript and Dr. James Theiler from Los Alamos National Laboratory for numerous suggestions, discussions and insights regarding the use of the ellipsoid log volume as a measure of detection accuracy.

[1] *Total correlation* is a related concept [36], generalizing the concept of mutual information to multiple random variables.

References

[1] J. B. Predd, S. R. Kulkarni, and H. V. Poor, "Distributed learning in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 23, no. 6, pp. 56–69, Jul. 2006.

[2] I. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

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

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

[5] J.-F. Chamberland and V. V. Veeravalli, "Wireless sensors in distributed detection applications," *IEEE Signal Processing Magazine*, vol. 25, no. 5, pp. 16–25, May 2007.

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

[7] 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, no. 4, pp. 448–463, Aug. 2008.

[8] 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.

[9] A. Wiesel and A. O. Hero, "Decomposable principal component analysis," *IEEE Trans. Signal Processing*, vol. 57, no. 11, pp. 4369–4377, Nov. 2009.

[10] M. Gastpar, P. Dragotti, and M. Vetterli, "The distributed Karhunen-Loeve transform," *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5177–5196, Dec. 2006.

[11] 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, Mar. 2010, pp. 2970–2973.

[12] 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.

[13] O. Roy and M. Vetterli, "Dimensionality reduction for distributed estimation in the infinite dimensional regime," *IEEE Trans. information theory*, vol. 54, no. 4, Apr. 2008.

[14] 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.

[15] 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.

[16] 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.

[17] J. Yoder, H. Medeiros, J. Park, and A. Kak, "Cluster-based distributed face tracking in camera networks," *IEEE Trans. Image Processing*, vol. 19, no. 10, pp. 2551–2563, Oct. 2010.

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

[19] 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, Oct. 2004, pp. 219–230.

[20] ——, "Mining anomalies using traffic feature distributions," in *Proc. 2005 Conf. Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM'05)*, vol. 34, Oct. 2005, pp. 217–228.

[21] 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.

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

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

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

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

[26] 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, no. 5, pp. 603–621, 2003.

[27] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, Jul. 2008.

[28] 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, no. 3, pp. 625–640, Mar. 2011.

[29] J. Theiler, "The incredible shrinking covariance estimator," *Proc. SPIE*, vol. 8391, p. 83910P, 2012.

[30] 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.

[31] 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. 2010.

[32] 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, no. 1, pp. 424–437, Jun. 2011.

[33] J.-F. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, Jan 1999.

[34] 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.

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

[36] S. Watanabe, "Information theoretical analysis of multivariate correlation," *IBM J. Research and Development*, no. 1, pp. 66–82, Jan. 1960.

[37] 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.

[38] 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.

[39] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

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

[41] ——, "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics*, vol. 11, no. 2, pp. 416–431, 1983.

[42] 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.

[43] 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.

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

[45] 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, Oct. 2011, pp. 1–8.

[46] 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.

[47] 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.

[48] S. Soderbergh, "Ocean's Eleven (film)," *Warner Bros.*, Dec. 2001.