The Sparse Matrix Transform for Covariance Estimation and Analysis of High Dimensional Signals

Guangzhi Cao*, *Member, IEEE,* Leonardo R. Bachega, and Charles A. Bouman, *Fellow, IEEE*

Abstract

Covariance estimation for high dimensional signals is a classically difficult problem in statistical signal analysis and machine learning. In this paper, we propose a maximum likelihood (ML) approach to covariance estimation, which employs a novel non-linear sparsity constraint. More specifically, the covariance is constrained to have an eigen decomposition which can be represented as a sparse matrix transform (SMT).

The SMT is formed by a product of pairwise coordinate rotations known as Givens rotations. Using this framework, the covariance can be efficiently estimated using greedy optimization of the log-likelihood function, and the number of Givens rotations can be efficiently computed using a cross-validation procedure. The resulting estimator is generally positive definite and well-conditioned, even when the sample size is limited. Experiments on a combination of simulated data, standard hyperspectral data, and face image sets show that the SMT-based covariance estimates are consistently more accurate than both traditional shrinkage estimates and recently proposed graphical lasso estimates for a variety of different classes and sample sizes.

An important property of the new covariance estimate is that it naturally yields a fast implementation of the estimated eigen-transformation using the SMT representation. In fact, the SMT can be viewed as a generalization of the classical fast Fourier transform (FFT) in that it uses "butterflies" to represent an orthonormal transform. However, unlike the FFT, the SMT can be used for fast eigen-signal analysis of general non-stationary signals.

EDICS Category: SMR-SMD, SMR-REP

G. Cao, 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: {gcao, lbachega, bouman}@purdue.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 the National Science Foundation under Contract CCR-0431024.

The Sparse Matrix Transform for Covariance Estimation and Analysis of High Dimensional Signals

I. Introduction

As the capacity to measure and collect data increases, high dimensional signals and systems have become much more prevalent. Medical imaging, remote sensing, internet communications, and financial data analysis are just a few examples of areas in which the dimensionality of signals is growing explosively, and leading to an unprecedented quantity of information and potential knowledge.

However, this growth also presents new challenges in the modeling and analysis of high dimensional signals (or data). In practice, the dimensionality of signals (p) often grows much faster than the number of available observations (n). The resulting "small n, large p" scenario [1] tends to break the basic assumptions of classical statistics and can cause conventional estimators to behave poorly. In fact, Donoho makes the very reasonable claim that $p \gg n$ is in fact the more generic case in learning and recognition problems [2]; so, this "curse of dimensionality" [3], [4] represents a very fundamental challenge for the future.

A closely related problem to the curse of dimensionality is the super-linear growth in computation that can occur with classical estimators as p grows large. For example, classical methods such as singular value decomposition (SVD) and eigen-analysis depend on the use of dense $p \times p$ transformations that can quickly become intractable to apply (or estimate) as the dimension grows. Therefore, the modeling and analysis of high dimensional signals pose a fundamental challenge not only from the perspective of inference, but also from the perspective of computation.

A fundamental step in the analysis of high dimensional signals is the estimation of the signal's covariance. In fact, an accurate estimate of signal covariance is often a key step in detection, classification, and modeling of high dimensional signals, such as images [5], [6]. However, covariance estimation for high dimensional signals is a classically difficult problem because the number of coefficients in the covariance grows as the dimension squared [7], [8]. In a typical application, one may measure n versions of a p dimensional vector; so if $n < p$, then the sample covariance matrix will be singular with $p - n$ eigenvalues equal to zero.

Over the years, a variety of techniques have been proposed for computing a nonsingular estimate of the covariance. For example, shrinkage and regularized covariance estimators are examples of such techniques. Shrinkage estimators are a widely used class of estimators which regularize the covariance matrix by shrinking it toward some positive definite target structures, such as the identity matrix or the diagonal of the sample covariance [9], [10], [11], [12], [13].

More recently, a number of methods have been proposed for regularizing the covariance estimate by constraining the estimate of the covariance or its inverse to be sparse [14], [15]. For example, the graphical lasso method enforces sparsity by imposing an L_1 norm constraint on the inverse covariance [15]. Theoretical justification for the lasso-type penalty on the inverse covariance matrix is provided in [16]. Banding or thresholding have also been used to obtain a sparse estimate of the covariance [14], [17]. Some other methods apply L_1 sparsity constraints to the eigen-transform itself, and are collectively referred to as sparse principal component analysis (SPCA) [18], [19], [20], [21].

In this paper, we propose a new approach to covariance estimation, which is based on constrained maximum likelihood (ML) estimation of the covariance from sample vectors [22], [23]. In particular, the covariance is constrained to be formed by an eigen-transformation that can be represented by a sparse matrix transform (SMT) [24]; and we define the SMT to be an orthonormal transformation formed by a product of pairwise coordinate rotations known as Givens rotations [25]. Using this framework, the covariance can be efficiently estimated using greedy maximization of the log likelihood function, and the number of Givens rotations can be efficiently computed using a cross-validation procedure. The estimator obtained using this method is generally positive definite and well-conditioned even when the sample size is limited.

Due to its flexible structure and data-dependent design, the SMT can be used to model behaviors of various kinds of natural signals. We will show that the SMT can be viewed as a generalization of both the classical fast Fourier transform (FFT) [26] and the orthonormal wavelet transforms. Since these frequency transforms are commonly used to decorrelate and therefore model stationary random processes, the SMT inherits this valuable property. We will also demonstrate that autoregressive (AR) and moving average (MA) random processes can be accurately modeled by a low-order SMT. However, the SMT is more expressive than conventional frequency transforms because it can accurately model high dimensional natural signals that are not stationary, such as hyperspectral data measurements. In addition, it is shown that the SMT covariance estimate is invariant to permutations of the data coordinates; a property that is not shared by models based on the FFT or wavelet transforms [16]. Nonetheless, the SMT model does impose a substantial sparsity constraint through a restriction in the number of Givens rotations. When

this sparsity constraint holds for real data, then the SMT model can substantially improve the accuracy of covariance estimates; but conversely if the eigenspace of the random process has no structure, then the SMT model provides no advantage [27].

The fast transformation algorithms resulting from SMT covariance estimation are perhaps just as important as the improved statistical power of the method. Conventional PCA analysis requires multiplication by a $p \times p$ dense eigen-transformation to de-correlate and model signals. This requires p^2 operations, which is typically not practical for high dimensional signals such as images. Alternatively, the eigen-transformation resulting from the proposed method is constrained to be an SMT, so application of the de-correlating transform is typically linear in p.[1]

In order to validate our model, we perform experiments using simulated data, standard hyperspectral image data, and face image data sets. We compare against both traditional shrinkage estimates and recently proposed graphical lasso estimates. Our experiments show that, for these examples, the SMT-based covariance estimates are consistently more accurate for a variety of different classes and sample sizes. Moreover, the method seems to work particularly well for estimating small eigenvalues and their associated eigenvectors; and the cross-validation procedure used to estimate the SMT model order can be implemented with a modest increase in computation.

II. Covariance Estimation for High Dimensional Signals

In the general case, we observe a set of n vectors, $y_1, y_2, \cdots, y_n$, where each vector, y_i, is p dimensional. Without loss of generality, we assume y_i has zero mean. We can represent this data as the following $p \times n$ matrix

$$Y = [y_1, y_2, \cdots, y_n] \ . \tag{1}$$

If the vectors y_i are identically distributed, then the sample covariance is given by

$$S = \frac{1}{n} Y Y^t \ , \tag{2}$$

and S is an unbiased estimate of the true covariance matrix with $R = \mathrm{E}\left[y_i y_i^t\right] = \mathrm{E}[S]$.

While S is an unbiased estimate of R, it is also singular when $n < p$. This is a serious deficiency since as the dimension p grows, the number of vectors needed to estimate R also grows. In practical applications, n may be much smaller than p which means that most of the eigenvalues of R are erroneously estimated as zero.

[1]In our experiments, the SMT requires 1 to 5 rotation per coordinate, depending on the estimated order of the model.

A variety of methods have been proposed to regularize the estimate of R so that it is not singular. Shrinkage estimators are a widely used class of estimators which regularize the covariance matrix by shrinking it toward some target structures [9], [10], [11]. Shrinkage estimators generally have the form $\hat{R} = \alpha D + (1-\alpha)S$, where D is some positive definite matrix. Some popular choices for D are the identity matrix (or its scaled version) [10], [11] and the diagonal entries of S, $\mathrm{diag}(S)$ [10], [13]. In both cases, the shrinkage intensity α can be estimated using cross-validation or boot-strap methods.

Recently, a number of methods have been proposed for regularizing the estimate by making either the covariance or its inverse sparse [14], [15]. For example, the graphical lasso method enforces sparsity by imposing an L_1 norm constraint on the inverse covariance [15]. Banding or thresholding can also be used to obtain a sparse estimate of the covariance [14], [17].

A. Maximum Likelihood Covariance Estimation

Our approach will be to compute a constrained maximum likelihood (ML) estimate of the covariance R, under the modeling assumption that eigenvectors of R may be represented as a sparse matrix transform (SMT) [22], [24]. To do this, we first decompose R as

$$R = E\Lambda E^t \ , \tag{3}$$

where E is the orthonormal matrix of eigenvectors (also referred to as the eigen-transformation) and Λ is the diagonal matrix of eigenvalues. Then we will estimate the covariance by maximizing the likelihood of the data Y subject to the constraint that E is an SMT of order K (to be defined below in Section II-B). By varying the order, K, of the SMT, we may then reduce or increase the regularizing constraint on the covariance.

If we assume that the columns of Y are independent and identically distributed Gaussian random vectors with mean zero and positive-definite covariance R, then the likelihood of Y given R is given by

$$P_R(Y) = \frac{1}{(2\pi)^{\frac{np}{2}}} |R|^{-\frac{n}{2}} \exp\left\{ -\frac{1}{2}\mathrm{tr}\{Y^t R^{-1} Y\} \right\} \ . \tag{4}$$

The log-likelihood of Y is then given by (see Appendix A)

$$\log P_{(E,\Lambda)}(Y) = -\frac{n}{2}\mathrm{tr}\{\mathrm{diag}(E^t S E)\Lambda^{-1}\} - \frac{n}{2}\log|\Lambda| - \frac{np}{2}\log(2\pi) \ . \tag{5}$$

Jointly maximizing the likelihood with respect to E and Λ then results in the ML estimates given by

(see Appendix A)

$$\hat{E} = \arg\min_{E\in\Omega} \left\{ \left| \mathrm{diag}(E^t S E) \right| \right\} \quad (6)$$

$$\hat{\Lambda} = \mathrm{diag}(\hat{E}^t S \hat{E}) \ , \quad (7)$$

where Ω is the set of allowed orthonormal transforms, and $|\cdot|$ denotes the determinant of a matrix. Then $\hat{R} = \hat{E}\hat{\Lambda}\hat{E}^t$ is the ML estimate of the covariance matrix R. So we may compute the ML estimate by first solving the constrained optimization of (6), and then computing the eigenvalue estimates from (7).

An interesting special case occurs when S has full rank and Ω is the set of all orthonormal transforms. In this case, (6) and (7) are solved by selecting E and Λ as the eigenvector matrix and eigenvalue matrix of S, respectively (see Appendix B). So this leads to the well known result that when S is non-singular, then the ML estimate of the covariance is given by the sample covariance, i.e. $\hat{R} = S$. However, when S is singular and Ω is the set of all orthonormal transforms, then the log-likelihood is unbounded, with a subset of the estimated eigenvalues tending toward zero.

B. ML Estimation of Eigen-Transformation Using the SMT Model

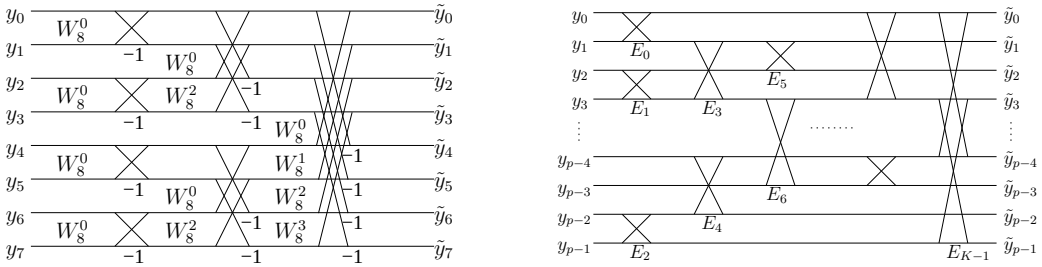
The ML estimate of E can be improved if the feasible set of eigen-transformations, Ω, can be constrained to a subset of all possible orthonormal transforms. By constraining Ω, we effectively regularize the ML estimate by imposing a model. However, as with any model-based approach, the key is to select a feasible set, Ω, which is as small as possible while still accurately modeling the behavior of real data.

Our approach is to select Ω to be the set of all orthonormal transforms that can be represented as an SMT of order K [22], [24]. More specifically, a matrix E is an SMT of order K if it can be written as a product of K sparse orthornormal matrices, so that

$$E = \prod_{k=1}^{K} E_k = E_1 E_2 \cdots E_K \ , \quad (8)$$

where each sparse matrix, E_k, is a Givens rotation operating on a pair of coordinate indices (i_k, j_k) [25]. More specifically, each Givens rotation E_k is an orthonormal rotation in the plane of the two coordinates, i_k and j_k, with the form

$$E_k = I + \Theta(i_k, j_k, \theta_k) \ , \quad (9)$$



(a) FFT (b) SMT

Fig. 1. (a) 8-point FFT. (b) An example of an SMT implementation of $\tilde{y} = Ey$. The SMT can be viewed as a generalization of both the FFT and the orthonormal wavelet transform. Notice that, unlike the FFT and the wavelet transform, the SMT's "butterflies" are not constrained in their ordering or rotation angles.

where $\Theta(i_k, j_k, \theta_k)$ is defined as

$$[\Theta]_{ij} = \begin{cases} \cos(\theta_k) - 1 & \text{if } i = j = i_k \text{ or } i = j = j_k \\ \sin(\theta_k) & \text{if } i = i_k \text{ and } j = j_k \\ -\sin(\theta_k) & \text{if } i = j_k \text{ and } j = i_k \\ 0 & \text{otherwise} \end{cases} . \tag{10}$$

Figure 1(b) shows the flow diagram for the application of an SMT to a data vector y. Notice that each 2D rotation, E_k, plays a role analogous to a "butterfly" used in a traditional fast Fourier transform (FFT) [26] in Fig. 1(a). However, unlike an FFT, the organization of the butterflies in an SMT is unstructured; so each butterfly can have an arbitrary rotation angle θ_k and can operate on pairs of coordinates in any order. Both the arrangement of butterflies and their rotations angles in the SMT can be adjusted for the specific characteristics of the data. This more general structure allows the SMT to implement a larger set of orthonormal transformations, and can be viewed as a generalization of the FFT.

In fact, the SMT can also be used to represent any orthonormal wavelet transform because, using the theory of paraunitary wavelets, orthonormal wavelets can be represented as a product of Givens rotations and delays [28], [29]. The SMT also includes the recently proposed class of treelets [30], which uses less than p Givens rotations to form a hierarchical orthonormal transform that is reminiscent of wavelets in their structure. More generally, when $K = \binom{p}{2}$, the SMT can be used to exactly represent any $p \times p$ orthonormal transformation (see Appendix C). Therefore, by varying the number of Givens rotations K, we can increase or decrease the set of orthonormal transforms that the SMT can represent.

Using the SMT model constraint, the ML estimate of E is given by

$$\hat{E} = \arg \min_{E=\prod_{k=1}^{K} E_k} \left|\mathrm{diag}(E^t S E)\right| \ , \tag{11}$$

where K is the SMT model order. Unfortunately, evaluating the constrained ML estimate of (11) requires the solution of an optimization problem with a non-convex constraint. So evaluation of the globally optimal solutions is difficult. Therefore, our approach will use greedy minimization to compute a locally optimal solution to (11). The greedy minimization approach works by selecting each new butterfly E_k to minimize the cost, while fixing the previous butterflies, E_l for $l < k$.

This greedy optimization algorithm can be implemented with the following simple recursive procedure. We start by setting $S_1 = S$ to be the sample covariance, and initialize $k = 1$. Then we apply the following two steps for $k = 1$ to K

$$\hat{E}_k = \arg \min_{E_k} \left|\mathrm{diag}\left(E_k^t S_k E_k\right)\right| \tag{12}$$

$$S_{k+1} = \hat{E}_k^t S_k \hat{E}_k \ . \tag{13}$$

The resulting values of $\hat{E}_k$ are the butterflies of the SMT.

The problem remains of how to compute the solution to (12). In fact, this can be done easily by first determining the two coordinates, i_k and j_k, that are most correlated,

$$(i_k, j_k) \leftarrow \arg \min_{(i,j)} \left(1 - \frac{[S_k]_{ij}^2}{[S_k]_{ii}[S_k]_{jj}}\right) \ . \tag{14}$$

It can be shown that this coordinate pair, (i_k, j_k), can most reduce the cost in (12) among all possible coordinate pairs (see Appendix D). Once i_k and j_k are determined, we apply the Givens rotation $\hat{E}_k$ to minimize the cost in (12), which is given by (see Appendix D)

$$\hat{E}_k = I + \Theta(i_k, j_k, \theta_k) \ , \tag{15}$$

where [2]

$$\theta_k = \frac{1}{2}\mathrm{atan}(-2[S_k]_{i_k j_k}, [S_k]_{i_k i_k} - [S_k]_{j_k j_k}) \ . \tag{16}$$

[2]Here we use $\mathrm{atan}(y, x) = \mathrm{atan}(y/x)$ when y and x are positive. By using the four quadrant inverse tangent function, we intentionally put the decorrelated components in a descending order along the diagonal.

By iterating (12) and (13) K times, we obtain the constrained ML estimate of E and Λ given by

$$\hat{E} = \prod_{k=1}^{K} \hat{E}_k \tag{17}$$

$$\hat{\Lambda} = \operatorname{diag}\left(S_{K+1}\right) \ . \tag{18}$$

Notice that the resulting SMT covariance estimate $\hat{R} = \hat{E}\hat{\Lambda}\hat{E}^t$ is always non-negative definite.[3] In fact, as we find in various numerical experiments that this greedy algorithm consistently results in a positive definite covariance estimate.

C. Model Order Estimation Using Cross-Validation

The model order, K, can be efficiently determined using a simple cross-validation procedure [1]. Let $Y^{(1)}$ denote a $p \times n_1$ matrix containing a randomly selected subset of column vectors from Y. And let $\bar{Y}^{(1)}$ be a $p \times (n - n_1)$ matrix containing the complimentary set of data vectors that will be used for training. From these data sets, we form the two sample covariance matrices, $S^{(1)} = \frac{1}{n_1} Y^{(1)} \left[Y^{(1)}\right]^t$ and $\bar{S}^{(1)} = \frac{1}{n-n_1} \bar{Y}^{(1)} \left[\bar{Y}^{(1)}\right]^t$. Then the log-likelihood of $Y^{(1)}$ given the order-k SMT covariance estimate can be computed by iterating the following steps starting with $k = 1$,

$$\hat{E}_k = \arg\min_{E_k} \left| \operatorname{diag}\left(E_k^t \bar{S}_k^{(1)} E_k\right) \right| \tag{19}$$

$$\hat{\Lambda}_k = \operatorname{diag}\left(\hat{E}_k^t \bar{S}_k^{(1)} \hat{E}_k\right) \tag{20}$$

$$\bar{S}_{k+1}^{(1)} = \hat{E}_k^t \bar{S}_k^{(1)} \hat{E}_k \tag{21}$$

$$S_{k+1}^{(1)} = \hat{E}_k^t S_k^{(1)} \hat{E}_k \tag{22}$$

$$L(k|1) = -\frac{1}{2}\operatorname{tr}\left\{\operatorname{diag}(S_{k+1})\hat{\Lambda}_k^{-1}\right\} - \frac{1}{2}\log\left|\hat{\Lambda}_k\right| - \frac{p}{2}\log(2\pi) \ , \tag{23}$$

where $L(k|1)$ is the cross-validated log-likelihood of the model with order k using the 1^{st} data subset. The process is then repeated using t non-intersecting subsets of the data to yield the average cross-validated log-likelihood of

$$L(k) = \frac{1}{t}\sum_{i=1}^{t} \log L(k|i) \ . \tag{24}$$

From this function, the model order can be estimated by finding $K^* = \arg\max_k L(k)$ where the search can be stopped when $L(k)$ begins to decrease. Once K^* is determined, the SMT covariance estimate is re-computed using all the data and the estimated model order.

[3]We know $\hat{\Lambda} = \operatorname{diag}\left(\hat{E}^t S \hat{E}\right) = \frac{1}{n}\operatorname{diag}\left((\hat{E}^t Y)(\hat{E}^t Y)^t\right)$, thus $\hat{\Lambda}_{ii} \geq 0$.

Notice that the log-likelihood of (23) is evaluated on-the-fly as the greedy SMT design algorithm proceeds. This iterative process dramatically reduces the computation by eliminating the need to compute the cross-validated log-likelihood K times.[4] However, since the $L(k|i)$ must be computed for t values of i, the implementation of this cross-validation procedure does increase the computation of SMT estimation by a factor of t, which in our examples is typically a small integer value such as 3.

D. SMT Covariance Estimation with Minimum Eigenvalue Constraint

In some circumstances, it may be known that the eigenvalues of the covariance are bounded below by some minimum value. For example, this may occur when fixed additive noise or quantization error lower bounds the eigenvalues of the covariance. The SMT covariance estimate can be extended to satisfy this constraint. To do this, we use a regularized version of the sample covariance given by $S + \sigma I$, where $\sigma > 0$ is the required minimum value of the eigenvalue estimate. In this case, the maximum likelihood estimates of the eigenvectors and eigenvalues are given by

$$\begin{aligned} \hat{E} &= \arg\min_{E\in\Omega} \left\{ \left| \mathrm{diag}\left(E^t (S+\sigma I) E \right) \right| \right\} \qquad (25) \\ &= \arg\min_{E\in\Omega} \left\{ \left| \mathrm{diag}\left(E^t S E \right) + \sigma I \right| \right\} \\ \hat{\Lambda} &= \mathrm{diag}\left(\hat{E}^t S E \right) + \sigma I \ . \qquad (26) \end{aligned}$$

From the form of (26), it can be seen that all the resulting SMT eigenvalue estimates are then constrained to be larger than σ. Also, notice that if S has full rank and Ω is the set of all orthonormal transforms, then (25) can still be solved by letting E be the eigenvector matrix of S (see Appendix B). Under the constraint that eigenvector matrix E is an SMT of order K, then the greedy solution of (25) results in the following selection criterion at each step k

$$(i_k, j_k) \leftarrow \arg\min_{(i,j)} \left(1 - \frac{[S_k]_{ij}^2}{([S_k]_{ii} + \sigma) \cdot ([S_k]_{jj} + \sigma)} \right) \ . \qquad (27)$$

The selection criterion of (27) can also be used to stabilize the selection when diagonal entries of S go to zero. This can happen if either columns of Y are identically zero, or pairs of columns of Y are linearly dependent. However, for all numerical experiments in this paper we set $\sigma = 0$.

[4]We will see that K tends to grow in proportion to p, so a reduction in computation by a factor of K can be quite dramatic.

III. SMT Design Algorithms and Computational Complexity

Figure 2(a) shows a baseline algorithm to compute the SMT covariance estimate, which is derived directly from (12) – (18) in Section II. In this algorithm, each of the K iterations of the loop (lines 2–7) computes one Givens rotation in the plane spanned by the coordinates (i_k, j_k) with maximum correlation coefficient. In the end, the estimates for Λ and E are obtained (lines 8 and 9, respectively). The computation of each iteration of the greedy algorithm tends to be dominated by the time required to find the best coordinate pair, (i_k, j_k). A naive search for the best coordinate pair requires $O(p^2)$ time. Therefore, design of the entire order-K SMT using the baseline algorithm requires $O(Kp^2)$ time. Moreover, the baseline algorithm also requires the explicit storage of the $p \times p$ sample covariance matrix S, therefore resulting in a memory requirement of $O(p^2)$.

Fortunately, it is possible to reduce the computational complexity and memory requirements of the SMT design algorithm, as we discuss below.[5]

A. Fast Computation of the SMT Design

The search for the most correlated coordinate pair can be made much more efficient, as shown in Algorithm 2 of Fig. 2(b). The algorithm can be summarized as follows: (i) An initial scan over all the elements of each row of the matrix S is performed, and for each row i we store the maximum correlation coefficient $maxC_i$ and the index j associated with with $maxC_i$; (ii) At each of the K steps of the SMT design, the vector $maxC$ with only p elements is scanned instead of the whole matrix S; (iii) only the i_k-th and j_k-th rows and columns of S are modified in line 11, thus requiring a corresponding fraction of the elements in $maxC$ to be updated.

In the worst-case, every row may point to i_k or j_k as the row for which it is most correlated. In this case, the elements of all the rows must be re-scanned and their entries in the vector $maxC$ must be updated. In this worst-case, Algorithm 2 in Fig. 2(b) runs in $O(Kp^2)$ time. However, the empirical results shown in Fig. 3 suggest that the worst-case scenario does not happen in practice, e.g. for the datasets investigated in this paper. Notice that by keeping the maximum of each row in the vector $maxC$, the SMT design implementation now runs in time close to linear in p (as p grows large) while the baseline algorithm runs in $O(p^2)$ time.

In principle, the low computational complexity observed in practice allows this algorithm to be deployed in cases when the data dimension, p, is large. However, this algorithm still requires the storage of the

[5]We emphasize that regardless of how the SMT is designed, the application of the SMT to data decorrelation is always $O(K)$, and is therefore typically very fast.

TABLE I
COMPARISON OF THE SMT DESIGN ALGORITHMS. HERE, p IS DIMENSION OF THE DATA VECTORS, n IS NUMBER OF THE SAMPLES, AND K IS NUMBER OF GIVENS ROTATIONS IN THE SMT ESTIMATOR.

algorithm	storage required	initialization	computation/step		total computation(K steps)	
			empirical	worst-case	empirical	worst-case
Algorithm 1 (baseline)	$O(p^2)$	$O(np^2)$	$O(p^2)$	$O(p^2)$	$O(Kp^2)$	$O(Kp^2)$
Algorithm 2	$O(p^2)$	$O(np^2)$	$O(p)$	$O(p^2)$	$O(Kp)$	$O(Kp^2)$
Algorithm 3	$O(np)$	$O(np^2)$	$O(np)$	$O(np^2)$	$O(nKp)$	$O(nKp^2)$

$p \times p$ sample covariance matrix S in memory. This is impractical when p is large, a problem that we address with an improvement on Algorithm 2 as described next.

B. Direct SMT Design Based on Data Samples Y Instead of S

The storage of the sample covariance matrix S can be prohibitive when p is large, limiting the application of Algorithm 2 in Fig. 2(b). For instance, if one wishes to apply the SMT to the eigenface problem using faces of 100×100 pixels, the associated sample covariance S requires a memory storage of 762MB[6], imposing a limitation on the hardware that could be used to compute such a task.

In order to overcome this problem, Algorithm 3 in Fig. 2(c) presents an approach which is conceptually equivalent to Algorithm 2 in Fig. 2(b), but instead operates directly on the $p \times n$ data matrix, Y, rather than storing the sample covariance S. The primary difference in terms of complexity is that Algorithm 3 requires that the values of S_{ii}, S_{ij} and S_{jj} to be computed on-the-fly, each requiring an $O(n)$ scalar product between two rows of Y.

Table I summarizes all the computational complexities and the storage requirements of the three algorithms for SMT design. Notice that the empirical complexity is reduced from $O(Kp^2)$ in Algorithm 1 (baseline) to $O(Kp)$ in Algorithm 2, and the amount of memory required is reduced from $O(p^2)$ in Algorithm 2 to $O(np)$ in Algorithm 3.

IV. PROPERTIES OF SMT COVARIANCE ESTIMATOR AND ITS EXTENSIONS

A. Properties of SMT Covariance Estimator

Let $\hat{R} = \hat{E}\hat{\Lambda}\hat{E}^t$ be the SMT covariance estimator of the p dimensional data vectors Y as described in Section II. The SMT covariance estimator has the following properties.

[6]Assuming each element is of double float precision, requiring 8 bytes.

$S \leftarrow YY^t/n$
for $1 \leq k \leq K$ **do**
  $(i_k, j_k) \leftarrow \arg \max_{(i,j):i<j} \left\{ \frac{S_{ij}^2}{S_{ii} \cdot S_{jj}} \right\}$
  $\theta_k \leftarrow \frac{1}{2} \text{atan}(-2S_{i_k,j_k}, S_{i_k i_k} - S_{j_k j_k})$
  $E_k \leftarrow I + \Theta(i_k, j_k, \theta_k)$
  $S \leftarrow E_k^t S E_k$
end for
$\Lambda \leftarrow \text{diag}\,(S)$
$E \leftarrow \prod_{k=1}^{K} E_k$

(a)

$S \leftarrow YY^t/n$
for $1 \leq i \leq p$ **do**
  $MaxJ(i) \leftarrow \arg \max_{j:i<j} \left\{ \frac{S_{ij}^2}{S_{ii} \cdot S_{jj}} \right\}$
  $MaxC(i) \leftarrow \max_{j:i<j} \left\{ \frac{S_{ij}^2}{S_{ii} \cdot S_{jj}} \right\}$
end for
for $1 \leq k \leq K$ **do**
  $i_k \leftarrow \arg\max_i MaxC(i)$
  $j_k \leftarrow MaxJ(i_k)$
  $\theta_k \leftarrow \frac{1}{2} \text{atan}(-2S_{i_k,j_k}, S_{i_k i_k} - S_{j_k j_k})$
  $E_k \leftarrow I + \Theta(i_k, j_k, \theta_k)$
  $S \leftarrow E_k^t S E_k$
  for $1 \leq i \leq p$ **do**
    if $(i = i_k \text{ or } i = j_k)$ or $(MaxJ(i) = i_k \text{ or } MaxJ(i) = j_k)$ **then**
      $MaxJ(i) \leftarrow \arg \max_{j:i<j} \left\{ \frac{S_{ij}^2}{S_{ii} \cdot S_{jj}} \right\}$
      $MaxC(i) \leftarrow \max_{j:i<j} \left\{ \frac{S_{ij}^2}{S_{ii} \cdot S_{jj}} \right\}$
    end if
  end for
end for
$\Lambda \leftarrow \text{diag}\,(S)$
$E \leftarrow \prod_{k=1}^{K} E_k$

(b)

for $1 \leq i \leq p$ **do**
  $MaxJ(i) \leftarrow \arg \max_{j:j<i} \left\{ \frac{|Y(i,:)^t Y(j,:)|}{\|Y(i,:)\| \|Y(j,:)\|} \right\}$
  $MaxC(i) \leftarrow \max_{j:i<j} \left\{ \frac{|Y(i,:)^t Y(j,:)|}{\|Y(i,:)\| \|Y(j,:)\|} \right\}$
end for
for $1 \leq k \leq K$ **do**
  $i_k \leftarrow \arg\max_i MaxC(i)$
  $j_k \leftarrow MaxJ(i_k)$
  $\theta_k \leftarrow \frac{1}{2} \text{atan}(-2S_{i_k,j_k}, S_{i_k i_k} - S_{j_k j_k})$
  $E_k \leftarrow I + \Theta(i_k, j_k, \theta_k)$
  $Y \leftarrow E_k^t Y$
  for $1 \leq i \leq p$ **do**
    if $(i = i_k \text{ or } i = j_k)$ or $(MaxJ(i) = i_k \text{ or } MaxJ(i) = j_k)$ **then**
      $MaxJ(i) \leftarrow \arg \max_{j:i<j} \left\{ \frac{|Y(i,:)^t Y(j,:)|}{\|Y(i,:)\| \|Y(j,:)\|} \right\}$
      $MaxC(i) \leftarrow \max_{j:i<j} \left\{ \frac{|Y(i,:)^t Y(j,:)|}{\|Y(i,:)\| \|Y(j,:)\|} \right\}$
    end if
  end for
end for
$\Lambda \leftarrow \text{diag}(Y^t Y)/n$
$E \leftarrow \prod_{k=1}^{K} E_k$

(c)

Fig. 2. Pseudo-code of the greedy algorithms for the SMT covariance estimation. (a) Algorithm 1: baseline implementation of the SMT design; (b) Algorithm 2: optimized SMT design algorithm that runs in $O(Kp)$ empirically; (c) Algorithm 3: optimized SMT design algorithm that operates directly on the data samples, not requiring the matrix S to be explicitly stored in memory.

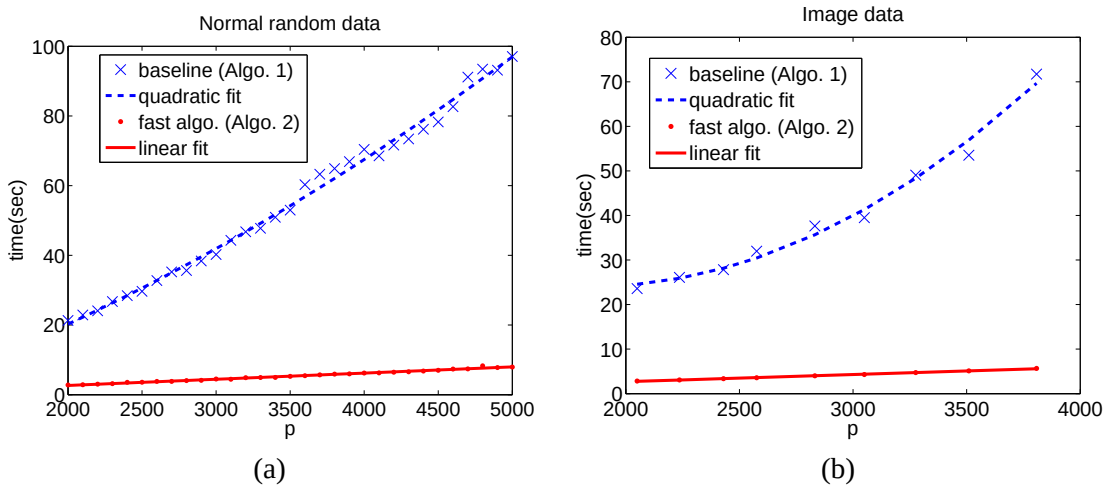


Fig. 3. Time to compute the SMT covariance estimate for (a) random data from the normal $N(0,1)$ distribution; (b) dataset with 40 face images from the ORL data set. Here, the SMT model order is fixed to $K = 100$ for each experiment. A polynomial fit using least-squares was performed for each algorithm. Notice that the time taken by the fast algorithm (Algo. 2) that keeps track of the maximum correlation of each row of the sample covariance matrix S remains linear as p grows large, while the time taken by the baseline algorithm (Algo. 1) of the SMT algorithm is verified to vary with p^2.

Property 1: The SMT covariance estimate is permutation invariant. More specifically, if $\hat{R}$ is the unique order-K SMT covariance estimate of the data Y, then for any permutation matrix P, the order-K SMT covariance estimate of PY is given by $P\hat{R}P^t$.

Uniqueness of $\hat{R}$ means that (14) is assumed to have a unique minimum at each step $k \leq K$. The proof of Property 1 is given in Appendix E. This property shows that the SMT covariance estimator does not depend on the ordering of the data. Therefore, it can potentially be used to model data sets whose ordering does not have explicit meaning, such as text data, financial data, and data from distributed sensor networks.

Property 2: If the SMT covariance estimate $\hat{R}$ is of model order K, then the resulting eigen-transformation can be computed in $O(K)$ time.

The eigen-transform resulting from SMT covariance estimation can be efficiently computed by applying the K Givens rotations in sequence

$$\hat{E}^t y = \prod_{k=K}^{1} \hat{E}_k^t y \ . \tag{28}$$

Every Givens rotation E_k requires at most 4 multiplies (actually only 2 multiplies with a more efficient implementation [24]). Therefore, the SMT eigen-transformation has a complexity of $O(K)$. As we find in our experiments, usually K is a small multiple of p. As a comparison, a general dense eigen-transformation

has a complexity of $O(p^2)$. The computational advantage of the SMT is due to its sparse structure, which makes it attractive for applications using high dimensional data such as eigen-image analysis.

Property 3: The inverse covariance matrix estimator $\hat{R}^{-1}$ has the same SMT structure as $\hat{R}$.

In many applications, it is more interesting to know the inverse covariance rather than the covariance itself. Fortunately, once the SMT covariance estimate $\hat{R}$ is obtained, its inverse $\hat{R}^{-1}$ is immediately known as

$$\hat{R}^{-1} = \hat{E}\hat{\Lambda}^{-1}\hat{E}^t = \left(\prod_{k=1}^{K}\hat{E}_k\right)\hat{\Lambda}^{-1}\left(\prod_{k=K}^{1}\hat{E}_k^t\right) \ . \tag{29}$$

Note that the inverse covariance estimate $\hat{R}^{-1}$ has the same SMT structure as the covariance estimate $\hat{R}$.

B. SMT Shrinkage Estimator

In some cases, the accuracy of the SMT covariance estimator can be improved by shrinking it towards the sample covariance. Let $\hat{R}$ be the SMT covariance estimator. Then the SMT shrinkage estimator (SMT-S) can be obtained as

$$\hat{R}_s = \alpha\hat{R} + (1-\alpha)S \ , \tag{30}$$

where α is the shrinkage intensity, and $\hat{R} = \hat{E}\hat{\Lambda}\hat{E}^t$ is the SMT covariance estimate. The value of α can be determined using leave-one-out likelihood (LOOL) cross-validation [10], which can be done efficiently in the SMT transformed domain. Let $S^{(i)}$ be the sample covariance excluding y_i,

$$S^{(i)} = \frac{1}{n-1}\sum_{\substack{j=1\\ j\neq i}}^{n} y_j y_j^t = \frac{n}{n-1}S - \frac{1}{n-1}y_i y_i^t \ . \tag{31}$$

Then we can define $\hat{R}_{s|i} = \alpha\hat{R} + (1-\alpha)S^{(i)}$ to be the corresponding SMT-S covariance estimator. Notice that

$$P(y_i \mid \hat{R}_{s|i}) = P(\hat{E}^t y_i \mid \hat{E}^t\hat{R}_{s|i}\hat{E}) = P(\tilde{y}_i \mid \alpha\hat{\Lambda} + (1-\alpha)\tilde{S}^{(i)}) \ , \tag{32}$$

where $\tilde{y}_i = \hat{E}^t y_i$ and $\tilde{S}^{(i)} = \hat{E}^t S^{(i)}\hat{E}$.

Define

$$G = \alpha\hat{\Lambda} + (1-\alpha)\frac{n}{n-1}\tilde{S} \tag{33}$$

where $\tilde{S} = \hat{E}^t S \hat{E}$, then the log-likelihood of y_i given $\hat{R}_{s|i}$ in (32) can be efficiently computed as

$$\log P\left(y_i \mid \hat{R}_{s|i}\right) = \log P\left(\tilde{y}_i \mid G - \beta \tilde{y}_i \tilde{y}_i^t\right) \quad (34)$$

$$= -\frac{1}{2}\log\left(\mid G \mid (1-\beta d_i)\right) - \frac{1}{2}\left(\frac{d_i}{1-\beta d_i}\right) - \frac{p}{2}\log\left(2\pi\right) \ , \quad (35)$$

where $\beta = \frac{1-\alpha}{n-1}$ and $d_i = \tilde{y}_i^t G^{-1} \tilde{y}_i$. Notice that for all the y_i, we only need to compute $\mid G \mid$ once for a given value of α. So this saves a large amount of computation. The value of α that leads to the maximum average LOOL is chosen as the final shrinkage intensity

$$\alpha^* = \arg \max_{\alpha \in (0,1]} \frac{1}{n} \sum_{i=1}^{n} \log P\left(y_i \mid \hat{R}_{s|i}\right) \ . \quad (36)$$

Once the shrinkage intensity α^* is determined, the SMT-S covariance estimator $\hat{R}_s$ is computed using all the samples and the estimated shrinkage intensity, α^*.

V. Experimental Results

In this section, we compare the performance of the proposed method to commonly used shrinkage estimators, and the recently proposed graphical lasso estimator, and in some cases we compare to sparse PCA. We do this comparison using simulated data, standard hyperspectral remotely sensed data and face image sets as examples of high dimensional signals.

A. Review of Alternative Estimators

Shrinkage estimators are a widely used class of estimators. A popular choice of the shrinkage target is the diagonal of S [10], [13]. In this case, the estimator (referred to as the shrinkage estimator hereafter) is given by

$$\hat{R} = \alpha \mathrm{diag}\left(S\right) + \left(1-\alpha\right) S \ . \quad (37)$$

Similar to the SMT-S estimator, an efficient algorithm for the leave-one-out likelihood (LOOL) cross-validation has been suggested for choosing the shrinkage intensity α in [10].

Another popular shrinkage target is the identity matrix as in [12]. In this case, the estimator is given by

$$\hat{R} = \alpha \frac{\mathrm{tr}\left(S\right)}{p} I + \left(1-\alpha\right) S \ . \quad (38)$$

The analytical form of α that minimizes a quadratic loss between the estimator and the true covariance is derived by Ledoit and Wolf in [12]. We used the publically available Matlab code for this estimator

(referred to as the L-W estimator hereafter) [31]. It is easy to see that the L-W estimator only regularizes the eigenvalues of the sample covariance, while keeping the eigenvectors unchanged.

An alternative estimator is the graphic lasso (glasso) estimator recently proposed in [15] which is an L_1-regularized maximum likelihood estimate, such that

$$\hat{R} = \arg\max_{R \in \Psi} \left\{ [\log P(Y \mid R)] - \rho \parallel R^{-1} \parallel_1 \right\} \ , \tag{39}$$

where Ψ denotes the set of $p \times p$ positive definite matrices and ρ is a regularization parameter. Glasso enforces sparsity by imposing an L_1 norm constraint on the inverse covariance, and is a good representative of the general class of L_1 based methods. We used the implementation of glasso in the R software that is publicly available without penalizing the diagonal (i.e. "penalize.diagonal=FALSE") [32]. The parameter ρ is chosen using cross-validation that maximizes the average log-likelihood of the left-out subset. The glasso estimate in (39) has a computational complexity of $O(ip^3)$ for a given value of ρ, where i is the number of iterations in glasso. Cross-validation for ρ requires (39) to be solved for every different value ρ, which is is computationally very expensive. We compared the SMT estimators with glasso only for real data cases.

B. SMT Covariance Estimation for Simulated Data

1) Model Order Estimation: The best scenario for the SMT covariance estimator occurs when the true covariance matrix has an eigenvector matrix which is an SMT of order K where K is relatively small ($K \ll p(p-1)/2$). If this is the case, it is important to demonstrate that the SMT covariance estimator is able to recover an appropriate model order and covariance estimate with limited sample data. To do this, we first generate a covariance $R = \left(\prod_{k=1}^{K} E_k\right) \Lambda \left(\prod_{k=K}^{1} E_k^t\right)$ where every E_k is a Givens rotation with randomly generated coordinate pair (i_k, j_k) and rotation angle θ_k, and eigenvalues given by $\Lambda_{ii} = i^{-2}$. More specifically, for each E_k both the coordinate pair (i_k, j_k) and the rotation angle θ_k are independent and have uniform distribution. Then Gaussian samples are generated with zero mean and covariance R and used for covariance estimation. In simulation, we used $p = 200$, $n = \{50, 100, 200\}$ and $K \in [50, 800]$. The experiment was repeated 10 times with re-generated covariance R each time. The performance measure is based on the average of all the runs.

Figure 4 shows the results of the model order estimation using 3-fold cross-validation (i.e. $t = 3$ in (24)). As expected, the estimated model order is a function of both the true model order K and the sample size n. We compared the estimated covariance for each method to the true covariance using the Kullback-Leibler (KL) distance (Appendix F) [33]. The KL distance is a measure of the error between

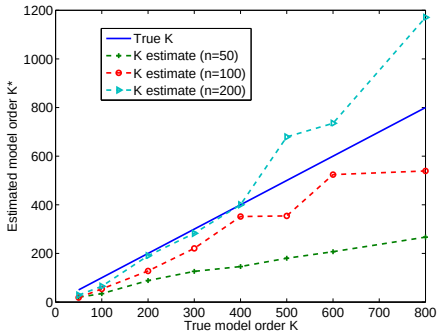


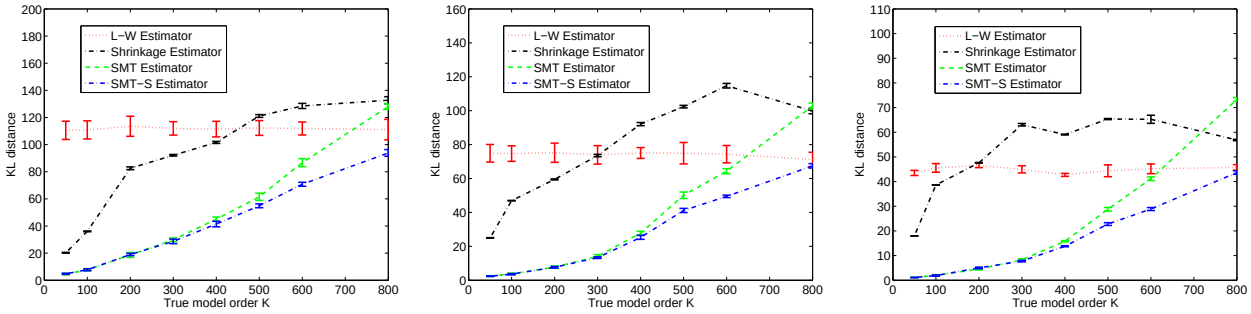
Fig. 4. Estimated SMT model order when the true covariances consist of K randomly generated Givens rotations, $p = 200$.

the estimated and true distributions. Figure 5 shows the KL distances of different estimators as a function the sample number n. The error bars indicate the standard deviation of the KL distance due to random variation in the sample statistics. Notice that when the real model order K is small, the SMT and SMT-S perform substantially better than the other estimators. As K becomes large, the SMT performance becomes close to the shrinkage and L-W estimators, which may be caused by the fact that the SMT greedy minimization leads to a local minimum. However, the SMT-S still performs best among all the estimators in this case.

Figure 6 shows the estimated eigenvalues for $K = 200$ and $K = 600$ with $n = 100$, respectively. It can be seen that the SMT and SMT-S achieves more accurate eigenvalue estimates, especially for the small eigenvalues. We also measured the agreement of the eigenspaces resulting from the estimated eigenvectors and the true eigenvectors as in [34]. The measure that was used to compare the eigenspaces spanned by the first q eigenvectors is defined in [35] as

$$D(q) = \sum_{i=1}^{q} \sum_{j=1}^{q} \left(\hat{e}_{(i)}^{t} \cdot e_{(j)} \right)^2 , \tag{40}$$

where $\hat{e}_{(i)}$ denotes the estimated eigenvector corresponding to the i-th largest estimated eigenvalue, and $e_{(j)}$ is the corresponding j-th true eigenvector. For any $1 \leq q \leq p$, perfect agreement between the two eigenspaces will result in $D(q) = q$. Figure 7 shows for eigenspace measure between the various estimators and the true covariance for the case of $K = 200$ and $K = 600$ with $n = 100$. Note that the plots of the SMT and SMT-S almost overlap with the other in Fig. 7(a), as do the plots of the L-W estimator and the sample covariance. It can be seen, when the true model order K is small, the SMT and SMT-S estimator achieve a much better estimate of the eigenspaces. When K is large, the SMT-S improves the estimates of the eigenvectors associated with large eigenvalues over the SMT (see Fig. 7(b)).



(a) $n = 50$ (b) $n = 100$ (c) $n = 200$

Fig. 5. Kullback-Leibler distance from true covariances that consist of K randomly generated Givens rotations, $p = 200$.

(a) $K = 200$ (b) $K = 600$

Fig. 6. Eigenvalue estimation when the true covariances consist of K randomly generated Givens rotations with $p = 200$ and $n = 100$.

(a) $K = 200$ (b) $K = 600$

Fig. 7. Eigenspace estimation when the true covariances consist of K randomly generated Givens rotations with $p = 200$ and $n = 100$.

2) Autoregressive Model, Moving Average Model and SPCA: Autoregressive (AR) and moving average (MA) models are very common test cases for covariance estimators in the literature. It is known that

AR and MA models represent stationary processes (without considering the boundary effect), therefore, their eigenvector matrix can be approximately represented by the SMT with a low model order K. We also use the AR example to illustrate the difference between the SMT and sparse principal component analysis (SPCA) methods [19], [20], and we used the publicly available R source code [36] for SPCA implementation of $n \ll p$ case.

First, an AR(1) model is constructed with

$$R_{ij} = \rho^{|i-j|} \ , \tag{41}$$

where $\rho = 0.5$ and $p = 200$. Then, we generate n Gaussian random samples with zero mean and covariance R, and use these sample data for covariance estimation. The experiment was repeated 10 times for $n = 50$, 100 and 200. Figure 8(a) and (b) show the eigenvalue and eigenspace estimates for $n = 100$, respectively. Since the eigenspace estimate of the L-W method is essentially equivalent to the sample covariance, its corresponding plot is not shown in Fig. 8(b). It is clear that SPCA failed to estimate the small eigenvalues and eigenvectors ($q > 100$). This is because the SPCA method only estimates up to n principal components (PC) when $n < p$ and results in the remaining PCs being all zero. The number of non-zero PCs can be smaller $(< n)$ if sparsity regularization is increased. Figure 9(a) shows the estimated model order for the SMT using 3-fold cross-validation, and Fig. 9(b) shows the aggregate results of the KL distances of the various estimators. It can be seen that the SMT estimators outperform the other estimators in all cases. Notice that the KL distance of the low-rank SPCA estimator is infinity and thus not shown in the figure.

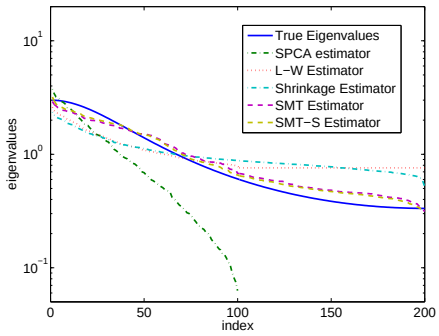
A similar experiment is investigated for an MA(2) model with the covariance given by

$$R_{ij} = \begin{cases} \rho^{|i-j|} & \text{if } |i-j| \leq 2 \\ 0 & \text{otherwise} \end{cases} \ , \tag{42}$$

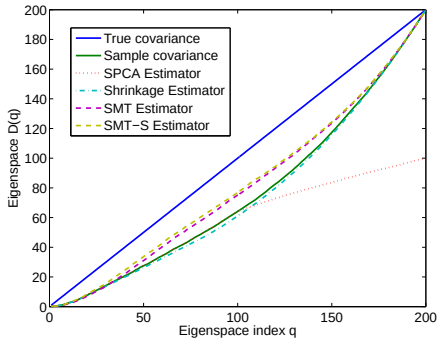
where $\rho = 0.5$ and $p = 200$. The results, shown in Fig. 10, are quite similar to the AR case.

C. SMT Covariance Estimation for Hyperspectral Data Classification

The hyperspectral data we use is available in the recently published book [37]. Figure 11(a) shows a simulated color IR view of an airborne hyperspectral data flightline over the Washington DC Mall. The sensor system measured the pixel response in 191 effective bands ($p = 191$) in the 0.4 to 2.4 μm region of the visible and infrared spectrum. The data set contains 1208 scan lines with 307 pixels in each scan line. The image was made using bands 60, 27, and 17 for the red, green, and blue colors, respectively.

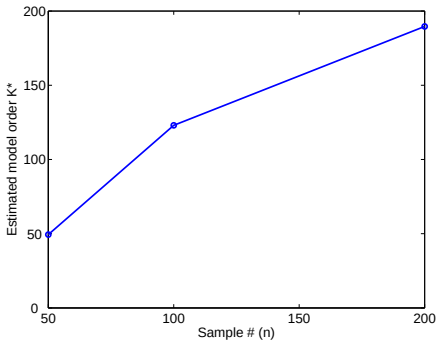


(a) Eigenvalue estimation

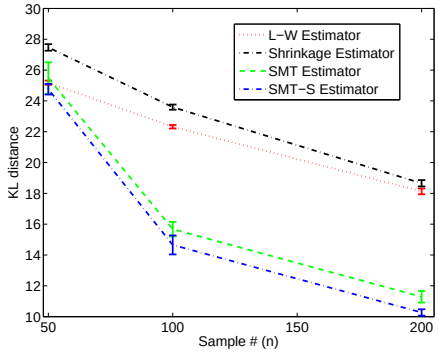


(b) Eigenspace estimation

Fig. 8. Eigenvalue and eigenspace estimates for AR(1) model using various methods. Here, $p = 200$ and $n = 100$.

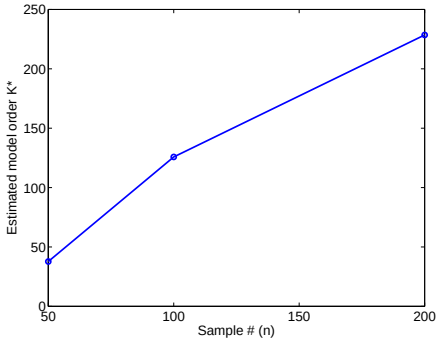


(a) Estimated model order of SMT

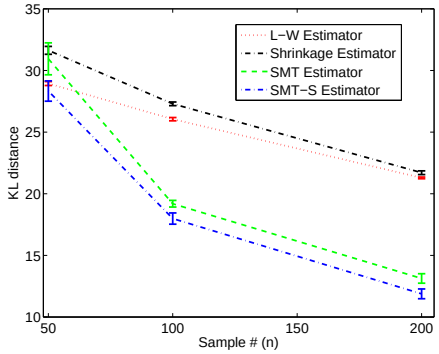


(b) KL distance

Fig. 9. Covariance estimation for AR(1) model with $p = 200$: (a) Estimated model order of SMT; (b) KL distance.



(a) Estimated model order of SMT



(b) KL distance

Fig. 10. Covariance estimation for MA(2) model with $p = 200$: (a) Estimated model order of SMT; (b) KL distance.

The data set also provides ground-truth pixels for five classes designated as grass, water, street, roof, and tree. In Fig. 11(a), the ground-truth pixels of the grass class are outlined with a white rectangle.

Figure 11(b) shows the spectrum of the grass pixels, and Fig. 11(c) shows multivariate Gaussian vectors that were generated using the measured sample covariance for the grass class.

For each class, we computed what we will call a "ground-truth covariance" from the full set of ground-truth pixels for that class.[7] Each ground-truth covariance was computed by first subtracting the sample mean vector for each class, and then computing the sample covariance for the zero mean vectors. The number of pixels for the ground-truth classes of grass, water, roof, street, and tree are 1928, 1224, 3579, 416, and 388, respectively.

1) Gaussian case: First, we compare how different estimators perform when the data vectors are samples from an ideal multivariate Gaussian distribution. To do this, we first generated zero mean multivariate vectors with covariance corresponding to the five ground-truth covariances. Next we estimated the covariance using the different methods, the L-W estimator, shrinkage estimator, glasso, SMT, and SMT shrinkage estimation. Since the L-W estimator performed significantly worse than the other methods for the hyperspectral datasets (see Fig. 13(a) for example), we only focus on the other four methods here for clarity. In each case, a 3-fold cross-validation (i.e. $t = 3$ in (24)) is used to choose the regularization parameter for SMT and glasso. Figure 12 is an example of the plot of the average cross-validated log-likelihood as a function of the number of Givens rotations K in the SMT covariance estimate. In order to determine the effect of sample size, we also performed each experiment for a sample size of $n = 80$, 40, and 20, respectively. Every experiment was repeated 10 times with re-generated data Y each time.

In order to get an aggregate assessment of the effectiveness of SMT covariance estimation, we compared the estimated covariance for each method to the corresponding ground-truth covariance using the KL distance. Figures 13(a), (b) and (c) show plots of the KL distances as a function of sample size for the four estimators. Notice that the SMT shrinkage (SMT-S) estimator is consistently the best of the four.

Figure 14(a) shows the estimated eigenvalues for the grass class with $n = 80$. Notice that the eigenvalues of the SMT and SMT-S estimators are much closer to the true values than the shrinkage and glasso methods. In particular, the SMT estimators tend to generate better estimates of the small eigenvalues.

Table II compares the computational complexity, CPU time (with and without cross-validation) and the chosen regularization parameter values of the different covariance estimation methods. The numerical results were based on the Gaussian case of the grass class with $n = 80$. Notice that even with cross-validation, the SMT and SMT-S estimators are much faster than glasso without cross-validation. In this

[7]We call this the "ground-truth covariance" because it will be used to generate multivariate Gaussian simulation data in some experiments.

(a) (b) (c)

Fig. 11. (a) Simulated color IR view of an airborne hyperspectral data over the Washington DC Mall [37]. (b) Ground-truth pixel spectrum of grass pixels that are outlined with the white rectangles in (a). (c) Synthesized data spectrum using the Gaussian distribution.

example, the SMT uses an average of $K = 495$ rotations, which is equal to $K/p = 495/191 = 2.59$ rotations per spectral sample.

2) Non-Gaussian case: In practice, the sample vectors may not be from an ideal multivariate Gaussian distribution. In order to see the effect of the non-Gaussian statistics on the accuracy of the covariance estimate, we performed a set of experiments which used random samples from the ground-truth pixels as input. Since these samples are from the actual measured data, their distribution is not likely to be precisely Gaussian. Using these samples, we computed the covariance estimates for the five classes using the four different methods with sample sizes of $n = 80$, 40, and 20.

Plots of the KL distances for the non-Gaussian case[8] are shown in Fig. 13(d), (e) and (f); and Figure 14(b) shows the estimated eigenvalues for grass with $n = 80$. Note that the results are similar to those found for the ideal Gaussian case. This indicates that the SMT estimators perform robustly with real data that often have non-Gaussian distributions.

D. SMT Covariance Estimation for Eigen Image Analysis

Eigen-image analysis is an important problem in statistical image processing and pattern recognition. For example, eigenface analysis is a well-known technique in face recognition and face image compression [38].

Figure 15 shows how the SMT can be used to efficiently perform eigen-image analysis. First, SMT covariance estimation is used to estimate the covariance from n image samples, as in Fig. 15(a). Here every column of Y is an 2D face image. The SMT estimator can produce a full set of eigenfaces from

[8]In fact, these are the KL distances between the estimated covariance and the sample covariance computed from the full set of training data, under the assumption of a multivariate Gaussian distribution.

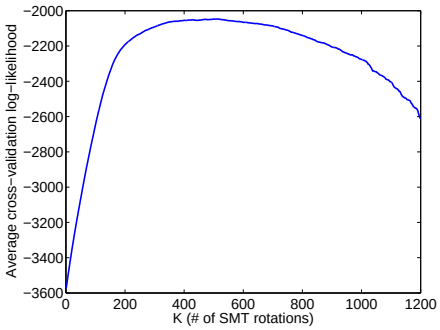
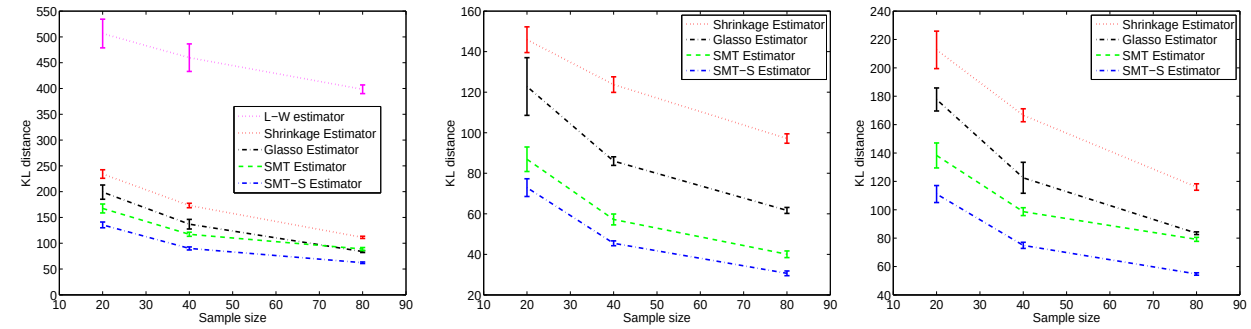


Fig. 12. Plot of the average log-likelihood as a function of the number of Givens rotations K in the SMT cross-validation of the grass class. The value of K that achieves the highest average log-likelihood is chosen as the number of rotations in the final SMT covariance estimator. $K = 495$ in this example (Gaussian case, $n = 80$).



(a) Grass (b) Water (c) Street

(d) Grass (e) Water (f) Street

Fig. 13. Kullback-Leibler distance from true distribution versus sample size for various classes: (a) (b) (c) Gaussian case (d) (e) (f) non-Gaussian case.

the limited number of images. Also, with the fast transform property, one can either compute the eigen-image decomposition of a single image (see Fig. 15(b)), or using the adjoint transform, one can compute individual eigen images on-the-fly (see Fig. 15(c)). Notice that it is typically not practical to store all the eigen images since this would require the storage of a $p \times p$ matrix, where p is the number of pixels

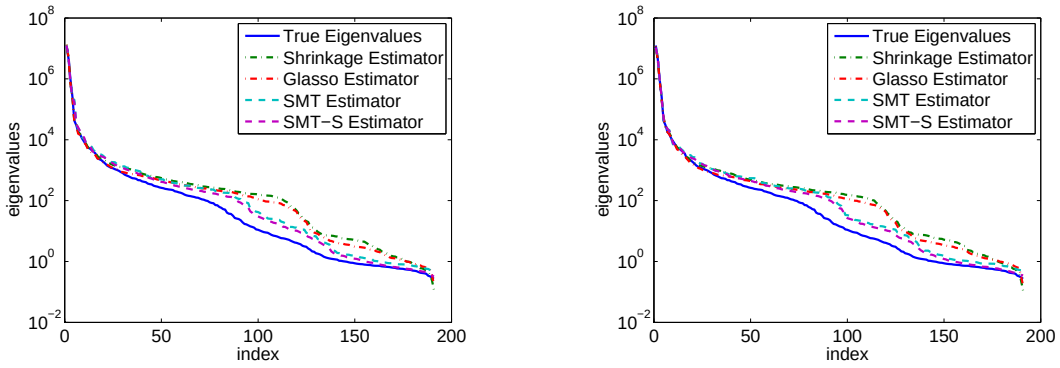


Fig. 14. The distribution of estimated eigenvalues for the grass class with $n = 80$: (a) Gaussian case (b) Non-Gaussian case.

TABLE II

COMPARISON OF COMPUTATIONAL COMPLEXITY AND CPU TIME OF VARIOUS COVARIANCE ESTIMATORS. THE COMPLEXITY DOES NOT INCLUDE THE COMPUTATION OF THE SAMPLE COVARIANCE. HERE, THE NUMERICAL RESULTS ARE BASED ON THE GAUSSIAN CASE OF THE GRASS CLASS WITH $n = 80$. m – NUMBER OF TEST VALUES FOR THE REGULARIZATION PARAMETER, t – NUMBER OF SPLIT SUBSETS IN CROSS-VALIDATION, AND i – NUMBER OF ITERATIONS IN GLASSO. C.V. STANDS FOR CROSS-VALIDATION.

	Observed Complexity		CPU time (sec.)		Parameter
	w/o c.v.	with c.v.	w/o c.v.	with c.v.	
Shrinkage	p	$m(p^3 + np^2)$	≈ 0	8.6	$\alpha = 0.0016$
glasso	$p^3 i$	$tmp^3 i$	422.6	38141.3	$\rho = 0.0005$
SMT	$p^2 + Kp$	$t(p^2 + Kp)$	1.6	6.5	$K = 495$
SMT-S	$p^2 + Kp$	$m(p^3 + np^2)$	1.6	7.2	$(K, \alpha) = (495, 0.6)$

in the image. However, the new method only requires the storage of the parameters of the K Givens rotations, which can be easily stored even for large images.

The face image dataset we used is from the ORL Face Database [39], with the images re-scaled to 28×23 pixels ($p = 644$). There are 40 different individuals and we used 2 face images for each individual as our training data, which results in $n = 80$. Examples of the image set used in the experiments are shown in Fig. 16. First, we subtracted the sample mean from these images, and used the mean-subtracted images as our sample data for covariance estimation. We compared the SMT estimators with other covariance estimators in terms of both the accuracy and visual quality. In particular, we included the diagonal covariance estimator, i.e. $\hat{R} = \mathrm{diag}(S)$, which represents an independent pixel model.

1) Eigenfaces: Figure 17(a) shows the plot of average cross-validated log-likelihood ($t = 3$) for the face images as a function of model order K. The value of K that achieved the highest average log-likelihood is 974 in this example. Figure 17(b) shows the values of the regularization parameters for different estimators chosen by cross-validation (except the L-W estimator). Figures 18(a) – (f) show the first 80 estimated eigenfaces (i.e. columns of $\hat{E}$) using the different methods. Interestingly, compared to the eigenfaces resulting from the other estimators, the SMT eigenfaces clearly show much more visual structure corresponding to hair, glasses etc. Also notice that the SMT eigenfaces tend to be sparse.

2) Cross-Validated Log-Likelihood: Since it is not possible to obtain the "true" covariance of face images due to the limited sample size, we used the cross-validated log-likelihood as a measure of accuracy of different estimators. Figure 19(a) shows the average 3-fold cross-validated log-likelihood of the face images using the SMT covariance estimators, as compared to the diagonal, L-W, shrinkage, and glasso covariance estimators. Notice that the SMT covariance estimators produced much higher average cross-validated log-likelihood than the traditional shrinkage estimator, and the SMT-S estimator resulted in the highest likelihood. In Fig. 19(b), we show the maximum log-likelihood values for all the methods, and the differences from the traditional shrinkage estimator. Notice that SMT-S has an increase in log-likelihood of 167.3 as compared to the shrinkage estimate. Also notice the difference between shrinkage and an independent pixel model (i.e. diagonal covariance) is 349.7. This is interesting since an independent pixel model of faces is known to be a poor model. The glasso has an increase in log-likelihood of 164.5, which is consistent with the common belief that a sparse inverse covariance is a good model for images.

3) Automated Generation of Face Image Samples: In order to better illustrate the advantage of SMT covariance estimation, we generated random face image samples using these different covariance estimates. Specifically, the face image samples were generated under the Gaussian distribution

$$y \sim N(\bar{y}, \hat{R}) \ , \tag{43}$$

where $\bar{y}$ is the sample mean of the training images and $\hat{R}$ denotes different covariance estimates. The generated sample images are shown in Fig. 20(a)–(f). While these results are subjective in nature, the faces generated by the SMT models tend to have substantially more detail than those generated with the shrinkage model, and are perhaps comparable in quality to the faces generated by the glasso model.

VI. Conclusion

We have proposed a novel method for covariance estimation of high dimensional signals. The new method is based on constrained maximum likelihood (ML) estimation in which the eigen-transformation

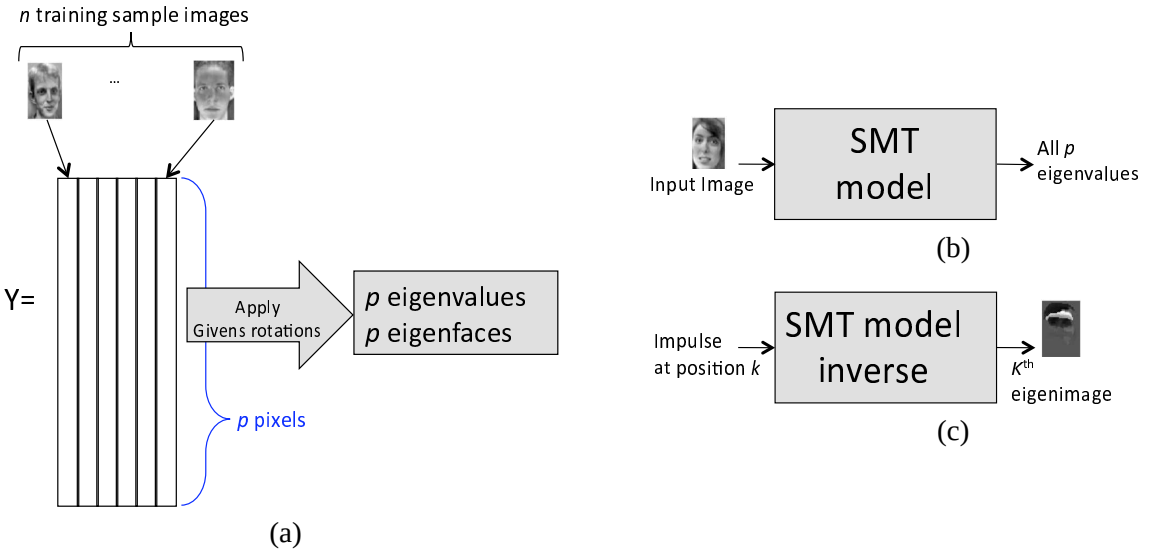
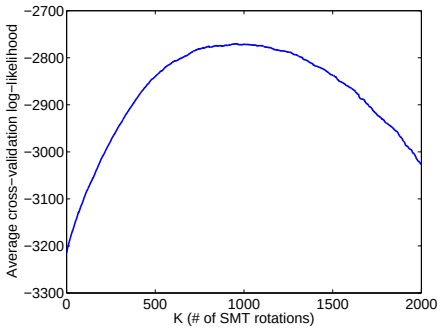


Fig. 15. This figure illustrates how the SMT covariance estimation can be used for eigen-image analysis. (a) A set of n images can be used to estimate the associated SMT. (b) The resulting SMT can be used to analyze a single input image, or (c) the transpose (i.e. inverse) of the SMT can be used to compute the k-th eigen image by applying an impulse at position k. Notice that both the SMT and inverse SMT are sparse fast transforms even when the associated image is very large.

Fig. 16. Face image samples from the face image database [39] for eigen-image analysis ($n = 80$ and $p = 28 \times 23$)

is constrained to be the composition of K Givens rotations. This model seems to capture the essential behavior of the signals with a relatively small number of parameters. The constraint set is a K dimensional manifold in the space of orthonormal transforms, but since it is not a linear space, the resulting ML estimation optimization problem does not yield a closed form global optimum. However, we show that a recursive local greedy optimization procedure is simple, intuitive, and yields good results.

We demonstrate the effectiveness of the new approach on simulated data, hyperspectral data and face



(a)

Method	Parameter
Diagonal	–
L-W	$\alpha = 0.14$
Shrinkage	$\alpha = 0.28$
glasso	$\rho = 0.08$
SMT	$K = 974$
SMT-S	$(K, \alpha) = (974, 0.8)$

(b)

Fig. 17. (a) Plot of the average log-likelihood as a function of the number of Givens rotations K in the SMT cross-validation. The value of K that achieves the highest average log-likelihood is chosen as the number of rotations in the final SMT covariance estimator. $K = 974$ in this example. (b) The values of the regularization parameters that were chosen by cross-validation for different covariance estimation methods.

image sets. In addition to providing a more accurate estimate of the covariance, the new method offers the potential for large computational advantages when the dimension of siginals is high, as is the case with images. The resulting SMT eigen-transformation is shown to be a generalization of the classical FFT and orthonormal wavelet transform. However, unlike the FFT and wavelet transform, the SMT is suitable for fast decorrelation of general non-stationary signals. The MATLAB code for SMT covariance estimation is available at: https://engineering.purdue.edu/~bouman.

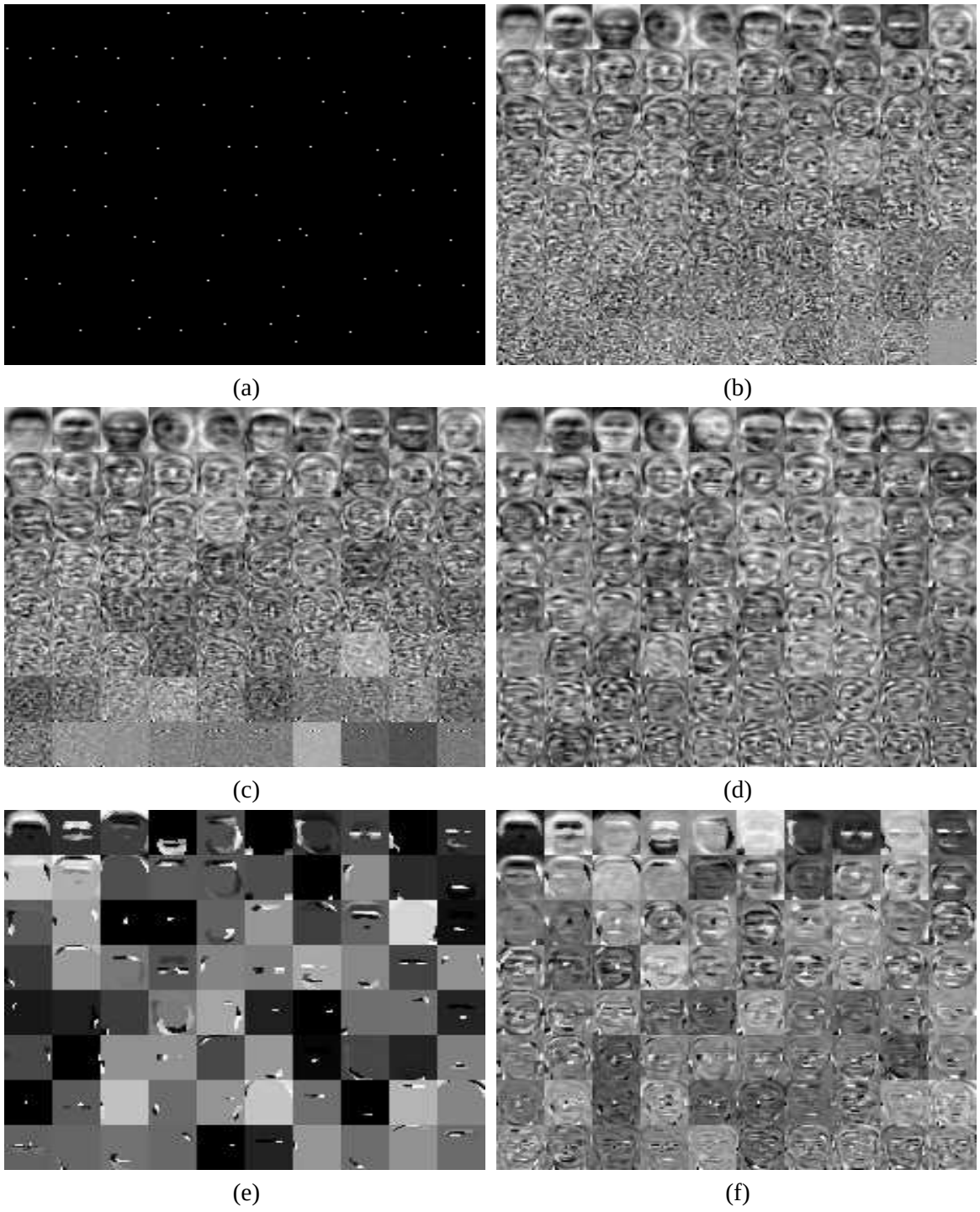
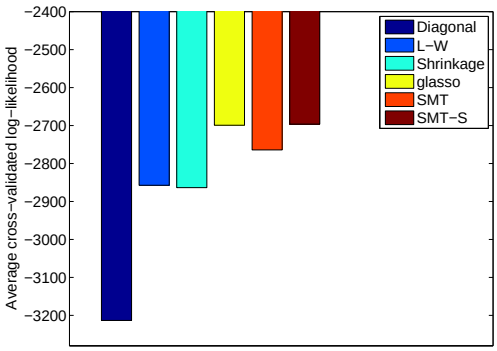


Fig. 18. Experimental results of eigen-image analysis. First 80 eigen-images for each of the following methods: (a) Diagonal covariance estimate (i.e. independent pixels); (b) L-W covariance estimate; (c) Shrinkage covariance estimate; (d) graphical lasso covariance estimate; (e) SMT covariance estimate; (f) SMT-S covariance estimate. Notice that the SMT covariance estimate tends to generate eigen-images that correspond to well defined spatial features such as hair or glasses in faces.

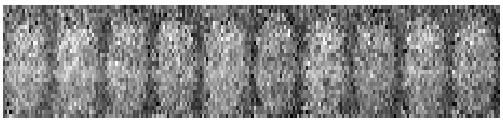


(a)

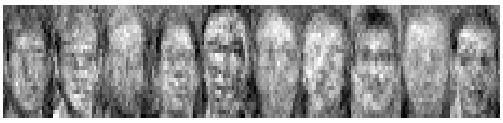
Method	Log-Likelihood	Δ
Diagonal	-3213.3	-349.7
L-W	-2857.2	6.4
Shrinkage	-2863.6	0
glasso	-2699.1	164.5
SMT	-2764.2	99.4
SMT-S	-2696.3	167.3

(b)

Fig. 19. (a) The graph shows the average cross-validated log-likelihood of the face images using the diagonal, L-W, shrinkage, glasso, SMT and SMT-S covariance estimates. (b) The table shows the value of the cross-validated log-likelihood for each estimator and their difference. Notice that SMT-S has an increase in log-likelihood over shrinkage of 167.3. This is comparable to 349.7, the difference between shrinkage and an independent pixel model (i.e. diagonal covariance).

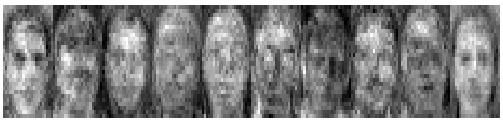


(a) diagonal

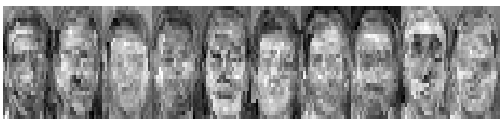


(b) L-W

(c) shrinkage



(d) glasso



(e) SMT

(f) SMT-S

Fig. 20. Generated face image samples under the Gaussian distribution with the sample mean and different covariance estimates: (a) Diagonal covariance estimate (b) L-W covariance estimate (c) Shrinkage covariance estimate (d) Glasso covariance estimate (e) SMT covariance estimate (f) SMT-S covariance estimate.

APPENDIX A
DERIVATION OF MAXIMUM LIKELIHOOD ESTIMATES OF EIGENVECTORS AND EIGENVALUES

If the columns of Y are independent and identically distributed Gaussian random vectors with mean zero and positive-definite covariance R, then the likelihood of Y given R is given by

$$\begin{aligned} P_{(E,\Lambda)}(Y) &= \frac{1}{(2\pi)^{\frac{np}{2}}} |R|^{-\frac{n}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\{Y^t R^{-1} Y\}\right\} && (44)\\ &= \frac{1}{(2\pi)^{\frac{np}{2}}} |\Lambda|^{-\frac{n}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\{Y^t E\Lambda^{-1} E^t Y\}\right\} && (45)\\ &= \frac{1}{(2\pi)^{\frac{np}{2}}} |\Lambda|^{-\frac{n}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\{E^t Y Y^t E\Lambda^{-1}\}\right\} && (46)\\ &= \frac{1}{(2\pi)^{\frac{np}{2}}} |\Lambda|^{-\frac{n}{2}} \exp\left\{-\frac{n}{2}\mathrm{tr}\{E^t S E\Lambda^{-1}\}\right\} && (47)\\ &= \frac{1}{(2\pi)^{\frac{np}{2}}} |\Lambda|^{-\frac{n}{2}} \exp\left\{-\frac{n}{2}\mathrm{tr}\{\mathrm{diag}(E^t S E)\Lambda^{-1}\}\right\} \ . && (48) \end{aligned}$$

Taking the logarithm yields

$$\log P_{(E,\Lambda)}(Y) = -\frac{n}{2}\mathrm{tr}\{\mathrm{diag}(E^t S E)\Lambda^{-1}\} - \frac{n}{2}\log|\Lambda| - \frac{np}{2}\log(2\pi) \ . \quad (49)$$

Therefore, the maximum likelihood (ML) estimator of (E, Λ) is given by

$$\begin{aligned} (\hat{E}, \hat{\Lambda}) &= \arg\max_{(E,\Lambda)} \log P_{(E,\Lambda)}(Y) && (50)\\ &= \arg\max_{E}\max_{\Lambda} \log P_{(E,\Lambda)}(Y) \ . && (51) \end{aligned}$$

We first maximize the log-likelihood with respect to Λ. Setting the derivatives of $\log P_{(E,\Lambda)}(Y)$ with respect to all the diagonal entries of Λ to zero, we obtain

$$\hat{\Lambda} = \mathrm{diag}(E^t S E) \ . \quad (52)$$

Therefore, the ML estimation of E is given by

$$\begin{aligned} \hat{E} &= \arg\max_{E\in\Omega} \log P_{(E,\hat{\Lambda}(E))}(Y) && (53)\\ &= \arg\max_{E\in\Omega} \left\{-\frac{np}{2}\log(2\pi) - \frac{n}{2}\log\left|\mathrm{diag}(E^t S E)\right| - \frac{np}{2}\right\} && (54)\\ &= \arg\min_{E\in\Omega} \left\{\left|\mathrm{diag}(E^t S E)\right|\right\} \ , && (55) \end{aligned}$$

where Ω is the set of allowed orthonormal transforms. So the minimization of $\left|\mathrm{diag}(E^t S E)\right|$ leads to

the ML estimate of E, and hence the ML estimate of Λ which is given by

$$\hat{\Lambda} = \mathrm{diag}(\hat{E}^t S \hat{E}) \ . \tag{56}$$

APPENDIX B
UNCONSTRAINED ML ESTIMATE

Proposition: Let S be a $p \times p$ positive definite symmetric matrix with eigenvalue decomposition given by $S = E^* \Lambda_S E^{*t}$, and let Ω be the set of all $p \times p$ orthonormal transforms. Then E^* achieves the global minimization of (6), so that

$$\left|\mathrm{diag}(E^{*t} S E^*)\right| = \min_{E \in \Omega} \left\{ \left|\mathrm{diag}(E^t S E)\right| \right\} \ . \tag{57}$$

First, we show for any symmetric, positive definite matrix S, we have

$$|\mathrm{diag}(S)| \geq |S| \ . \tag{58}$$

We know there exists a unique lower triangular $p \times p$ matrix G, such that

$$S = GG^t \ , \tag{59}$$

which is called the Cholesky factorization [40]. Therefore, $|S| = |G|^2 = \prod_{i=1}^p G_{ii}^2$. Clearly, we have $S_{ii} = \sum_{j=1}^p G_{ij}^2 \geq G_{ii}^2$ for $i = 1, 2, \ldots, p$. This gives

$$|\mathrm{diag}(S)| \geq \prod_{i=1}^p G_{ii}^2 = |S| \ . \tag{60}$$

The equality holds if and only if $S_{ii} = G_{ii}^2$ for $i = 1, 2, \ldots, p$, which is equivalent to the fact that S is diagonal. Therefore, we know for any orthonormal transform E,

$$\left|\mathrm{diag}(E^t S E)\right| \geq \left|E^t S E\right| = |S| \ . \tag{61}$$

If $S = E^* \Lambda_S E^{*t}$ is an eigen-decomposition of S, then we know

$$\left|\mathrm{diag}(E^{*t} S E^*)\right| = |\Lambda_S| = |S| \ . \tag{62}$$

Therefore, E^* is the solution of global minimization of (6) if the sample covariance S is non-singular.

APPENDIX C
EXACT SMT FACTORIZATION OF ORTHONORMAL TRANSFORMS

We know the Givens QR factorization can be used to find a decomposition of a $p \times p$ matrix into $\binom{p}{2}$ Givens rotations [40]. Let A be an $p \times p$ orthonormal matrix, and let $Q = G_1 G_2 ... G_K$ with $K = \binom{p}{2}$, so that

$$A = QR \ , \tag{63}$$

where every G_k is a Givens rotation and R is upper triangular. Since A and Q are orthonormal, R must be orthonormal. Since R is also upper triangular, this means that it must be diagonal. Therefore, R is a diagonal orthonormal matrix, which means that it is the identity matrix. Hence, we have $A = Q$.

APPENDIX D
SOLUTION OF (12) FOR A SPECIFIED COORDINATE INDEX PAIR

In this appendix, we will find the solution to the optimization problem of (12) for a specified coordinate pair and the corresponding change of the cost function. Since the coordinate index pair is specified, we can assume all the matrices to be 2×2 without loss of generality.

From Appendix B, we know that E minimizes the cost function (12) if and only if E is the eigenvector matrix of S. Next we obtain an expression for E in terms of a Givens rotation. Let

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \ , \tag{64}$$

and let $E = I + \Theta(1, 2, \theta)$ with $\theta = \frac{1}{2}\text{atan}(-2s_{12}, s_{11} - s_{22})$. Then we have

$$E^t S E = \begin{bmatrix} s'_{11} & 0 \\ 0 & s'_{22} \end{bmatrix} \ , \tag{65}$$

where

$$s'_{11} = \frac{1}{2}\left(s_{11} + s_{22} + \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}\right) \tag{66}$$

$$s'_{22} = \frac{1}{2}\left(s_{11} + s_{22} - \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}\right) \ . \tag{67}$$

This shows that E of the given form is the eigenvector matrix of S. Hence E must minimize the cost function of (12). Based on (62), we know that the ratio of the cost function before and after the transform

of E is given as

$$\frac{\left|\mathrm{diag}\left(E^tSE\right)\right|}{|\mathrm{diag}\left(S\right)|} = \frac{|S|}{|\mathrm{diag}\left(S\right)|} = 1 - \frac{s_{12}^2}{s_{11}s_{22}} \ . \tag{68}$$

APPENDIX E
PERMUTATION INVARIANCE OF THE SMT ESTIMATOR

Property: The SMT covariance estimate is permutation invariant. More specifically, if $\hat{R} = \hat{E}\hat{\Lambda}\hat{E}^t$ is the unique order-K SMT covariance estimate of the data Y, then for any permutation matrix P, the order-K SMT covariance estimate of the permuted data PY is given by $P\hat{R}P^t$.

Uniqueness of $\hat{R}$ means that (14) is assumed to have a unique minimum at each step $k \leq K$. Let S be the sample covariance of Y, and $\tilde{S} = PSP^t$ be the sample covariance of the permuted data PY. The proof can be shown by construction. First consider the case of $k = 1$. Let

$$(i_k, j_k) \leftarrow \arg\min_{(i,j)} \left(1 - \frac{[S_k]_{ij}^2}{[S_k]_{ii}[S_k]_{jj}}\right) \ , \tag{69}$$

and the Givens rotation is given by $E_k = I + \Theta(i_k, j_k, \theta_k)$. Let i'_k and j'_k be the corresponding row (or column) indices of i_k and j_k in the permuted matrix $\tilde{S}$. Without loss of generality, we assume $i_k < j_k$ and $i'_k < j'_k$. Then the Givens rotation resulting from the permuted data is given by $\tilde{E}_k = I + \Theta(i'_k, j'_k, \theta_k)$. We know $\tilde{E}_k = PE_kP^t$. Thus we have

$$\tilde{\Lambda}_k = \mathrm{diag}(\tilde{E}_k^t\tilde{S}\tilde{E}_k) = \mathrm{diag}(P\hat{E}_k^tS\hat{E}_kP^t) = P\mathrm{diag}(\hat{E}_k^tS\hat{E}_k)P^t = P\hat{\Lambda}_kP^t \ . \tag{70}$$

Therefore, the order-k SMT covariance estimator of PY is given by

$$\tilde{R}^{(k)} = \tilde{E}_k\tilde{\Lambda}_k\tilde{E}_k^t = P\hat{E}_k\hat{\Lambda}_k\hat{E}_k^tP^t = P\hat{R}^{(k)}P^t \ . \tag{71}$$

Next at step $k+1$, we have

$$S^{(k+1)} = E_k^tSE_k \tag{72}$$

$$\tilde{S}^{(k+1)} = \tilde{E}_k^t\tilde{S}\tilde{E}_k = PS^{(k+1)}P^t \ . \tag{73}$$

Following the same derivation, we know the conclusion in (71) still holds at step $k+1$.

APPENDIX F
KULLBACK-LEIBLER DISTANCE

The Kullback-Leibler (KL) distance between two distributions $P_\theta(y)$ and $P_{\hat{\theta}}(y)$ is defined as [33]

$$d(\theta, \hat{\theta}) = E_\theta \left[\log P_\theta(y) - \log P_{\hat{\theta}}(y) \right] \ .$$

So if $\theta = (E, \Lambda)$ and $\hat{\theta} = (\hat{E}, \hat{\Lambda})$, then under the assumption of Gaussian distribution the KL distance is given by

$$\begin{aligned} d(\theta, \hat{\theta}) &= E_\theta \left[\log P_\theta(y) - \log P_{\hat{\theta}}(y) \right] \\ &= -\frac{1}{2}\mathrm{tr}\{\mathrm{diag}(E^t R E)\Lambda^{-1}\} - \frac{1}{2}\log|\Lambda| + \frac{1}{2}\mathrm{tr}\{\mathrm{diag}(\hat{E}^t R \hat{E})\hat{\Lambda}^{-1}\} + \frac{1}{2}\log\left|\hat{\Lambda}\right| \\ &= \frac{1}{2}\mathrm{tr}\{\mathrm{diag}(\hat{E}^t R \hat{E})\hat{\Lambda}^{-1} - I\} + \frac{1}{2}\log\left|\hat{\Lambda}\Lambda^{-1}\right| \ . \end{aligned} \tag{74}$$

We use the Kullback-Leibler distance as one of the measures for the various covariance estimators.

REFERENCES

[1] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York: Springer, 2009, 2nd Edition.

[2] D. L. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," in *Math Challenges of the 21st Century*. Los Angeles: American Mathematical Society, August 8 2000.

[3] R. E. Bellman, *Adaptive Control Processes*. Princeton, NJ: Princeton University Press, 1961.

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

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

[6] J. Theiler, "Quantitative comparison of quadratic covariance-based anomalous change detectors," *Applied Optics*, vol. 47, no. 28, pp. F12–F26, 2008.

[7] C. Stein, B. Efron, and C. Morris, "Improving the usual estimator of a normal covariance matrix," Dept. of Statistics, Stanford University, Report 37, 1972.

[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Boston, MA: Academic Press, 1990, second Edition.

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

[10] 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.

[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, "A well-conditioned estimator for large-dimensional covariance matrices," *J. Multivar. Anal.*, vol. 88, no. 2, pp. 365–411, 2004.

[13] 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.

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

[15] 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.

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

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

[18] C. Chennubhotla and A. Jepson, "Sparse PCA: Extracting multi-scale structure from data," *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 641–647 vol.1, 2001.

[19] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin, "A modified principal component technique based on the lasso," *Journal of Computational and Graphical Statistics*, vol. 12, no. 3, pp. 531–547, 2003.

[20] 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.

[21] I. M. Johnstone and A. Y. Lu, "Sparse Principal Components Analysis," *ArXiv e-prints*, Jan. 2009.

[22] G. Cao and C. Bouman, "Covariance estimation for high dimensional data vectors using the sparse matrix transform," in *Advances in Neural Information Processing Systems*. MIT Press, 2008, pp. 225–232.

[23] G. Cao and C. A. Bouman, "Covariance estimation for high dimensional data vectors using the sparse matrix transform," Purdue University, Technical Report TR-ECE 08-05, 2008.

[24] G. Cao, C. Bouman, and K. Webb, "Noniterative map reconstruction using sparse matrix representations," *IEEE Transactions on Image Processing*, vol. 18, no. 9, pp. 2085–2099, Sept. 2009.

[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, no. 1, pp. 26–50, March 1958.

[26] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, April 1965.

[27] G. Cao, C. A. Bouman, and J. Theiler, "Weak signal detection in hyperspectral imagery using sparse matrix transformation (SMT) covariance estimation," in *WHISPERS: First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, Grenoble, France, August 2009.

[28] A. Soman and P. Vaidyanathan, "Paraunitary filter banks and wavelet packets," *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 4, pp. 397–400 vol.4, Mar 1992.

[29] P. Vaidyanathan, *Multirate systems and filter banks*. Englewood Cliffs: Prentice Hall, 1993.

[30] 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.

[31] `http://www.ledoit.net/ole1_abstract.htm`.

[32] `http://cran.r-project.org/web/packages/glasso/index.html`.

[33] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[34] A. J. Rothman, E. Levina, and J. Zhu, "Generalized thresholding of large covariance matrices," *Journal of the American Statistical Association*, vol. 104, no. 485, pp. 177–186, 2009.

[35] W. Krzanowski, "Between-groups comparison of principal components," *Journal of the American Statistical Association*, vol. 74, no. 367, pp. 703–707, 1979.

[36] `http://cran.r-project.org/web/packages/elasticnet/index.html`.

[37] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. New York: Wiley-Interscience, 2005.

[38] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[39] `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`.

[40] G. Golub and C. V. Loan, *Matrix Computations*. Baltimore: Johns Hopkins University Press, 1996.