MODELING AND PROCESSING OF HIGH DIMENSIONAL SIGNALS AND SYSTEMS USING THE SPARSE MATRIX TRANSFORM

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Guangzhi Cao

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2009

Purdue University

West Lafayette, Indiana

To my parents and Yue

ACKNOWLEDGMENTS

First of all, I would like to thank my outstanding advisors, Professor Charles A. Bouman and Professor Kevin J. Webb. Without their advice, this dissertation can not be made possible. Their exceptional vision, guidance, patience, and kindness have had a great influence on me both in research and life, which I will always be grateful of.

I would also like to thank my other committee members, Professor Peter C. Doerschuk, Professor Philip S. Low and Doctor James Theiler, for dedicating their time and energy. I am especially grateful to James for his enormous effort to bring me to Los Alamos National Laboratory for summer internship, and I really enjoyed my time and research there. I thank Professor Jan Allebach abd Professor Mark Bell for their invaluable advice. My gratitude also goes to the staff members in the graduate office of the department. They have always been very responsible and helpful.

I thank my colleagues, Vaibhav Gaind, Leonardo Bachega, Jianing Wei, Zhou Yu, Yandong Guo and Dalton Lunga. Their collaboration is an indispensable part of this dissertation. I would also like to thank many of my other friends at Purdue, whose names I know so well but are too long to be listed here. Their help and support have made my Purdue life a really fun and worthwhile experience.

I would like to express my sincere gratitude to my wonderful parents, sisters and brother, for their unconditioned love and support in my life. Finally, special thanks to my special one, Yue, who witnessed and shared all my strive and struggle towards the completion of this work. Her endless love and support finally got me here.

TABLE OF CONTENTS

Page

LIST OF TABLES viii

LIST OF FIGURES ix

ABSTRACT xiv

1 INTRODUCTION 1

2 NON-ITERATIVE MAP RECONSTRUCTION USING SPARSE MATRIX REPRESENTATIONS 4

2.1 Introduction 4

2.2 Non-Iterative MAP Reconstruction Framework 7

2.3 Lossy Source Coding of the Inverse Matrix 8

2.3.1 Encoding Framework 8

2.3.2 Distortion Metric 9

2.3.3 Transform Coding for $\tilde{H}$ 11

2.4 Sparse Matrix Transform 15

2.4.1 Cost Function for SMT Design 18

2.4.2 SMT Design Using Greedy Minimization 20

2.4.3 Relation of SMT to PCA and ICA 23

2.5 Numerical Results 23

2.5.1 ODT Example ($M \ll N$) 24

2.5.2 FODT example (M is close to N) 26

2.5.3 Discussion 36

2.6 Conclusion 37

3 INHOMOGENEITY LOCALIZATION IN A SCATTERING MEDIUM IN A STATISTICAL FRAMEWORK 38

3.1 Localization versus Reconstruction 38

Page

3.2 Maximum Likehood Localization 39
3.3 Detection 42
3.4 Source-Detector Geometry Design 43
3.5 Conclusions 45
4 COVARIANCE ESTIMATION FOR HIGH DIMENSIONAL DATA VECTORS USING THE SPARSE MATRIX TRANSFORM 46
4.1 Introduction 46
4.2 Covariance Estimation for High Dimensional Vectors 48
4.2.1 Maximum Likelihood Covariance Estimation 49
4.2.2 ML Estimation of Eigenvectors Using SMT Model 50
4.2.3 Model Order 54
4.2.4 Numerical Stable SMT Covariance Estimator 54
4.3 Algorithm Implementation and Complexity Analysis 55
4.3.1 Case I: $n > p$ or $n \approx p$ 56
4.3.2 Case II: $n \ll p$ 56
4.4 Properties of SMT Covariance Estimator and Its Extensions 57
4.4.1 Properties of SMT Covariance Estimator 57
4.4.2 SMT Shrinkage Estimator 59
4.5 Experimental Results 60
4.5.1 Review of Alternative Estimators 61
4.5.2 SMT Covariance Estimation for Hyperspectral Data Classification 61
4.5.3 SMT Covariance Estimation for Eigen Image Analysis 64
4.6 Conclusion 69
5 WEAK SIGNAL DETECTION IN HYPERSPECTRAL IMAGERY USING THE SPARSE MATRIX TRANSFORM (SMT) COVARIANCE ESTIMATION 76
5.1 Introduction 76
5.2 Review of the SMT Covariance Estimation 77

Page

5.3 Matched Filter Criterion 78
5.4 Numerical Experiments 80
5.4.1 Structure in the Covariance Matrix 81
5.5 Conclusion 85
6 HIGH DIMENSIONAL REGRESSION USING THE SPARSE MATRIX TRANSFORM (SMT) 86
6.1 Introduction 86
6.2 Regression Model 87
6.3 SMT Regression for High Dimensional Data 88
6.3.1 SMT-Lasso 89
6.3.2 SMT-Shrinkage 90
6.3.3 SMT-Subset Selection 91
6.4 Alternative Regression Methods 91
6.4.1 Ordinary Least Squares Regression 91
6.4.2 Ridge Regression 92
6.4.3 Traditional Lasso Regression 92
6.5 Numerical Experiments 92
6.5.1 When τ Is a Small Eigenvector 93
6.5.2 When τ Is a Random Signal 94
6.6 Conclusions 97
LIST OF REFERENCES 98
A COMPUTATION OF H 106
B FAST COMPUTATION OF SMT 107
C OPTIMALITY OF T_{opt} FOR THE SMT COST FUNCTION 108
D OPTIMAL GIVENS ROTATION 110
E RUN-LENGTH CODING 112
F DERIVATION OF MAXIMUM LIKELIHOOD ESTIMATES OF EIGENVECTORS AND EIGENVALUES 113
G UNCONSTRAINED ML ESTIMATE 115

Page

H EXACT SMT FACTORIZATION OF ORTHONORMAL TRANSFORMS 116
I GREEDY ALGORITHM FOR THE SMT DESIGN 117
J PROOF OF PERMUTATION INVARIANCE OF THE SMT ESTIMATOR 118
K KULLBACK-LEIBLER DISTANCE 119
VITA 120

LIST OF TABLES

Table Page

2.1 Comparison of on-line and off-line computation required by various reconstruction methods for the ODT example. Results use number of voxels $N = 65 \times 65 \times 33 = 139425$, number of measurements $M = 720$ (with number of sources $M_1 = 9$ and number of detectors $M_2 = 40$), number of iterations of CG $I = 30$, a run-length coding compression ratio of $c = 1808 : 1$, and number of iterations required to solve the forward PDE $L = 12$. $NRMSE \approx 10\%$ for the compression case. Notice that the non-iterative MAP reconstruction requires much lower on-line computation and memory than the iterative reconstruction method. However, it requires greater off-line computation to encode the inverse transform matrix. 29

2.2 Comparison of on-line and off-line computation required by various reconstruction methods for the FODT example. Results use number of voxels $N = 33 \times 33 \times 17 = 18513$, number of measurements $M = 2500$ (with number of sources $M_1 = 4$ and number of detectors $M_2 = 625$), number of iterations of CG $I = 100$, number of sparse rotations $K = M \log(M) = 28220$, a run-length coding compression ratios of $c_1 = 110 : 1$ and $c_2 = 102 : 1$, for KLT and SMT compression, respectively, and number of iterations required to solve the forward PDE $L = 12$. $NRMSE \approx 10\%$ for the compression cases. Notice that the non-iterative MAP reconstruction requires much less on-line computation and memory than the iterative reconstruction method. However, it requires greater off-line computation to encode the inverse transform matrix. 35

4.1 Complexity of greedy algorithm I and II for the SMT covariance estimation. 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. 57

4.2 Comparison of computational complexity, CPU time and model order for various covariance estimators with and without cross validation. The complexity does not include the computation of the sample covariance. Here, the CPU time and model order were measured as the average results for the Gaussian case of the grass class with $n = 80$. m – number of different cross validation values of the regularization parameter, t – number of splitted subsets in cross validation, and i – number of iterations used in glasso. c.v. stands for cross validation. 65

LIST OF FIGURES

Figure Page

2.1 Illustration of matrix source coding procedure. $\check{H}$ is a transformed representation of H using both KL and wavelet transforms. The gray regions represent the effective non-zero entries in the matrices. Notice that the two transforms concentrate the energy in $\check{H}$ toward the upper left hand corner, so that quantization results in a very sparse matrix. This procedure is performed off-line, so that on-line reconstruction can be very fast. 13

2.2 Implementation of non-iterative MAP reconstruction. (a) The off-line processing algorithm for the matrix source coding of the the inverse matrix H. (b) The on-line processing to compute the reconstruction from the matrix-vector product. 14

2.3 The structure of a pair-wise sparse transform T_k. Here, all the unlabeled diagonal elements are 1's, and all the unlabeled off-diagonal elements are 0's. A_k, B_k and $\tilde{T}_k$ have similar structures. 17

2.4 The structure of SMT implementation. Every $\tilde{T}_k$ is a "butterfly" that can be computed using 2 multiplies. In addition, multiplications by normalization factors is required in the end for a total of $2K + M$ multiplies when K butterflies are used in an M-point transform. The irregular structure of the SMT makes it a generalization of the FFT and allows it to be used to accurately approximate a general orthonormal transform. 17

2.5 Pseudo-code implementation of the greedy algorithm used for the SMT design. 22

2.6 The measurement geometry for optical breast imaging. (a) Imaging geometry. (b) Source-detector probe configuration. The open circles indicate the source fiber locations and the solid circles indicate the detector fiber location. Source fibers and detector fibers are connected to the left and right plates, respectively, and are on 1-cm grid. (Adapted from [45].) . 26

2.7 The reconstructed images of $\mu_a(r)$ at $z = 3$ cm using the compressed H matrix based on KL transform (used both for data whitening and matrix decorrelation). The compression ratios in (c) and (d) are 4267:1 and 1982:1, respectively. Here "bpme" stands for bits per matrix entry. . . 27

Figure Page

2.8 Distortion versus rate for compression using KL transforms for data whitening and matrix decorrelation for the ODT example. Notice here simply whitening the data yields close distortion-rate performance to the theoretically optimal KL transforms. The performance drops significantly with the other three methods that do not perform data whitening. 28

2.9 The measurement geometry for an FODT example. (a) A graphic depiction of the imaging geometry. (b) An illustration of the source-detector probe where the solid circles indicate the locations of sources and the rectangular grid represents the CCD sensor array. 32

2.10 The reconstructed images of $\eta\mu_{a_f}(r)$ at the depth of 2 cm using different compression methods. The compression ratios in (c) and (d) are 110:1 and 103:1, and the NRMSE's are 9.96% and 10.24%, respectively. 33

2.11 Distortion versus rate for the FODT example. (a) Distortion versus rate for compression using the KL transforms for data whitening and matrix decorrelation. (b) Distortion versus rate for compression using the sparse matrix transform (SMT). $M\log_2(M)$ SMT butterflies were used to whiten the measurements and decorrelate the columns of H. Notice that the SMT distortion-rate tradeoff is very close to the distortion-rate of the KL transform. 34

3.1 Measurement geometry for localization. A spherical absorber at depth d is assumed in the simulation. The background optical parameters are: $\mu_{a0} = 0.02$ cm^{-1}, $D_0 = 0.03$ cm, and the modulation frequency is $\omega = 2\pi \times 10^6$ rad/s. 41

3.2 Localization versus reconstruction: (a) Negative log likelihood: $\circ$ denotes the true inhomogeneity location and $\times$ the estimated location. (b) Optical diffusion tomography reconstruction of μ_a. Parameters: 5 sources and 5 detectors and background parameters as in Fig. 3.1; inhomogeneity $\mu_a = 0.12$ cm^{-1}, $D = 0.03$ cm; average SNR is 40 dB; spherical inhomogeneity diameter of 0.625 cm. 41

3.3 Influence of inhomogeneity depth, size and optical contrast ($\Delta\mu_a$) on P_D for the geometry and parameters shown in Fig. 3.1, with $P_F = 0.03$ and an average SNR of 40 dB. (a) P_D as a function of depth, for an inhomogeneity having: diameter 0.625 cm, $\mu_a = 0.12$ cm^{-1}, and $D = 0.03$ cm. (b) P_D as a function of size and $\Delta\mu_a$, with $d = 1.5$ cm. (c) P_D as a function of depth and $\Delta\mu_a$, with a 0.625 cm diameter inhomogeneity. (d) P_D as a function of depth and size, with $\Delta\mu_a = 0.1$ cm^{-1}. 44

Figure Page

3.4 Detection sensitivity as a function of S-D distance for two inhomogeneity depths. The background optical parameters are: $\mu_{a0} = 0.1$ cm^{-1}, $D_0 = 0.03$ cm, which give $k = 0.9$ cm^{-1}. The sensitivity for inhomogeneity depth 3 cm is magnified 20 times. The points are the approximate solution from (3.9). 45

4.1 (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. 52

4.2 Pseudo-code of the greedy algorithms for the SMT covariance estimation. Notations of the operators follows the style of Matlab. (a) Algorithm I where $n > p$ or $n \approx p$. In this case, the sample covariance is explicitly constructed. (b) Algorithm II where $n \ll p$. In this case, the sample covariance is computed on-the-fly to save memory. 58

4.3 (a) Simulated color IR view of an airborne hyperspectral data over the Washington DC Mall [79]. (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. 64

4.4 Plot of the average log-likelihood as a function of the number of Givens rotations K in cross validation. The value of K that achieves the highest average log-likelihood is choosen as the number of rotations in the final SMT covariance estimator. $K = 495$ in this example. 65

4.5 Kullback-Leibler distance from true distribution versus sample size for various classes: (a) (b) (c) Gaussian case (d) (e) (f) non-Gaussian case. 66

4.6 The distribution of estimated eigenvalues for the grass class with $n = 80$: (a) Gaussian case (b) Non-Gaussian case. 66

4.7 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. 70

Figure Page

4.8 Experimental results of eigen-image analysis for $n = 80$ and 28×23 thumbnail face images from the face image database [83]. (a) Example face image samples. First 80 eigen-images for each of the following methods: (b) Diagonal covariance estimate (i.e. independent pixels); (c) Shrinkage of sample covariance to diagonal; (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. 71

4.9 Plot of the average log-likelihood as a function of the number of Givens rotations K in cross validation. The value of K that achieves the highest average log-likelihood is choosen 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. 72

4.10 Generated face image samples under the Gaussian distribution with the sample mean and different covariance estimates: (a) Diagonal (b) Shrinkage (c) Glasso (d) SMT (e) SMT-S. 73

4.11 (a) The graph shows the average cross-validated log-likelihood of the face images using the diagonal, shrinkage, glasso, SMT and SMT-S covariance estimators. (b) The table shows the cross-validated log-likelihood for each estimator. 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). 74

5.1 Broadband image of the 224-channel hyperspectral AVIRIS data used in the experiments. The image is 150×500 pixels, and was obtained from flight $f960323t01p02_r04_sc01$ over the Florida coast. 81

5.2 Average of SCRR as a function of sample size for (a,b,c) Florida image and (d,e,f) Washington image. In all cases, the target signals are randomly generated from a Gaussian distribution, and the error bars are based on runs with 30 trials. (a,d) Gaussian samples are generated from the "true" covariance matrices for these two images. (b,e) Non-Gaussian samples are drawn at random with replacement from the image data itself. (c,f) Gaussian samples generated from randomly rotated covariance matrices. All plots are based on 30 trials, and each trial used a different rotation (for the randomly rotated covariances) and a different target $\mathbf{t}$. 82

Figure Page

5.3 Non-rotationally-invariant structure in the covariance matrix of real hyperspectral data is evident in the image of eigenvectors for (a,b,c,d) Florida data and for (e,f,g,h) Washington data. In (a,e) the covariance matrix is shown with larger values of R_{ij} plotted darker; in (b,f) the matrix E of eigenvectors of R is shown with larger values of the absolute value $|E_{ij}|$ shown darker; in (c,g) the eigenvectors are shown for a randomly rotated covariance matrix; and in (d,h) a histogram of eigenvector values is shown for both the original and the randomly rotated covariance matrix. . . . 84

5.4 Same as Fig. 5.3 but for simulated data. $p = 200$ channels, and 75000 pixels. The data were generated in such a way that each pixel is independently chosen from a uniform distribution so that the mth channel has distribution in the range $[0, r_m]$ where r_m is itself a number chosen in the range $[0, 1]$. 84

6.1 (a) Simulated color IR view of an airborne hyperspectral data over the Washington DC Mall [79]. (b) Ground-truth pixel spectrum of grass. (c) Ground-truth pixel spectrum of water. 93

6.2 Plots of average SNR when τ is the 170-th eigenvector of R_w. Notice that SMT-Lasso regression results in the highest SNR in the range of $n < p$. (a) Clutter W is generated using hyperspectral grass data. (b) Clutter W is generated using hyperspectral water data. 94

6.3 SMT-Lasso versus SMT-Shrinkage versus SMT-Subset when τ is the 170-th eigenvector of R_w. SMT-Lasso works best, but are much more computationally expensive. (a) Clutter W is generated using hyperspectral grass data. (b) Clutter W is generated using hyperspectral water data. . . . 95

6.4 Plots of average SNR when τ is a random Gaussian signal. Notice that SMT-Lasso regression results in consistently higher SNR in the range of $n < p$ compared to the other regression methods. (a) Clutter W is generated using hyperspectral grass data. (b) Clutter W is generated using hyperspectral water data. 95

6.5 SMT-Lasso versus SMT-Shrinkage versus SMT-Subset when τ is a random Gaussian signal. SMT-Lasso works best, but is more computationally expensive. (a) Clutter W is generated using hyperspectral grass data. (b) Clutter W is generated using hyperspectral water data. 96

ABSTRACT

Cao, Guangzhi Ph.D., Purdue University, December 2009. Modeling and Processing of High Dimensional Signals and Systems Using the Sparse Matrix Transform . Major Professors: Charles A. Bouman and Kevin J. Webb.

In this work, a set of new tools is developed for modeling and processing of high dimensional signals and systems, which we refer to as the sparse matrix transform (SMT). The SMT can be viewed as a generalization of the FFT and wavelet transforms in that it uses "butterflies" for efficient implementation. However, unlike the FFT and wavelet transforms, the design of the SMT is adapted to data, and therefore it can be used to process more general non-stationary signals.

To demonstrate the potential of the SMT, we first show how the non-iterative maximum a posteriori (MAP) reconstruction can be made possible for tomographic systems using the SMT and a novel matrix source coding theory. In fact, for a class of difficult optical tomography problems, this non-iterative MAP reconstruction can reduce both computation and storage by well over two orders of magnitude.

The SMT can also be used for accurate covariance estimation of high dimensional data vectors from a limited number of samples ("small n, large p"). Experiments on standard hyperspectral data and face image sets show that the SMT covariance estimation is consistently more accurate than alternative methods. This has also resulted in successful applications of the SMT for weak signal detection in hyperspectral imagery and eigen-image analysis.

We conclude by proposing a novel approach to high dimensional regression using the SMT, and we demonstrate that the new approach can significantly improve prediction accuracy as compared to traditional regression methods.

1. INTRODUCTION

As our capability of data measurement and collection keeps increasing, high dimensional signals and systems are becoming more and more popular nowadays. Medical imaging systems (e.g. CT, MRI), hyperspectral imagery, Internet consumer data, financial data are just a few examples of such signals and systems. The explosive growth of signal and system's dimensionality exposes us to an unprecedented amount of data and potential information; however, there is still a need for a general set of mathematical and statistical tools for modeling and processing such high dimensional signals and systems similar to the fast Fourier transform (FFT) and wavelet transforms for traditional stationary signals and systems. On one side, classical methods based on singular/eigen value analysis are computationally prohibitively expensive in such high dimensional spaces and hence usually not feasible. On the other side, in a lot of scenarios the growth of data dimensionality (p) is much faster than the number of available observations (n), which contradicts with the basic assumption of $n \gg p$ in the classic statistics literature and is usually referred as "small n, large p" problems. Therefore, high dimensional signals and systems pose not only a computational challenge, but also an inference challenge. This is commonly referred to as the "curse of dimensionality".

In this dissertation, we will introduce our work in modeling and processing of high dimensional signals and systems. The sparse matrix transform (SMT) is a tool that we developed for such purposes. The SMT is a generalization of the FFT and wavelet transforms in that it uses "butterflies" for efficient implementation. However, unlike the FFT and wavelet transforms, the butterfly pattern of the SMT is designed adaptively based on data, and therefore it can be used to process more general non-stationary signals and systems. In this work, we specifically focus on the applications

of the SMT in tomographic reconstruction, covariance estimation, signal detection and regression, all in high dimensional spaces.

In Chapter 2, we present a method for non-iterative maximum *a posteriori* (MAP) tomographic reconstruction which is based on the use of sparse matrix representations. Our approach is to pre-compute and store the inverse matrix required for MAP reconstruction. This approach has generally not been used in the past because the inverse matrix is typically large and dense. In order to overcome this problem, we introduce two new ideas. The first idea is a novel theory for the lossy source coding of matrix transformations which we refer to as matrix source coding. This theory is based on a distortion metric that reflects the distortions produced in the final matrix-vector product, rather than the distortions in the coded matrix itself. The resulting algorithms are shown to require orthonormal transformations of both the measurement data and the matrix rows and columns before quantization and coding. The second idea is a method for efficiently storing and computing the required orthonormal transformation using the sparse matrix transform (SMT). The SMT can be numerically designed to best approximate the desired transforms. We demonstrate the potential of the non-iterative MAP reconstruction with examples from optical tomography. The method requires off-line computation to encode the inverse transform. However, once these off-line computations are completed, the non-iterative MAP algorithm is shown to reduce both storage and computation by well over 2 orders of magnitude, as compared to a linear iterative reconstruction method.

In chapter 3, an approach for fast localization and detection of an absorbing inhomogeneity in a tissue-like scattering medium based on the diffusion model is presented as an alternative to volumetric reconstruction. The probability of detection as a function of the size, location, and absorptive properties of the inhomogeneity is investigated. The detection sensitivity in relation to the source and detector location can be used to serve a basis for instrument design.

In Chapter 4, we propose a maximum likelihood (ML) approach to covariance estimation for high dimensional data vectors, which is a classically difficult problem

in statistical analysis and machine learning. More specifically, the covariance is constrained to have an eigen decomposition which can be represented as the sparse matrix transform (SMT). Using this framework, the covariance can be efficiently estimated using greedy optimization of the log-likelihood function, and the number of Givens rotations in the SMT can be efficiently computed using a cross-validation procedure. The resulting estimator is positive definite and well-conditioned even when the sample size is limited. Experiments on 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.

In Chapter 5, we investigate the utility of the sparse matrix transform (SMT) for weak signal detection in hyperspectral imagery. Many detection algorithms in hyperspectral image analysis, from well-characterized gaseous and solid targets to deliberately uncharacterized anomalies and anomalous changes, depend on accurately estimating the covariance matrix of the background. The accuracy of the SMT covariance estimate can lead to a better detector and hence a gain of detection power. Experiments on hyperspectral data show that using the SMT to estimate the covariance matrix in the adaptive matched filter leads to consistently higher signal-to-clutter ratios than other regularization methods.

In Chapter 6, we propose a novel approach to high dimensional regression for applications when $n < p$. The approach works by first decorrelating the high dimensional observation vectors using the sparse matrix transform (SMT) estimate of the data covariance. Then the decorrelated observations are used in a regularized regression procedure such as Lasso or shrinkage. Numerical results demonstrate that the proposed regression approach can significantly improve the prediction accuracy, especially when n is small and the signal to be predicted lies the subspace of the observations corresponding to the small eigenvalues.

2. NON-ITERATIVE MAP RECONSTRUCTION USING SPARSE MATRIX REPRESENTATIONS

2.1 Introduction

Sparsity is of great interest in signal processing due to its fundamental role in efficient signal representation. In fact, sparse representations are essential to data compression methods, which typically use the Karhunen-Loeve (KL) transform, Fourier transform, or wavelet transform to concentrate energy in a few primary components of the signal [1, 2]. Recently, there has been increasing interest in exploiting sparsity in the data acquisition process through the use of coded aperture or compressed sensing techniques [3–6]. The key idea in these approaches is that the sparsity of data in one domain can lead to a reduced sampling rate in another domain.

Interestingly, little work has been done on the sparse representation of general transforms which map data between different domains. Nonetheless, the sparse representation of transforms is important because many applications, such as iterative image reconstruction and de-noising, require the repeated transformation of high dimensional data vectors. Although sparse representation of some special orthonormal transforms, such as the discrete Fourier transform and discrete wavelet transform [7–9], have been widely studied, there is no general methodology for creating sparse representations of general dense transforms.

Sparsity is of particular importance in the inversion of tomographic data. The forward operator of computed tomography (CT) can be viewed as a sparse transformation; and reconstruction algorithms, such as filtered back projection, must generally be formulated as sparse operators to be practical. In recent years, iterative reconstruction using regularized inversion [10, 11] has attracted great attention because it can produce substantially higher image quality by accounting for both the sta-

tistical nature of measurements and the characteristics of reconstructed images [12]. For example, maximum *a posteriori* (MAP) reconstruction works by iteratively minimizing a cost function corresponding to the probability of the reconstruction given the measured data [13–15]. Typically, the MAP reconstruction is computed using gradient-based iterative optimization methods such as the conjugate gradient method. Interestingly, when the prior model and system noise are Gaussian, the MAP reconstruction of a linear or linearized system is simply a linear transformation of the measurements. However, even in this case, the MAP reconstruction is usually not computed using a simple matrix-vector product because the required inverse matrix is enormous (number of voxels by the number of measurements) and is also generally dense. Consequently, both storing the required matrix and computing the matrix-vector product are typically not practical.

In this paper, we introduce a novel approach to MAP reconstruction based on our previous work [16, 17], in which we directly compute the required matrix-vector product through the use of a sparse representation of the inverse matrix. In order to make the large and dense inverse matrix sparse, we introduce two new ideas.

The first idea is a novel theory for the lossy source coding of matrix transformations, which we refer to as matrix source coding. Source coding of matrix transforms differs from source coding of data in a number of very important ways. First, minimum mean squared error encoding of a matrix transformation does not generally imply minimum mean squared error in a resulting matrix-vector product. Therefore, we first derive an appropriate distortion metric for this problem which reflects the distortions produced in matrix-vector multiplication. The proposed matrix source coding algorithms then require orthonormal transformations of both the measurement data and matrix rows and columns before quantization and coding. After quantization, the number of zeros in the transformed matrix can dramatically increase, making the resulting quantized matrix very sparse. This sparsity not only reduces storage, but it also reduces the computation required to evaluate the matrix-vector product used in reconstruction.

The second idea is a method for efficiently storing and computing the required orthonormal transformations, which we call a sparse-matrix transform (SMT). The SMT is a generalization of the classical fast Fourier transform (FFT) in that it uses butterflies to compute an orthonormal transform; but unlike the FFT, the SMT uses the butterflies in an irregular pattern and is numerically designed to best approximate the desired transforms. Furthermore, we show that SMT can be designed by minimizing a cost function that approximates the bit-rate at low reconstruction distortion, and we introduce a greedy SMT design algorithm which works by repeatedly decorrelating pairs of coordinates using Givens rotations [18].

The SMT is related to both principal component analysis (PCA) [19, 20] and independent component analysis (ICA) [21–24], which sometimes use Givens rotations to parameterize orthonormal transformations. However, the SMT differs from these methods in that it uses a small number of rotations to achieve a fast and sparse transform, thereby reducing computation and storage. In fact, the SMT can be shown to be a generalization of orthonormal wavelet transforms [25], and is perhaps most closely related to the very recently introduced treelet transform in its structure [26]. Moreover, we have recently shown that the SMT can be used for maximum-likelihood PCA estimation [27].

Our non-iterative MAP approach requires an off-line computation in which the inverse transform matrix is compressed and encoded. However, once this off-line computation is complete, the on-line reconstruction consists of a very fast matrix-vector computation. This makes the method most suitable for applications where reconstructions are computed many times with different data but the same geometry.

We demonstrate the potential of our non-iterative MAP reconstruction by showing examples of its use for optical diffusion tomography (ODT) [28]. For the ODT examples, which are normally very computationally intensive, the non-iterative MAP algorithm reduces on-line storage and computation by well over 2 orders of magnitude, as compared to a traditional iterative reconstruction method.

2.2 Non-Iterative MAP Reconstruction Framework

Let $x \in \mathbb{R}^N$ denote the image to be reconstructed, $y \in \mathbb{R}^M$ be the surface measurements, and $A \in \mathbb{R}^{M \times N}$ be the linear or linearized forward model, so that

$$y = Ax + w \ , \tag{2.1}$$

where w is zero mean additive noise.

For a typical inverse problem, the objective is to estimate x from the measurements of y. However, direct inversion of A typically yields a poor quality solution due to noise in the measurements and the ill-conditioned or non-invertible nature of the matrix A. For such problems, a regularized inverse is often computed [29], or in the context of Bayesian estimation, the MAP estimate of x is used, which is given by

$$\hat{x} = \arg\max_x \{\log p(y|x) + \log p(x)\} \ , \tag{2.2}$$

where $p(y|x)$ is the data likelihood and $p(x)$ is the prior model for the image x. In some cases, non-negativity is required to make x physically meaningful.

If we assume that w is zero mean Gaussian noise with covariance Λ^{-1}, and that x is modeled by a zero mean Gaussian random field with covariance S^{-1}, then the MAP estimate of x given y is given by the solution to the optimization problem

$$\hat{x} = \arg\min_x \parallel y - Ax \parallel_\Lambda^2 + x^t S x \ . \tag{2.3}$$

Generally, this optimization problem is solved using gradient-based iterative methods. However, iterative methods tend to be expensive both in computational time and memory requirements for practical problems, especially when A is not sparse, which is the case in some important inverse problems such as in optical tomography.

However, if we neglect the possible positivity constraint, then the MAP reconstruction of (2.3) may be computed in closed form as

$$\hat{x} = (A^t \Lambda A + S)^{-1} A^t \Lambda y \ . \tag{2.4}$$

Therefore, if we pre-compute the inverse matrix

$$H \stackrel{\triangle}{=} (A^t \Lambda A + S)^{-1} A^t \Lambda \ , \tag{2.5}$$

then we may reconstruct the image by simply computing the matrix-vector product

$$\hat{x} = Hy \ . \tag{2.6}$$

The non-iterative computation of the MAP estimate in (2.6) seems very appealing since there are many inverse problems in which a Gaussian prior model (i.e. quadratic regularization) is appropriate and positivity is not an important constraint. However, non-iterative computation of the MAP estimate is rarely used because the matrix H can be both enormous and non-sparse for many inverse problems. Even when A is a sparse matrix, H will generally not be sparse. Therefore, as a practical matter, it is usually more computationally efficient to iteratively solve (2.3) using forward computations of Ax, rather than computing Hy once. Moreover, the evaluation of Hy is not only a computational challenge but it is also a challenge to store H for large inverse problems. Our objective is to develop methods for sparse representation of H so that the matrix-vector product of (2.6) may be efficiently computed, and so the matrix H may be efficiently stored.

2.3 Lossy Source Coding of the Inverse Matrix

2.3.1 Encoding Framework

For convenience, we assume that both the columns of H and the measurements y have zero mean. [1] For the 3D tomographic reconstruction problem, the columns of H are 3D images corresponding to the reconstruction that results if only a single sensor measurement is used. Since each column is an image, the columns are well suited for compression using conventional lossy image coding methods [30]. However, the lossy encoding of H will create distortion, so that

$$H = [H] + \delta H \ , \tag{2.7}$$

[1] Let the row vector μ_H be the means of the columns of H, and let $\bar{y}$ be the mean of the measurements. Then the reconstructed image can be expressed as $\hat{x} = (H + \mathbf{1}\mu_H)(y + \bar{y}) = Hy + \mathbf{1}\mu_H y + (H + \mathbf{1}\mu_H)\bar{y}$, where $\mathbf{1}$ denotes the column vector with all elements being 1. Once Hy is computed, the quantity $\mathbf{1}\mu_H y + (H + \mathbf{1}\mu_H)\bar{y}$ can be added to account for the non-zero mean.

where $[H]$ is the quantized version of H and δH is the quantization error. The distortion in the encoding of H produces a corresponding distortion in the reconstruction with the form

$$\delta\hat{x} = \delta H y \ . \tag{2.8}$$

2.3.2 Distortion Metric

The performance of any lossy source coding method depends critically on the distortion metric that is used. However, conventional distortion metrics such as the mean squared error (MSE) of H may not correlate well with the MSE distortion in the actual reconstructed image, $\| \delta\hat{x} \|^2$, which is typically of primary concern. Therefore, we would like to choose a distortion metric for H that relates directly to $\| \delta\hat{x} \|^2$.

Assuming the measurement y is independent of the quantization error δH, we can obtain the following expression for the conditional MSE of $\hat{x}$ given δH.

Theorem 2.3.1

$$\mathrm{E}\left[\| \delta\hat{x} \|^2 | \ \delta H\right] = \| \delta H \|^2_{R_y} \ , \tag{2.9}$$

where $R_y = \mathrm{E}[yy^t]$ *and* $\| \delta H \|^2_{R_y} = \mathrm{trace}\{\delta H R_y \delta H^t\}$.

Proof

$$\begin{aligned} \mathrm{E}\left[\| \delta\hat{x} \|^2 | \ \delta H\right] &= \mathrm{E}\left[y^t \delta H^t \delta H y \mid \delta H\right] \\ &= \mathrm{E}\left[\mathrm{trace}\{\delta H y y^t \delta H^t\} \mid \delta H\right] \\ &= \mathrm{trace}\{\delta H \mathrm{E}[yy^t] \delta H^t\} \\ &= \mathrm{trace}\{\delta H R_y \delta H^t\} \end{aligned} \tag{2.10}$$

■

From this result, we have the following immediate corollary.

Corollary 2.3.1 *If* $R_y = I$, *then*

$$\mathrm{E}\left[\| \delta\hat{x} \|^2 \ | \delta H\right] = \| \delta H \|^2 \ . \tag{2.11}$$

Corollary 2.3.1 implies that if the measurements are uncorrelated and have equal variance (i.e. are white), then the reconstruction distortion is proportional to the Frobenius error in the source coded matrix. This implies that it is best to whiten the measurements (i.e. make $R_y = I$) before lossy coding of H, so that the minimum MSE distortion introduced by lossy source coding of H leads to minimum MSE distortion in reconstruction of $\hat{x}$.

In order to whiten the measurement vector y, we first form the eigenvalue decomposition of R_y given by

$$R_y = E\Lambda_y E^t \ , \tag{2.12}$$

where E is a matrix of eigenvectors and Λ_y is a diagonal matrix of eigenvalues. We next define the transformed matrix and whitened data as

$$\tilde{H} = HE\Lambda_y^{\frac{1}{2}} \tag{2.13}$$

$$\tilde{y} = \Lambda_y^{-\frac{1}{2}} E^t y \ . \tag{2.14}$$

Notice that with these definitions $\mathrm{E}[\tilde{y}\tilde{y}^t] = I$, and $\hat{x} = \tilde{H}\tilde{y}$. As in the case of (2.8), the distortion in $\hat{x}$ due to quantization of $\tilde{H}$ may be written as

$$\delta\hat{x} = \delta\tilde{H}\tilde{y} \ , \tag{2.15}$$

where $\delta\tilde{H}$ denotes the quantization error in $\tilde{H}$. Using the result of Corollary 2.3.1 and the fact that $\tilde{y}$ is whitened, we then know that

$$\mathrm{E}[\| \ \delta\hat{x} \ \|^2 | \ \delta\tilde{H}] = \| \ \delta\tilde{H} \ \|^2 \ . \tag{2.16}$$

This means that if we minimize $\| \ \delta\tilde{H} \ \|^2$, we obtain a reconstructed image with minimum MSE distortion.

2.3.3 Transform Coding for $\tilde{H}$

Our next goal is to find a sparse representation for $\tilde{H}$. We do this by decorrelating the rows and columns of $\tilde{H}$ using the orthonormal transformations W^t and Φ. More formally, our goal is to compute

$$\check{H} = W^t \tilde{H} \Phi \ . \tag{2.17}$$

where $\check{H}$ has its energy concentrated in a relatively small number of components.

First notice that if W^t and Φ exactly decorrelate the rows and columns of $\tilde{H}$, then this is essentially equivalent to singular value decomposition (SVD) of the matrix of $\tilde{H}$, with $\check{H}$ corresponding to the diagonal matrix of singular values, and W and Φ corresponding to the left and right singular transforms [31]. In this case, the matrix $\check{H}$ is very sparse; however, the transforms W^t and Φ are, in general, dense, so we save nothing in storage or computation. Therefore, our approach will be to find fast/sparse orthonormal transforms which approximate the exact SVD, thereby resulting in good energy compaction with practical decorrelating transforms.

In this work, we will choose W to be a 3D orthonormal wavelet transform. We do this because the columns of $\tilde{H}$ are 3D images, and wavelet transforms are known to approximate the Karhunen-Loeve (KL) transform for stationary random processes [32]. In fact, this is why wavelet transforms are often used in image source coding algorithms [33, 34]. Therefore, we will see that the wavelet transform approximately decorrelates the rows of the matrix $\tilde{H}$ for our 3D reconstruction problems.

We can exactly decorrelate the columns of $\tilde{H}$ by choosing Φ to be the eigenvector matrix of the covariance $R_{\tilde{H}} = \frac{1}{N} \tilde{H}^t \tilde{H}$. More specifically, we choose Φ so that

$$R_{\tilde{H}} = \Phi \Lambda_{\tilde{H}} \Phi^t \ , \tag{2.18}$$

where Φ is the orthonormal matrix of eigenvectors and $\Lambda_{\tilde{H}}$ is the diagonal matrix of eigenvalues.

In summary, the transformed inverse matrix and data vector are given by

$$\check{H} = W^t H T_{opt}^{-1} \tag{2.19}$$

$$\check{y} = T_{opt} y \ , \tag{2.20}$$

where T_{opt} is defined as

$$T_{opt} = \Phi^t \Lambda_y^{-1/2} E^t \ , \tag{2.21}$$

with E, Λ_y and Φ given in (2.12) and (2.18), respectively. Using $\check{H}$ and $\check{y}$, the reconstruction is computed as

$$\hat{x} = W \check{H} \check{y} \ . \tag{2.22}$$

Finally, the sparse representation $\check{H}$ is quantized and encoded. Since Φ is orthonormal, the vector $\check{y}$ has covariance $\mathrm{E}[\check{y}\check{y}^t] = I$, and by Corollary 2.3.1 minimum MSE quantization of $\check{H}$ will achieve minimum MSE reconstruction of $\hat{x}$. Since the objective is to achieve minimum MSE quantization of $\check{H}$, we quantize each entry of $\check{H}$ with the same quantization step size and denote the resulting quantized matrix as $[\check{H}]$. A variety of coding methods, from simple run-length encoding to the Set Partitioning In Hierarchical Trees (SPIHT) algorithm [35] can be used to entropy encode $\check{H}$, depending on the specific preferences with respect to factors such as computation time and storage efficiency. Of course, it is necessary to first compute the matrix $\check{H}$ in order to quantize and encode it. Appendix A discusses some details of how this can be done.

In summary, the non-iterative reconstruction procedure requires two steps. In the first off-line step, matrix source coding is used to compute the sparse matrix $[\check{H}]$. In the second on-line step, the approximate reconstruction is computed via the relationship

$$\hat{x} \approx W [\check{H}] \, T_{opt} y \ . \tag{2.23}$$

By changing the quantization step size, we can control the accuracy of this approximation, but at the cost of longer reconstruction times and greater storage for $[\check{H}]$. Of course, the question arises of how much computation will be required for the evalua-

tions of the matrix-vector product $T_{opt}y$. This question will be directly addressed in Section 2.4 through the introduction of the SMT transform.

Importantly, matrix source coding is done off-line as a precomputation step, but the operations of equation (2.23) are done on-line during reconstruction. Figure 2.1 illustrates the procedure for the off-line step of matrix source coding, and Fig. 2.2(a) lists a pseudo-code procedure for its implementation. The gray regions of Fig. 2.1 graphically illustrate non-zero entries in the matrices, assuming that the eigenvalues of the KL transforms are ordered from largest to smallest. Notice that the transforms tend to compact the energy in the matrix $[\check{H}]$ into the upper lefthand region.

Figure 2.2(b) lists the pseudo-code for the on-line reconstruction of equation (2.23). Notice, that since the matrix $[\check{H}]$ is very sparse, the computation required to evaluate $[\check{H}]\,\check{y}$ is dramatically reduced. Also, notice that the inverse wavelet transform is only applied once, after $\check{x}$ is computed in order to reduce computation.

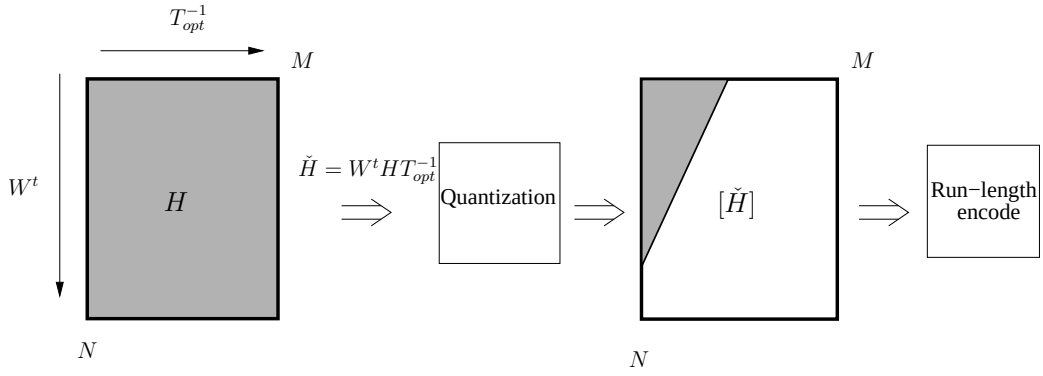


Fig. 2.1. Illustration of matrix source coding procedure. $\check{H}$ is a transformed representation of H using both KL and wavelet transforms. The gray regions represent the effective non-zero entries in the matrices. Notice that the two transforms concentrate the energy in $\check{H}$ toward the upper left hand corner, so that quantization results in a very sparse matrix. This procedure is performed off-line, so that on-line reconstruction can be very fast.

Off-line Processing: Lossy matrix source coding of H

1. Measurement whitening:

$$\begin{aligned}(E, \Lambda_y) &\leftarrow \text{EigenDecomposition}(R_y) \\ \tilde{H} &\leftarrow H E \Lambda_y^{\frac{1}{2}}\end{aligned}$$

2. Decorrelation of the columns of $\tilde{H}$:

$$\begin{aligned}R_{\tilde{H}} &\leftarrow \frac{1}{N}\tilde{H}^t\tilde{H} \\ (\Phi, \Lambda_{\tilde{H}}) &\leftarrow \text{EigenDecomposition}(R_{\tilde{H}}) \\ \check{H} &\leftarrow \tilde{H}\Phi \\ T_{opt} &\leftarrow \Phi^t \Lambda_y^{-\frac{1}{2}} E^t\end{aligned}$$

3. Wavelet transform of each column:

$$\check{H} \leftarrow W^t\check{H} \text{ (or } \check{H} \leftarrow W^t H T_{opt}^{-1})$$

4. Quantization and coding of $\check{H}$:

$$[\check{H}] \leftarrow \text{Quantize}(\check{H})$$

5. Store the coded version of $[\check{H}]$, and the transform matrix T_{opt}.

(a)

On-line Processing: Evaluation of Hy

1. Measurement data transform:$\check{y} \leftarrow T_{opt}y$
2. Decoding of $[\check{H}]$
3. Reconstruction of image in the wavelet domain: $\check{x} \leftarrow [\check{H}]\check{y}$
4. Final reconstructed image after the inverse wavelet transform: $\hat{x} \leftarrow W\check{x}$

(b)

Fig. 2.2. Implementation of non-iterative MAP reconstruction. (a) The off-line processing algorithm for the matrix source coding of the the inverse matrix H. (b) The on-line processing to compute the reconstruction from the matrix-vector product.

2.4 Sparse Matrix Transform

Step 1 of the on-line reconstruction procedure in Fig. 2.2(b) requires that the data vector y be first multiplied by the transform T_{opt}. However, T_{opt} is generally not sparse, so multiplication by T_{opt} requires order M^2 storage and computation. If the number of measurements M is small compared to the number of voxels N, this may represent a small overhead; but if the number of measurements is large, then storage and multiplication by T_{opt} represents a very substantial overhead.

In this section, we develop a general method to approximately whiten the measurements and decorrelate the inverse matrix using a series of sparse matrix transforms (SMT). We will see that the advantage of the SMT is that it can be implemented with many fewer multiplies than multiplication by the exact transform, T_{opt}, while achieving nearly the same result. More specifically, we approximate the exact transform T_{opt} using a product of K sparse matrices, so that

$$T_{opt} \approx \prod_{k=K-1}^{0} T_k = T_{K-1} T_{K-2} \cdots T_0 \ , \tag{2.24}$$

where every sparse matrix, T_k, operates on a pair of coordinate indices (i_k, j_k). Notice that since each T_k operates on only two coordinates, it can be implemented with no more than 4 multiplies. So, if $K \ll M$, then the total computation required for the SMT will be much less than that required for multiplication by T_{opt}. Therefore, our objective will be to design the SMT of (2.24) so that we may accurately approximate T_{opt} with a small number of T_k's.

Since each sparse matrix T_k only operates on the coordinate pair (i_k, j_k), it has a simple structure as illustrated in Fig. 2.3. In general, any such pair-wise transform T_k can be represented in the form

$$T_k = B_k \Lambda_k A_k \ , \tag{2.25}$$

where A_k and B_k are Givens rotations [18], and Λ_k is a diagonal normalization matrix [2]. A Givens rotation is simply an orthonormal rotation in the plane of the two

[2]Note also that (2.25) represents the singular value decomposition of T_k.

coordinates, i_k and j_k. So the matrices A_k and B_k can be represented by the rotation angles θ_k and ϕ_k, respectively. More specifically, A_k and B_k have the form

$$A_k = I + \Theta(i_k, j_k, \theta_k) \tag{2.26}$$

$$B_k = I + \Theta(i_k, j_k, \phi_k) \ , \tag{2.27}$$

where $\Theta(m, n, \theta)$ is defined as

$$[\Theta]_{ij} = \begin{cases} \cos(\theta) - 1 & \text{if } i = j = m \text{ or } i = j = n \\ \sin(\theta) & \text{if } i = m \text{ and } j = n \\ -\sin(\theta) & \text{if } i = n \text{ and } j = m \\ 0 & \text{otherwise} \end{cases} \ . \tag{2.28}$$

Given the form of (2.26) and (2.27), it is clear that multiplication by A_k and B_k should take no more than 4 multiplies corresponding to the four nonzero entries in Θ shown in (2.28). However, we can do better than this. In Appendix B, we show that the SMT of (2.24) can always be rewritten in the form $\prod_{k=K-1}^{0} T_k = S \prod_{k=K-1}^{0} \tilde{T}_k$, where S is a diagonal matrix and each pair-wise sparse transform $\tilde{T}_k$ requires only two multiplies.

In fact, it is useful to view the SMT as a generalization of the FFT [36]. In order to illustrate this point, Fig. 4.1 graphically illustrates the flow diagram of the SMT, with each sparse matrix $\tilde{T}_k$ serving the role of a "butterfly" in the traditional FFT. Using the result of Appendix B, each butterfly requires only 2 multiplies; so a K-butterfly SMT requires a total of $2K + M$ multiplies (including M normalization factors) for a M-dimensional transform.

A conventional M-point FFT can be computed using approximately $K = \frac{M}{2} \log_2 M$ butterflies. Therefore, one might ask how many butterflies are required for an SMT to compute a desired orthonormal transform? It is known that an arbitrary $M \times M$ orthonormal transform can be computed using $\binom{M}{2}$ butterflies [37], which means that the exact SMT implementation of a general orthonormal transform requires M^2 multiplies, the same as a conventional matrix-vector product.

However, our objective will be to use a much lower order SMT to adequately approximate T_{opt}. Later, we will show that $K = M \log_2 M$ butterflies can be used to accurately approximate the ideal KL transforms for some important example applications. Thus, we argue that the SMT can serve as a fast approximate transform for some important applications which require non-traditional or data dependent orthonormal transforms, such as the KL transform.

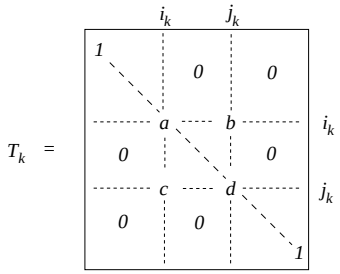


Fig. 2.3. The structure of a pair-wise sparse transform T_k. Here, all the unlabeled diagonal elements are 1's, and all the unlabeled off-diagonal elements are 0's. A_k, B_k and $\tilde{T}_k$ have similar structures.

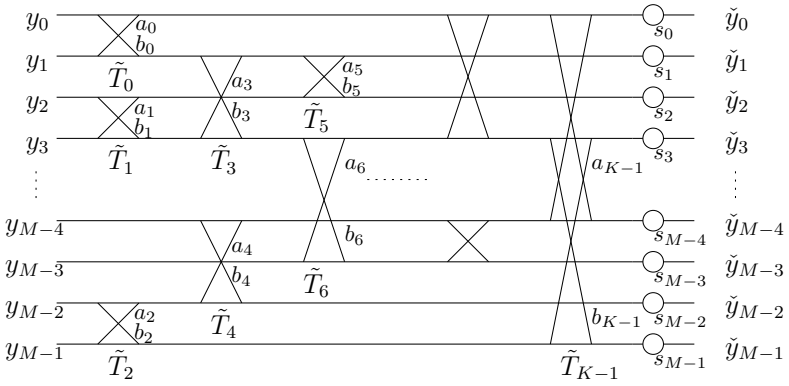


Fig. 2.4. The structure of SMT implementation. Every $\tilde{T}_k$ is a "butterfly" that can be computed using 2 multiplies. In addition, multiplications by normalization factors is required in the end for a total of $2K + M$ multiplies when K butterflies are used in an M-point transform. The irregular structure of the SMT makes it a generalization of the FFT and allows it to be used to accurately approximate a general orthonormal transform.

2.4.1 Cost Function for SMT Design

In order to accurately approximate T_{opt} by the SMT transform of (2.24), we will formulate a cost function whose value is related to the increased bit-rate or distortion incurred by using the SMT transform in place of T_{opt}. The SMT is then designed using greedy minimization of the resulting cost function.

In order to derive the desired cost function, we first generalize the definitions of $\check{H}$ and $\check{y}$ from (2.19) and (2.20) as

$$\check{H} = W^t H T^{-1} \Lambda^{\frac{1}{2}} \tag{2.29}$$

$$\check{y} = \Lambda^{-\frac{1}{2}} T y \ , \tag{2.30}$$

where $T = \prod_{k=K-1}^{0} T_k$ is the SMT transform of (2.24), and $\Lambda = \mathrm{diag}(T R_y T^t)$. First notice, that Λ is defined so that the variance of each component of $\check{y}$ is 1. Second notice, that when $T = T_{opt}$, then (2.29) and (2.30) reduce to (2.19) and (2.20) because $\Lambda = I$ in this case; and as before, the image can be exactly reconstructed by computing $\hat{x} = W \check{H} \check{y}$.

The disadvantage of using $T \neq T_{opt}$ is that the columns of $\check{H}$ and the components of $\check{y}$ will be somewhat correlated. This remaining correlation is undesirable since it may lead to inferior compression of $\check{H}$. We next derive the cost function for SMT design by approximating the increased rate due to this undesired correlation. Using (2.29) and (2.30), the covariance matrices for $\check{y}$ and $\check{H}$ are given by

$$R_{\check{y}} = \mathrm{E}[\check{y}\check{y}^t] = \Lambda^{-\frac{1}{2}} T R_y T^t \Lambda^{-\frac{1}{2}} \tag{2.31}$$

$$R_{\check{H}} = \frac{1}{N} \check{H}^t \check{H} = \Lambda^{\frac{1}{2}} \left(T^{-1}\right)^t R_H T^{-1} \Lambda^{\frac{1}{2}} \ , \tag{2.32}$$

where $R_H = \frac{1}{N} H^t H$. If T is a good approximation to T_{opt}, then $\check{y}$ will be approximately white, and by Corollary 2.3.1 we have that $\mathrm{E}\left[\| \delta\hat{x} \|^2 | \, \delta\check{H}\right] \approx \| \delta\check{H} \|^2$, where $\delta\check{H}$ is the quantization error in $\check{H}$.

Our objective is then to select a transform T which minimizes the required bit-rate (i.e. the number of bits per matrix entry) at a given expected distortion $\mathrm{E}[\| \delta\check{H} \|^2]$. To do this, we will derive a simplified expression for the distortion. In information

theory, we know that if $X_i \sim N(0, \sigma_i^2),\ i = 0, 1, 2, ..., M-1$ are independent Gaussian random variables, then the rate and distortion functions for encoding the X_i's is given by [38]

$$R(\lambda) = \sum_{i=0}^{M-1} \frac{1}{2} \max\left\{0, \log_2\left(\frac{\sigma_i^2}{\lambda}\right)\right\} \tag{2.33}$$

$$D(\lambda) = \sum_{i=0}^{M-1} \min\left\{\sigma_i^2, \lambda\right\} \ , \tag{2.34}$$

where we assume MSE distortion and λ is an independent parameter related to the square of the quantization step size. Since the wavelet transform approximately decorrelates the rows of $\check{H}$, we can model the rows as independent Gaussian random vectors, each with covariance $R_{\check{H}}$. However, we further assume that the encoder quantizes elements of the matrix independently, without exploiting the correlation between elements of a row. In this case, the rate-distortion performance is given by

$$R(\lambda) = N \sum_{i=0}^{M-1} \frac{1}{2} \max\left\{0, \log_2\left(\frac{R_{\check{H}ii}}{\lambda}\right)\right\} \tag{2.35}$$

$$D(\lambda) = N \sum_{i=0}^{M-1} \min\left\{R_{\check{H}ii}, \lambda\right\} \ , \tag{2.36}$$

where $D(\lambda) = \mathrm{E}[\| \delta\check{H} \|^2]$. If the distortion is sufficiently low so that $\lambda < \min(R_{\check{H}ii})$, then (2.36) reduces to $D(\lambda) = NM\lambda$. In this case, the rate in (2.35) can be expressed as the following function of distortion

$$R(D) = \frac{N}{2}\left\{\log_2 \mid \mathrm{diag}(R_{\check{H}}) \mid -M \log_2\left(\frac{D}{NM}\right)\right\} \ . \tag{2.37}$$

Therefore, minimization of $\mid \mathrm{diag}\left(R_{\check{H}}\right) \mid$ corresponds to minimizing the bit-rate required for independently encoding the columns of $\check{H}$ at low distortion. Consequently, our objective is to minimize the cost function $C(T, R_H, R_y)$, defined by

$$C(T, R_H, R_y) = \mid \mathrm{diag}\left(R_{\check{H}}\right) \mid \ . \tag{2.38}$$

Substituting in the expression for $R_{\check{H}}$ from (2.32) and the definition $\Lambda = \mathrm{diag}(TR_yT^t)$ into (2.38) yields

$$C(T, R_H, R_y) = \mid \mathrm{diag}\left(\left(T^{-1}\right)^t R_H T^{-1}\right) \mid \cdot \mid \mathrm{diag}\left(TR_yT^t\right) \mid \ . \tag{2.39}$$

The cost function of (2.39) is also justified by the fact that the exact transform T_{opt} of (2.21) achieves its global minimum (see Appendix C). Our goal is then to find a sparse matrix transform T that minimizes the cost function $C(T, R_H, R_y)$.

2.4.2 SMT Design Using Greedy Minimization

In this subsection, we show how the SMT may be designed through greedy minimization of the cost function of (2.39). More specifically, we will compute the sparse matrices T_k in sequence starting from $k = 0$ and continuing to $k = K - 1$. With each step, we will choose T_k to minimize the cost function while leaving previous selections of T_i for $i < k$ fixed.

Our ideal goal is to find T^* such that

$$T^* = \arg \min_{T=\prod_{k=K-1}^{0} T_k} \{| \operatorname{diag}\left((T^{-1})^t R_H T^{-1}\right) | \cdot | \operatorname{diag}\left(T R_y T^t\right) |\} \ , \tag{2.40}$$

where each T_k is a pair-wise sparse transform. Given that we start with the covariance matrices R_H and R_y, the k-th iteration of our greedy optimization method uses the following three steps:

$$T_k^* \leftarrow \arg \min_{T_k} \{| \operatorname{diag}\left((T_k^{-1})^t R_H T_k^{-1}\right) | \cdot | \operatorname{diag}\left(T_k R_y T_k^t\right) |\} \tag{2.41}$$

$$R_H \leftarrow \left(T_k^{*-1}\right)^t R_H T_k^{*-1} \tag{2.42}$$

$$R_y \leftarrow T_k^* R_y \left(T_k^*\right)^t \ , \tag{2.43}$$

where "$\leftarrow$" indicates assignment of a value in pseudocode. For a specified coordinate pair (i, j), the cost function in (2.41) is minimized when both the measurement pair (y_i, y_j) and the (i, j)-th columns of H are decorrelated. Appendix D gives the solution of (2.41) for this case and also the ratio of the minimized cost function to its original value which is given by

$$\left(1 - \frac{R_{yij}^2}{R_{yii} R_{yjj}}\right)\left(1 - \frac{R_{Hij}^2}{R_{Hii} R_{Hjj}}\right) \ , \tag{2.44}$$

where i and j are the indices corresponding to the pair-wise transform of T_k. Therefore, with each iteration of the greedy algorithm, we select the coordinate pair (i_k, j_k)

that reduces the cost in (2.41) most among all possible pairs. The coordinate pair with the greatest cost reduction is then

$$(i_k, j_k) \leftarrow \arg\min_{(i,j)} \left\{ \left(1 - \frac{R_{yij}^2}{R_{yii}R_{yjj}}\right) \left(1 - \frac{R_{Hij}^2}{R_{Hii}R_{Hjj}}\right) \right\} \ . \tag{2.45}$$

Once i_k and j_k are determined, $T_k^* = B_k \Lambda_k A_k$ can be obtained by computing A_k, Λ_k and B_k, as derived in Appendix D. Specifically, we first normalize the variances of the components of y to 1, as shown in line 2 of Fig. 2.5. Then as shown in Appendix D, A_k is given by

$$A_k = I + \Theta(i_k, j_k, \theta_k) \ , \text{ where } \theta_k = \frac{\pi}{4} \ ; \tag{2.46}$$

Λ_k is given by

$$[\Lambda_k]_{ij} = \begin{cases} 1/\sqrt{1 + R_{yi_k j_k}} & \text{if } i = j = i_k \\ 1/\sqrt{1 - R_{yi_k j_k}} & \text{if } i = j = j_k \\ 1 & \text{if } i = j \neq i_k \text{ and } i = j \neq j_k \\ 0 & \text{if } i \neq j \end{cases} \ ; \tag{2.47}$$

and

$$B_k = I + \Theta(i_k, j_k, \phi_k) \ , \tag{2.48}$$

where

$$\phi_k = \frac{1}{2} \text{atan} \left((R_{Hj_k j_k} - R_{Hi_k i_k}) \sqrt{1 - R_{yi_k j_k}^2}, (R_{Hi_k i_k} + R_{Hj_k j_k}) R_{yi_k j_k} + 2R_{Hi_k j_k} \right) \tag{2.49}$$

and $\text{atan}(\cdot, \cdot)$ denotes the four quadrant arctangent function. [3] The final SMT operator is then given by

$$T^* = \prod_{k=K-1}^{0} B_k \Lambda_k A_k = \prod_{k=K-1}^{0} T_k^* \ . \tag{2.50}$$

Figure 2.5 shows the pseudo-code for the greedy SMT design. Naive implementation of the design algorithm requires M^2 operations for the selection of each Givens rotation. This is because it is necessary to find the the two coordinates, i_k and j_k, that minimize the criteria of equation (2.45) with each iteration. However, this operation

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

can be implemented in order M time by storing the minimal values of the criteria for each value of the index i. At the end of each iteration, these minimum values can then be updated with order M complexity. Using this technique, SMT design has a total complexity of order $M^2 + MK$ for known R_y and R_H.

$$
\begin{aligned}
&\Lambda_y \leftarrow \mathrm{diag}(R_y) \\
&R \leftarrow \Lambda_y^{-1/2} R_y \Lambda_y^{-1/2} \\
&C \leftarrow \Lambda_y^{1/2} R_H \Lambda_y^{1/2} \\
&\text{For } k = 0 : K-1 \ \{ \\
&\quad (i_k, j_k) \leftarrow \arg\min_{i<j} \left\{ \left(1 - R_{ij}^2\right) \cdot \left(1 - \frac{C_{ij}^2}{C_{ii} C_{jj}}\right) \right\} \\
&\quad \theta_k \leftarrow \pi/4 \\
&\quad A_k \leftarrow I + \Theta(i_k, j_k, \theta_k) \\
&\quad \Lambda_k \leftarrow I \\
&\quad [\Lambda_k]_{i_k, i_k} \leftarrow 1/\sqrt{1 + R_{i_k j_k}} \\
&\quad [\Lambda_k]_{j_k, j_k} \leftarrow 1/\sqrt{1 - R_{i_k j_k}} \\
&\quad \phi_k \leftarrow \frac{1}{2} \mathrm{atan}\left((C_{j_k j_k} - C_{i_k i_k}) \sqrt{1 - R_{i_k, j_k}^2}, (C_{j_k j_k} + C_{i_k i_k}) R_{i_k j_k} + 2 C_{i_k j_k} \right) \\
&\quad B_k \leftarrow I + \Theta(i_k, j_k, \phi_k) \\
&\quad T_k \leftarrow B_k \Lambda_k A_k \\
&\quad R \leftarrow T_k R T_k^t \\
&\quad C \leftarrow \left(T_k^{-1}\right)^t C T_k^{-1} \\
&\} \\
&T \leftarrow \left(\prod_{k=K-1}^{0} T_k \right) \Lambda_y^{-\frac{1}{2}}
\end{aligned}
$$

Fig. 2.5. Pseudo-code implementation of the greedy algorithm used for the SMT design.

2.4.3 Relation of SMT to PCA and ICA

The SMT has an interesting relationship to methods which have been used in PCA and ICA signal analysis. In fact, Givens rotations have been used as a method to parameterize the orthonormal transforms used in both these methods [19–21, 39, 40]. However, these applications use $\binom{M}{2}$ or more Givens rotations to fully parameterize the set of all orthonormal transforms. In the case of the SMT, the number of Givens rotations is limited so that the transform can be computed with a small number of multiplies and can be stored with much less than M^2 values. In practice, we have found that K can be chosen as as constant multiple of M in many applications, so the resulting SMT can be computed with order M complexity. While ICA methods often use a cost minimization approach, the cost functions are typically designed for the very different application of minimizing the dependence of data.

The SMT is perhaps most closely related to the recently introduced treelet [26] in that both transforms use a small number of Givens rotations to obtain an efficient/sparse matrix transform. However, the treelet is constrained to a hierarchical tree structure and uses at most $M - 1$ Givens rotations. Also, it is constructed using a selection criteria for each rotation instead of global cost optimization framework. More recently, we have shown that a minor modification of the cost function we propose for SMT design can also be used for maximum likelihood PCA estimation [27].

2.5 Numerical Results

In this section, we illustrate the value of our proposed methods by applying them to two inverse problems in optical imaging: optical diffusion tomography (ODT) and fluorescence optical diffusion tomography (FODT). Both these techniques use near infrared (NIR) or visible light to image deep within living tissue [28, 41, 42]. This is done by modeling the propagation of light in the tissue with the diffusion equation [42–44], and then solving the associated inverse problem to determine the

parameters of light propagation in the tissue. Typically, parameters of importance include absorption, μ_a, and diffusivity, D, which is related to the scattering coefficient.

2.5.1 ODT Example ($M \ll N$)

Description of experiment: Figure 2.6 illustrates the geometry of our ODT simulation. The geometry and parameters of this simulation are adapted from [45], where two parallel plates are used to image a compressed breast for the purpose of detecting breast cancer. There are 9 light sources modulated at 70 MHz on one of the plates, and there are 40 detectors on the other. This results in a total of $360 = 9 \times 40$ complex measurements, or equivalently 720 real valued measurements. We treat the region between the two plates as a 3D box with a size of $16 \times 16 \times 6$ cm^3. For both the forward model computation and reconstruction, the imaging domain was discretized into a $65 \times 65 \times 33$ uniform grid having a spatial resolution of 0.25 cm in the $x - y$ plane and 0.1875 cm along the z coordinate. The bulk opitcal parameters were set to $\mu_{a0} = 0.02$ cm^{-1} and $D_0 = 0.03$ cm for both the breast and the outside region in the box, which can be physically realized by filling the box with intralipid that has optical characteristics close to breast tissue [46]. The measurements were generated with a spherical heterogeneity of radius 1 cm present at the position with the xyz coordinate $(5, 8, 3)$ cm. The optical values of the heterogeneity were $\mu_a = 0.12$ cm^{-1} and $D = 0.03$ cm. Additive noise was introduced based on a shot noise model, giving an average SNR of 35.8 dB [47].

For reconstruction, we assumed the bulk optical parameters, μ_{a0} and D_0, were known. Our objective was then to reconstruct the image x, which is a vector containing the change in the absorption coefficients, $\Delta\mu_a(r) = \mu_a(r) - \mu_{a0}$, at each voxel r. Accordingly, y is the measurement perturbation caused by the absorption perturbation x. The measurements, y, and the absorption perturbations, x, are related through the linearized forward model, A. So this yields the relationship that $E[y] = Ax$. Using a Gaussian Markov random field (GMRF) prior model [48] with an empirically

determined regularization parameter and the shot-noise model for noise statistics, we computed the matrix H so that $\hat{x} = Hy$, where $\hat{x}$ is the MAP reconstruction. The covariance matrix of the measurement y was constructed as $R_y = A\mathrm{E}[xx^t]A^t = AA^t$, where an i.i.d. model was used as the covariance matrix of the image. The inverse matrix H had $66 \times 65 \times 33 = 139425$ rows and 720 columns, which required a memory size of 765.9 Mbytes using double precision floats. The inverse matrix was then transformed using the KL transform along the rows and wavelet transform along the columns, as described in Section 2.3. The wavelet transform was constructed with biorthogonal 9/7 tap filters (which are nearly orthonormal) using a symmetric boundary extension [33, 49]. The transformed inverse matrix $\check{H}$ was quantized and coded using a run-length coder (see Appendix E for details). The numerical experiments were run on a 64-bit dual processor Intel machine.

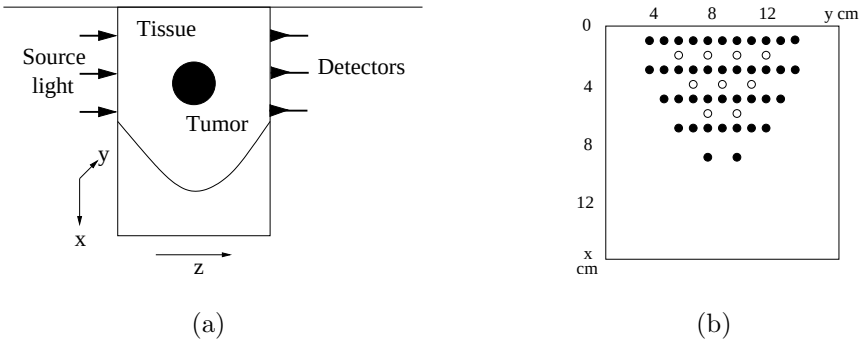
Discussion of experimental results: Figure 2.7 shows the reconstructed images of the absorption $\mu_a(r)$ at $z = 3$ cm using the compressed inverse matrix at different bit-rates where the KL transform is used both for data whitening and matrix decorrelation. The distortion is calculated in terms of the normalized root mean squared error (NRMSE), defined as:

$$\mathrm{NRMSE} = \sqrt{\frac{\parallel [H]y - Hy \parallel^2}{\parallel Hy \parallel^2}}. \tag{2.51}$$

Figure 2.8 shows a plot of the distortion (NRMSE in the reconstructed image) versus the rate (number of bits per matrix entry in H), with different transform methods for data whitening and matrix decorrelation. From Fig. 2.8 we can see that applying the KL transform to both the data and matrix columns dramatically increases the compression ratio as compared to no whitening or decorrelation processing. However, it is interesting to note that simple whitening of the data without matrix column decorrelation works nearly as well. This suggests that data whitening is a critical step in matrix source coding.

Table 2.1 compares the computational complexity of the three methods: iterative MAP using conjugate gradient; non-iterative MAP with no compression; and

non-iterative MAP with the KLT compression. The non-iterative MAP with KLT compression used the KL transform for both data whitening and matrix decorrelation. The compression was adjusted to achieve a distortion of approximately 10% in the reconstructed image, which resulted in a compression ratio of 1808:1 using a run-length coder. The total storage includes both the storage of $[\check{H}]$ (0.4 Mbyte) and the storage of the required transform T_{opt} (4.0 Mbyte). From the table we can see that both the on-line computation time and storage are dramatically reduced using the compressed inverse matrix.

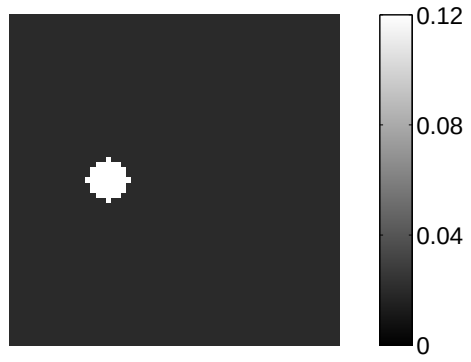


(a) (b)

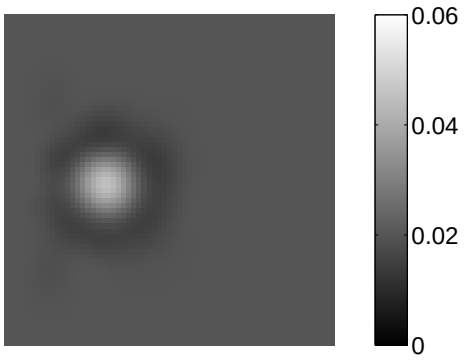
Fig. 2.6. The measurement geometry for optical breast imaging. (a) Imaging geometry. (b) Source-detector probe configuration. The open circles indicate the source fiber locations and the solid circles indicate the detector fiber location. Source fibers and detector fibers are connected to the left and right plates, respectively, and are on 1-cm grid. (Adapted from [45].)

2.5.2 FODT example (M is close to N)

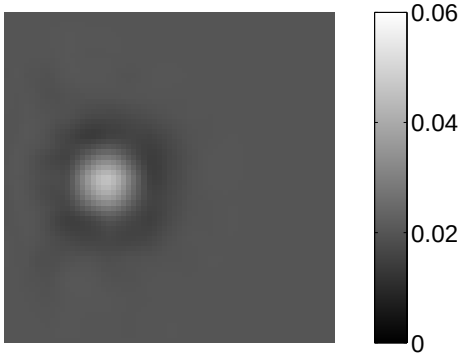
Description of experiment: In the ODT example, the dimension of the measurements M is much less than the dimension of the image to be reconstructed, N, therefore, the overhead required for the computation and storage of the transform matrix T_{opt} is not significant. However, in some cases, the number of measurements may be large. This is especially true in some optical tomography systems where a CCD camera may be used as the detector. In this situation, SMT might be preferred



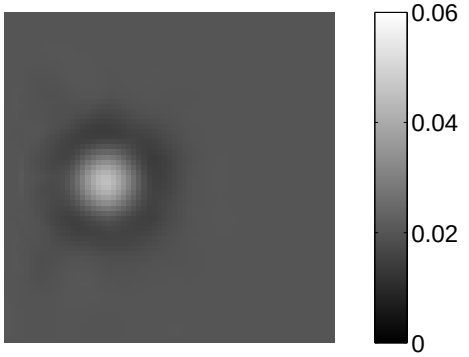
(a) Original Image



(b) Uncompressed



(c) 0.015 bpme, NRMSE = 16.98%



(d) 0.032 bpme, NRMSE = 10.52%

Fig. 2.7. The reconstructed images of $\mu_a(r)$ at $z = 3$ cm using the compressed H matrix based on KL transform (used both for data whitening and matrix decorrelation). The compression ratios in (c) and (d) are 4267:1 and 1982:1, respectively. Here "bpme" stands for bits per matrix entry.

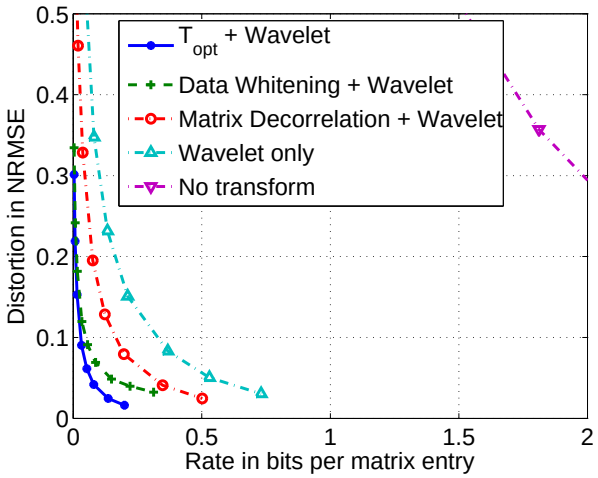


Fig. 2.8. Distortion versus rate for compression using KL transforms for data whitening and matrix decorrelation for the ODT example. Notice here simply whitening the data yields close distortion-rate performance to the theoretically optimal KL transforms. The performance drops significantly with the other three methods that do not perform data whitening.

Table 2.1

Comparison of on-line and off-line computation required by various reconstruction methods for the ODT example. Results use number of voxels $N = 65 \times 65 \times 33 = 139425$, number of measurements $M = 720$ (with number of sources $M_1 = 9$ and number of detectors $M_2 = 40$), number of iterations of CG $I = 30$, a run-length coding compression ratio of $c = 1808 : 1$, and number of iterations required to solve the forward PDE $L = 12$. $NRMSE \approx 10\%$ for the compression case. Notice that the non-iterative MAP reconstruction requires much lower on-line computation and memory than the iterative reconstruction method. However, it requires greater off-line computation to encode the inverse transform matrix.

	On-line Computation		On-line Storage	
	Order	Seconds	Order	Mbytes
Iterative MAP Using Conjugate Grad.	NMI	125.8	NM	765.9
Non-Iterative MAP without Compression	NM	0.89	NM	765.9
Non-Iterative MAP with KLT Compression	$\frac{NM}{c} + N + M^2$	0.03	$\frac{NM}{c} + M^2$	$0.4 + 4.0$ $([\check{H}] + T_{opt})$

	Off-line Computation		Off-line Storage	
	Order	Seconds	Order	Mbytes
Iterative MAP Using Conjugate Grad.	$NM + N(M_1 + M_2)L$	287.6	NM	765.9
Non-Iterative MAP without Compression	NM^2I	9445.4	NM	765.9
Non-Iterative MAP with KLT Compression	$NM^2I + M^3$	$9445.4 + 359.1$ (pre-comp. + coding)	$\max\{NM, M^2\}$	776.4

over the KL transform since the SMT's sparse structure can reduce both storage and computation for the required transform matrix. In order to illustrate the potential of the SMT, we consider the numerical simulation of a fluorescence optical diffusion tomography (FODT) system [42] which uses reflectance measurements. The measurement geometry for this system is shown in Fig. 2.9, where a 6 cm $\times$ 6 cm probe scans the top of a semi-infinite medium. Such a scenario is useful for a real-time imaging application, which would require very fast reconstruction. The probe contains 4 continuous wave (CW) light sources and 625 detectors that are uniformly distributed, as shown in Fig. 2.9(b), resulting in a total of 2500 real measurements. A similar imaging geometry has been adopted for some preliminary *in vitro* studies [50]. The reflectance measurement is clinically appealing, however, it also provides a very challenging tomography problem because it is usually more ill-conditioned than in the case of the transmission measurement geometry. In FODT, the goal is to reconstruct the spatial distribution of the fluorescence yield $\eta\mu_{a_f}(r)$ (and sometimes also the lifetime $\tau(r)$) in tissue using light sources at the excitation wavelength λ_x and detectors filtered at the emission wavelength λ_m.

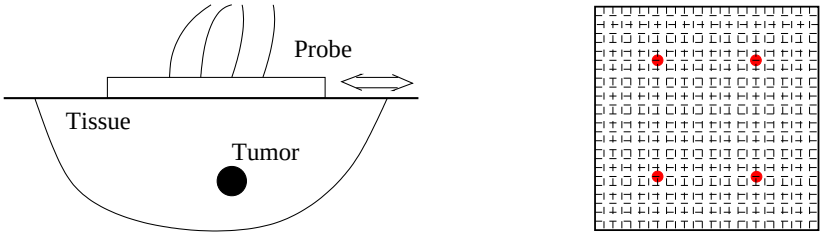
In this example, the bulk optical values were set to $\mu_{a_x} = \mu_{a_m} = 0.02$ cm^{-1} and $D_x = D_m = 0.03$ cm, where the subscripts x and m represent the wavelengths λ_x and λ_m, respectively, and the bulk fluorescence yield was set to $\eta\mu_{a_f} = 0$ cm^{-1}. The measurements were generated with a spherical heterogeneity of radius 0.5 cm present 2 cm below the center of the probe. The optical values of the heterogeneity were $\mu_{a_x} = 0.12$ cm^{-1}, $\mu_{a_m} = 0.02$ cm^{-1}, $D_x = D_m = 0.03$ cm^{-1}, and $\eta\mu_{a_f} = 0.05$ cm^{-1}. The size of the imaging domain is $8 \times 8 \times 4$ cm^3, which was discretized into $33 \times 33 \times 17 = 18513$ voxels, each with an isotropic spatial resolution of 0.25 cm. Additive noise was introduced based on the shot noise model yielding an average SNR of 38.7 dB [47].

For reconstruction, we assumed a homogeneous medium with $\mu_{a_x} = \mu_{a_m} = 0.02$ cm^{-1} and $D_x = D_m = 0.03$ cm set to the values of the bulk parameters. Our objective is to reconstruct the vector x whose elements are the fluorescence yield $\eta\mu_{a_f}(r)$

at individual voxels r. The measure vector y is then composed of the surface light measurements at wavelength λ_m. The two quantities are related by the linear forward model A, so that $E[y] = Ax$. Using a GMRF prior model with an empirically determined regularization parameter and a uniform-variance noise model, we computed the matrix H so that $\hat{x} = Hy$, where $\hat{x}$ is the MAP reconstruction. The covariance matrix of the measurement y was modeled by $R_y = AA^t$, as in the previous example. The inverse matrix H had $33 \times 33 \times 17 = 18513$ rows and $4 \times 625 = 2500$ columns, which required a memory size of 353.1 Mbytes using double precision floats. The inverse matrix was then transformed using the KL transform or SMT along the rows, and a wavelet transform along the columns. The same wavelet transform was implemented as in the ODT example. The transformed inverse matrix $\check{H}$ was quantized and encoded using a run-length coder (see Appendix E for details).

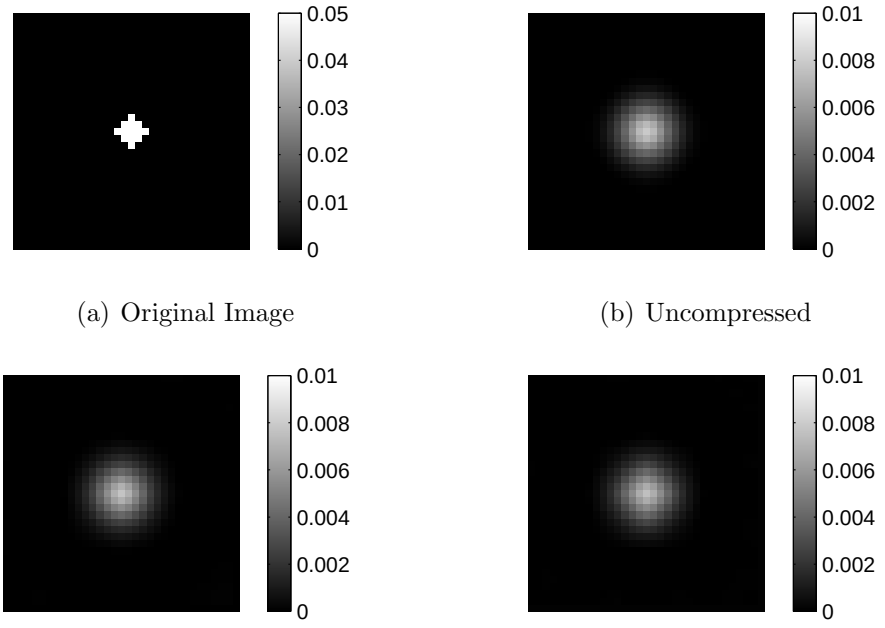
Discussion of experimental results: Figure 2.10 shows the reconstructed images of $\eta\mu_{a_f}(r)$ at a depth of $z = 2$ cm using the compressed inverse matrix based on the KL transform and SMT. The plots of the distortion versus rate based on the KL transform are given in Fig. 2.11(a). Each plot corresponds to a different transform method for data whitening and matrix decorrelation. From the plots, we can see simply whitening y yields a slightly better distortion-rate performance than the theoretically optimal transform, i.e. using the KL transform both for data whitening and matrix decorrelation. This might be caused by inaccurate modeling of the measurement covariance matrix. However, both approaches achieve much better performance than the other three methods where no data whitening was implemented. This again emphasizes the importance of data whitening. Figure 2.11(b) shows the distortion-rate performance where the SMT was used for data whitening and matrix decorrelation. A total number of $M \log_2 M$ SMT butterflies were used to whiten the measurements and decorrelate the columns of H. From the plot, we can see that the SMT results in distortion-rate performance that is very close to the theoretically optimal KL transform, but with much less computation and storage.

Table 2.2 gives a detailed comparison of non-iterative MAP with the KL transform and SMT based compression methods as compared to iterative MAP reconstruction using conjugate gradient optimization. For the KLT method, the KL transform is used both for data whitening and matrix decorrelation with a single stored transform. For this example, the conjugate gradient method required over 100 iterations to converge. The bit-rate for both the compression methods was adjusted to achieve a distortion of approximately 10%, which resulted in a compression ratio of 110:1 for the KL transform and 102:1 for the SMT, both using the same run-length coder. The total storage includes both the storage of $[\check{H}]$ and the storage of the required transform T_{opt} or T as shown explicitly in the table. Notice that the SMT reduces the on-line computation by over a factor of 2 and reduces on-line storage by over a factor of 10, as compared to the KLT. Using more sophisticated coding algorithms, such as SPIHT [35], can further decrease the required storage but at the expense of increased reconstruction time due to the additional time required for SPIHT decompression of the encoded matrix entries.



(a) (b)

Fig. 2.9. The measurement geometry for an FODT example. (a) A graphic depiction of the imaging geometry. (b) An illustration of the source-detector probe where the solid circles indicate the locations of sources and the rectangular grid represents the CCD sensor array.



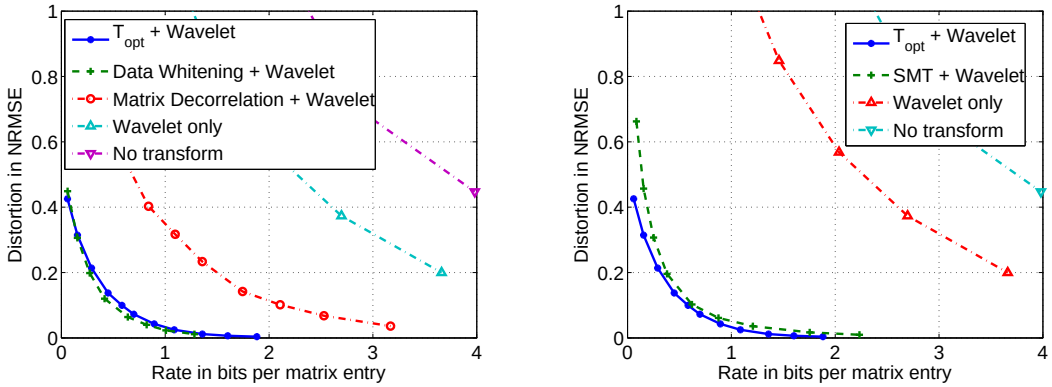
(a) Original Image

(b) Uncompressed

(c) KLT at 0.58 bpme

(d) SMT at 0.62 bpme

Fig. 2.10. The reconstructed images of $\eta\mu_{a_f}(r)$ at the depth of 2 cm using different compression methods. The compression ratios in (c) and (d) are 110:1 and 103:1, and the NRMSE's are 9.96% and 10.24%, respectively.



(a) KLT (b) SMT

Fig. 2.11. Distortion versus rate for the FODT example. (a) Distortion versus rate for compression using the KL transforms for data whitening and matrix decorrelation. (b) Distortion versus rate for compression using the sparse matrix transform (SMT). $M \log_2(M)$ SMT butterflies were used to whiten the measurements and decorrelate the columns of H. Notice that the SMT distortion-rate tradeoff is very close to the distortion-rate of the KL transform.

Table 2.2

Comparison of on-line and off-line computation required by various reconstruction methods for the FODT example. Results use number of voxels $N = 33 \times 33 \times 17 = 18513$, number of measurements $M = 2500$ (with number of sources $M_1 = 4$ and number of detectors $M_2 = 625$), number of iterations of CG $I = 100$, number of sparse rotations $K = M \log(M) = 28220$, a run-length coding compression ratios of $c_1 = 110 : 1$ and $c_2 = 102 : 1$, for KLT and SMT compression, respectively, and number of iterations required to solve the forward PDE $L = 12$. $NRMSE \approx 10\%$ for the compression cases. Notice that the non-iterative MAP reconstruction requires much less on-line computation and memory than the iterative reconstruction method. However, it requires greater off-line computation to encode the inverse transform matrix.

	On-line Computation		Online Storage	
	Order	Seconds	Order	Mbytes
Iterative MAP using Conjugate Grad.	NMI	184.6	NM	353.1
Non-Iterative MAP with No Compression	NM	0.38	NM	353.1
Non-Iterative MAP with KLT Compression	$\frac{NM}{c_1} + N + M^2$	0.08 (0.05 for $T_{opt}y$)	$\frac{NM}{c_1} + M^2$	$3.2 + 47.7$ $([\check{H}] + T_{opt})$
Non-Iterative MAP with SMT Compression	$\frac{NM}{c_2} + N + K$	0.03	$\frac{NM}{c_2} + K$	$3.5 + 0.5$ $([\check{H}] + T)$

	Off-line Computation		Off-line Storage	
	Order	Seconds	Order	Mbytes
Iterative MAP Using Conjugate Grad.	$NM + (M_1 + M_2)NL$	301.8	NM	353.1
Non-Iterative MAP without Compression	NM^2I	15647.2	NM	353.1
Non-Iterative MAP with KLT Compression	$NM^2I + M^3$	$15647.2 + 2282.7$ (pre-comp. + coding)	$\max\{NM, M^2\}$	923.5
Non-Iterative MAP with SMT Compression	$NM^2I + MK$	$15647.2 + 971.3$ (pre-comp. + coding)	$\max\{NM, M^2\}$	372.8

2.5.3 Discussion

From the numerical examples, we see that non-iterative MAP reconstruction can dramatically reduce the computation and memory usage for on-line reconstruction. However, this dramatic reduction requires the off-line pre-computation and encoding of the inverse transform. In our experiments, computation of the inverse matrix dominated off-line computation; so once the inverse transform was computed, it was easily compressed. Moreover, compression of the inverse transform then dramatically reduced storage and computation.

The proposed non-iterative reconstruction methods are best suited for applications where repeated reconstructions must be performed for different data. This could occur in clinical applications where the scanning geometry is fixed, and a new reconstruction is performed with each new scanned data set. The matrix source coding method might also be useful for encoding of the forward transform in iterative reconstruction, particularly if many forward iterations were required.

Once the inverse matrix was computed, the best transforms (KLT for our ODT example, and SMT for our FODT example) resulted in large reductions in computation and storage, as compared to direct storage of the inverse matrix. In particular, matrix source coding reduced computation by 30:1 and 13:1 for the ODT and FODT problems, respectively. And it reduced storage by 174:1 and 88:1, respectively. For these relatively small matrices, computation was ultimately dominated by the overhead required to compute the inverse wavelet transform, but for larger matrices we would expect the computation reduction to approximately equal storage reduction.

Generally, the computational and storage benefits of this method tend to increase with matrix size. Recently, we have begun to investigate the use of matrix source coding for the closely related problem of space-varying deconvolution of digital camera images. For example, a 1 mega pixel digital image can produce an inverse matrix of size $10^6 \times 10^6$. In this case, computational reductions of 10,000:1 are possible [51].

2.6 Conclusion

In this chapter, we presented a non-iterative MAP reconstruction approach for tomographic problems using sparse matrix representations. Compared to conventional iterative reconstruction algorithms, our new method offers much faster and more efficient reconstruction both in terms of computational complexity and memory usage. This makes the new method very attractive for applications. A theory for lossy compression of the inverse matrix with minimum distortion in the reconstruction was developed. Numerical simulations in optical tomography show that compression of the inverse matrix can be quite high, which in turn leads to more efficient computation of the matrix-vector product required for reconstruction. To extend our approach to more general tomography methodologies, we also addressed the problem when the number of measurements is large by introducing the sparse matrix transform (SMT) based on rate-distortion analysis. We demonstrated that the SMT is able to closely approximate orthonormal transforms but with much less complexity through the use of pair-wise sparse transforms.

3. INHOMOGENEITY LOCALIZATION IN A SCATTERING MEDIUM IN A STATISTICAL FRAMEWORK

3.1 Localization versus Reconstruction

Optical imaging in scattering media provides important opportunities for clinical imaging and environmental sensing, among others [28]. In the near-infrared wavelength range, soft tissue has both high scatter and low absorption, allowing use of a diffusion equation model for photon transport [43, 44], which with $\exp(j\omega t)$ time dependence is

$$\left[\nabla \cdot \mathrm{D}(\mathbf{r})\nabla - \mu_a(\mathbf{r}) + \frac{j\omega}{c}\right] \phi(\mathbf{r},\omega) = -\beta\delta(\mathbf{r}-\mathbf{r}_s), \tag{3.1}$$

where ϕ is the photon flux density, ω is the circular modulation frequency, β is the modulation amplitude, c is the speed of light in the intervening medium between the scatterers, μ_a is the absorption coefficient, D is the diffusion coefficient, and a Dirac delta function excitation is assumed. Reconstruction of the unknown optical parameters $\mu_a(\mathbf{r})$ and $D(\mathbf{r})$ requires inversion of measured data, which is formulated as an optimization problem as we have previously shown. This is a computationally intensive process, in large part due to the nonlinear relationship between the cost function and the image parameters. Another difficulty is caused by physical limitations of a practical measurement system, which may result in insufficient information for accurate volumetric imaging. These issues motivate interest in simpler, efficient approaches for detecting and localizing a heterogeneity in a scattering medium instead of quantitative three-dimensional reconstruction.

Methods have been studied for localizing injected fluorophores. Chen *et al.* used least squares curve fitting to compare a diffusion equation model for expected fluorescence with measurements based on the perturbation on a cancellation plane, formed

by dual-interfering sources [52], to localize a fluorophore in a mouse model [53]. Gannot *et al.* used the Levenberg-Marquardt method to fit measured data with a forward model based on random-walk theory for three-dimensional localization of a fluorophore in a mouse tongue [54]. Milstein *et al.* developed a statistical approach based on maximum likelihood (ML) estimation for localization and a binary hypothesis test to detect a fluorescent source [55]. Here, we extend Milstein's work for fluorescence detection to study issues related to the three-dimensional localization and detection of an intrinsic absorbing inhomogeneity in a scattering medium such as tissue [56]. The probability of detection is used to characterize the diagnostic capability of such a measurement system, and the detection sensitivity presented can be used to optimize the source-detector (SD) geometry, thereby providing a path to instrument design. We also investigate how factors such as SD geometry, and the physical and optical properties of the inhomogeneity, affect detection and localization.

3.2 Maximum Likehood Localization

We use ML estimation to estimate the location of an absorbing inhomogeneity in a homogeneous background having parameters μ_{a0} and D_0, which are assumed known. A model is needed to parameterize the unknown inhomogeneity, which could have varying size and optical contrast (defined as $\Delta\mu_a = \mu_a - \mu_{a0}$). We use the point inhomogeneity model suggested by Milstein *et al.* [55], which proved effective in localizing fluorescence, and account for the contrast through a weighting factor for this point absorber, given by $\delta u(\mathbf{r})$. A measurement vector $\mathbf{y}$ of length M, for example, the optical intensity at a series of points on the surface at a particular modulation frequency for the light, is compared with a predicted measurement $\mathbf{f}(\mathbf{r})$, based on (3.1), assuming there exists a point inhomogeneity at position $\mathbf{r}$. Let $\mathbf{y}_0$ represent the expected measurement in the absence of an inhomogeneity and $\mathbf{f}'(\mathbf{r})$ be the Fréchet

derivative which relates perturbations in $\mu_a(\mathbf{r})$ to the predicted measurement $\mathbf{f}(\mathbf{r})$, i.e., $\mathbf{f}(\mathbf{r}) \approx \mathbf{y}_0 + \mathbf{f}'(\mathbf{r})\delta u(\mathbf{r})$. The ML localization can thus be formulated as

$$C(\mathbf{r}, \delta u(\mathbf{r})) = \arg\min_{\mathbf{r}} \parallel \mathbf{y} - \mathbf{y}_0 - \mathbf{f}'(\mathbf{r})\delta u(\mathbf{r}) \parallel^2_{\Lambda}, \tag{3.2}$$

where $C(\mathbf{r}, \delta u(\mathbf{r}))$ is the negative log likelihood and is treated as a cost function, Λ^{-1} is the noise covariance matrix, for which we use a shot noise model [57], and $\parallel \mathbf{v} \parallel^2_{\mathbf{W}} = \mathbf{v}^H\mathbf{W}\mathbf{v}$, with H being the Hermitian transpose. This optimization can be implemented as a two-step procedure in which, for each discretized position $\mathbf{r}$ over the region of interest, $C(\mathbf{r}, \delta u(\mathbf{r}))$ is minimized with respect to $\delta u(\mathbf{r})$, giving the unique (because $C(\mathbf{r}, \delta u(\mathbf{r}))$ is quadratic) closed form estimate

$$\begin{aligned} \delta\hat{u}(\mathbf{r}) &= \arg\min_{\delta u} C(\mathbf{r}, \delta u(\mathbf{r})) \\ &= \frac{\mathrm{Re}[(\mathbf{y} - \mathbf{y}_0)^H \Lambda \mathbf{f}'(\mathbf{r})]}{\parallel \mathbf{f}'(\mathbf{r}) \parallel^2_{\Lambda}}, \end{aligned} \tag{3.3}$$

and then the ML estimate of inhomogeneity location is given by

$$\hat{\mathbf{r}} = \arg\min_{\mathbf{r}} \parallel \mathbf{y} - \mathbf{y}_0 - \mathbf{f}'(\mathbf{r})\delta\hat{u}(\mathbf{r}) \parallel^2_{\Lambda} . \tag{3.4}$$

Fig. 3.1 shows a simulated reflectance measurement geometry with 5 sources and 5 detectors with a separation of 0.5 cm on the top surface of a semi-infinite medium, giving $M = 25$ SD measurement pairs. We consider an 8 cm $\times$ 8 cm $\times$ 8 cm computational domain that is discretized with a grid size of 1.25 mm. The background has $\mu_{a0} = 0.02$ cm^{-1} and $D_0 = 0.03$ cm. An inhomogeneity of diameter 0.625 cm, having $\mu_a = 0.12$ cm^{-1} and $D = 0.03$ cm, is present at depth $d = 1.5$ cm. We assume an average signal-to-noise ratio (SNR) of approximately 40 dB and a modulation frequency of $\omega = 2\pi \times 10^6$ rad/s. An analytic solution of (3.1), with an extrapolated $\phi = 0$ boundary condition to represent the interface between the scattering medium and free space [55], leads to an expression for $\mathbf{f}'(\mathbf{r})$. Fig. 3.2(a) gives a plot of the negative log likelihood, and the estimated centroid of the inhomogeneity is within 2.5 mm of the true point. This is a promising result, given the simple measurement geometry. Fig. 3.2(b) shows the reconstruction of μ_a using the same data set. Note

that the reconstructed μ_a is not accurate, which is due to the limited data set. We have previously shown that the reconstruction can be made quantitative with more SD pairs and through use of nonlinear optimization methods [57]. The localization approach we present is thus a computationally efficient way of obtaining the position of an inhomogeneity, and anecdotally success appears possible with very limited measurement data.

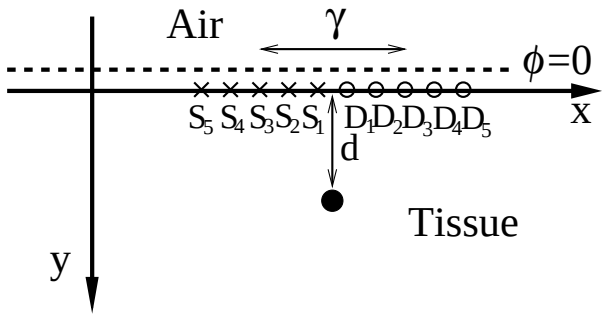


Fig. 3.1. Measurement geometry for localization. A spherical absorber at depth d is assumed in the simulation. The background optical parameters are: $\mu_{a0} = 0.02$ cm^{-1}, $D_0 = 0.03$ cm, and the modulation frequency is $\omega = 2\pi \times 10^6$ rad/s.

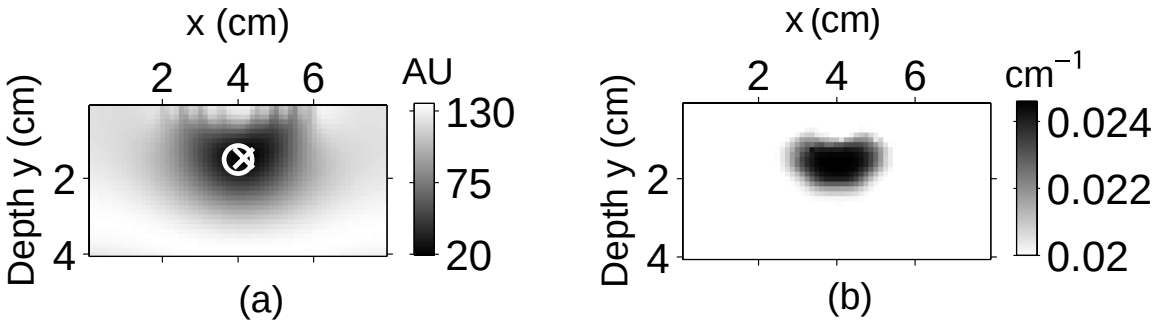


Fig. 3.2. Localization versus reconstruction: (a) Negative log likelihood: ∘ denotes the true inhomogeneity location and × the estimated location. (b) Optical diffusion tomography reconstruction of μ_a. Parameters: 5 sources and 5 detectors and background parameters as in Fig. 3.1; inhomogeneity $\mu_a = 0.12$ cm^{-1}, $D = 0.03$ cm; average SNR is 40 dB; spherical inhomogeneity diameter of 0.625 cm.

3.3 Detection

Determination of the inhomogeneity's presence, or lack thereof, is a detection problem for which we employ binary hypothesis testing. Let the hypothesis H_0 correspond to the absence of an inhomogeneity and $H_{1,\mathbf{r}}$ to the presence of an inhomogeneity at position $\mathbf{r}$. The probability densities for $\mathbf{y}$ under the two hypotheses are

$$p(\mathbf{y}|H_{1,\mathbf{r}}) = \frac{|\Lambda|}{(2\pi)^M} \exp\left(-\frac{1}{2} \parallel \mathbf{y} - \mathbf{f}(\mathbf{r}) \parallel_\Lambda^2\right) \tag{3.5}$$

$$p(\mathbf{y}|H_0) = \frac{|\Lambda|}{(2\pi)^M} \exp\left(-\frac{1}{2} \parallel \mathbf{y} - \mathbf{y}_0 \parallel_\Lambda^2\right). \tag{3.6}$$

The likelihood ratio test (LRT) is

$$L(\mathbf{y}, \mathbf{r}) = \log \frac{p(\mathbf{y}|H_{1,\mathbf{r}})}{p(\mathbf{y}|H_0)} = \mathrm{Re}[\mathbf{h}(\mathbf{r})^H(\mathbf{y} - \mathbf{y}_0)] - c(\mathbf{r}) \ , \tag{3.7}$$

where $\mathbf{h}(\mathbf{r})^H = \Delta\mathbf{y}(\mathbf{r})^H\Lambda$ can be viewed as a matching filter, $c(\mathbf{r}) = (1/2) \parallel \Delta\mathbf{y}(\mathbf{r}) \parallel_\Lambda^2$ is a constant for each position $\mathbf{r}$, and $\Delta\mathbf{y}(\mathbf{r}) = \mathbf{f}(\mathbf{r}) - \mathbf{y}_0$. Equation (3.7) provides the highest probability of detection for a specified false alarm rate. The LRT suggests that if the correlation between $\mathbf{y} - \mathbf{y}_0$ and $\mathbf{h}(\mathbf{r})$ is above a certain threshold, then we say an inhomogeneity exists. The decision statistic $q = \mathrm{Re}[\mathbf{h}(\mathbf{r})^H(\mathbf{y} - \mathbf{y}_0)]$ has a normal distribution under the two hypotheses, i.e., $(q \mid H_0) \sim N(0, \sigma_q^2)$ and $(q \mid H_{1,\mathbf{r}}) \sim N(\bar{q}, \sigma_q^2)$, where both the mean $\bar{q}$ and variance σ_q^2 are equal to $\parallel \Delta\mathbf{y}(\mathbf{r}) \parallel_\Lambda^2$. For a specified false alarm rate P_F, the threshold k_{P_F} can be determined as $k_{P_F} = \sigma_q\Phi^{-1}(1 - P_F)$, where Φ is a normal distribution function with mean 0 and variance 1. Thus we declare that an inhomogeneity exists if $q > k_{P_F}$. For a specific measurement system, the probability of detection is

$$P_D = \int_{k_{P_F}}^{\infty} p(q|H_{1,\mathbf{r}})dq = 1 - \Phi(\frac{k_{P_F} - \bar{q}}{\sigma_q}). \tag{3.8}$$

Consider now the influence of physical (size, depth) and optical (contrast) properties of the inhomogeneity on P_D, assuming that these properties are known. In practice, the parameters describing the inhomogeneity are unknown and must be estimated. Therefore, the results of our simulation, with P_D computed using (3.8),

gives an upper bound for the P_D of a measurement system. The measurement geometry of Fig. 3.1 is used. Fig. 3.3(a) plots P_D as a function of the inhomogeneity depth for the case of Fig. 3.2. P_D decreases as the inhomogeneity depth increases, and the reliable detection depth is about 2 cm . Fig. 3.3(b) gives P_D as a function of inhomogeneity size and contrast for a fixed depth of 1.5 cm. Notice that detection becomes more reliable as both the size and contrast increase, with a fixed source power (SNR). Fig. 3.3(c) shows P_D as a function of inhomogeneity depth and contrast. The achievable detection depth increases with the contrast but finally saturates at about 3 cm. This saturation is dictated by the detector noise, i.e., by the SNR. Fig. 3.3(d) gives P_D as a function of inhomogeneity depth and size. The achievable detection depth increases with the inhomogeneity size, but saturates also due to the noise floor.

3.4 Source-Detector Geometry Design

The placement and number of sources and detectors amounts to instrument design. Our strategy is to maximize the detection sensitivity $S = |y_i - y_{0i}|^2/y_{0i}$ for each SD pair, where y_i is the element of $\mathbf{y}$ with $\mathrm{S}_i - \mathrm{D}_i$ and the inhomogeneity present, and y_{0i} that without the inhomogeneity. An increase in S corresponds to an increase in P_D, as (3.8) indicates. The analytical result for the sensitivity is plotted in Fig. 3.4 for inhomogeneity depths of 2 cm and 3 cm. The optimal SD separations are about 2.3 cm and 3.5 cm, respectively. The optimal SD separation increases as the inhomogeneity depth increases, ultimately being limited by the detector noise floor. By obtaining such information, one can optimize the design of a measurement system. A convenient approximation for the closed form semi-infinite medium solution for S can be found under the assumption that $d \gg l^*$ and $\gamma \gg l^*$, where $l^* = 3D$ is the mean free path and γ is the distance between the SD pair, which we find to be

$$S \approx A \cdot \frac{\gamma^2 \exp\{-4k(\frac{\gamma^2}{4} + d^2)^{1/2}\}}{(\frac{\gamma^2}{4} + d^2)^4 \exp\{-k\gamma\}} \ , \tag{3.9}$$

where A is a constant and $k = [(c^2\mu_a^2 + \omega^2)/(c^2D^2)]^{1/4} \cos[(1/2)\tan^{-1}(\omega/(c\mu_a))]$ is the decay coefficient. The scaled result from (3.9) is shown as points in Fig. 3.4, and

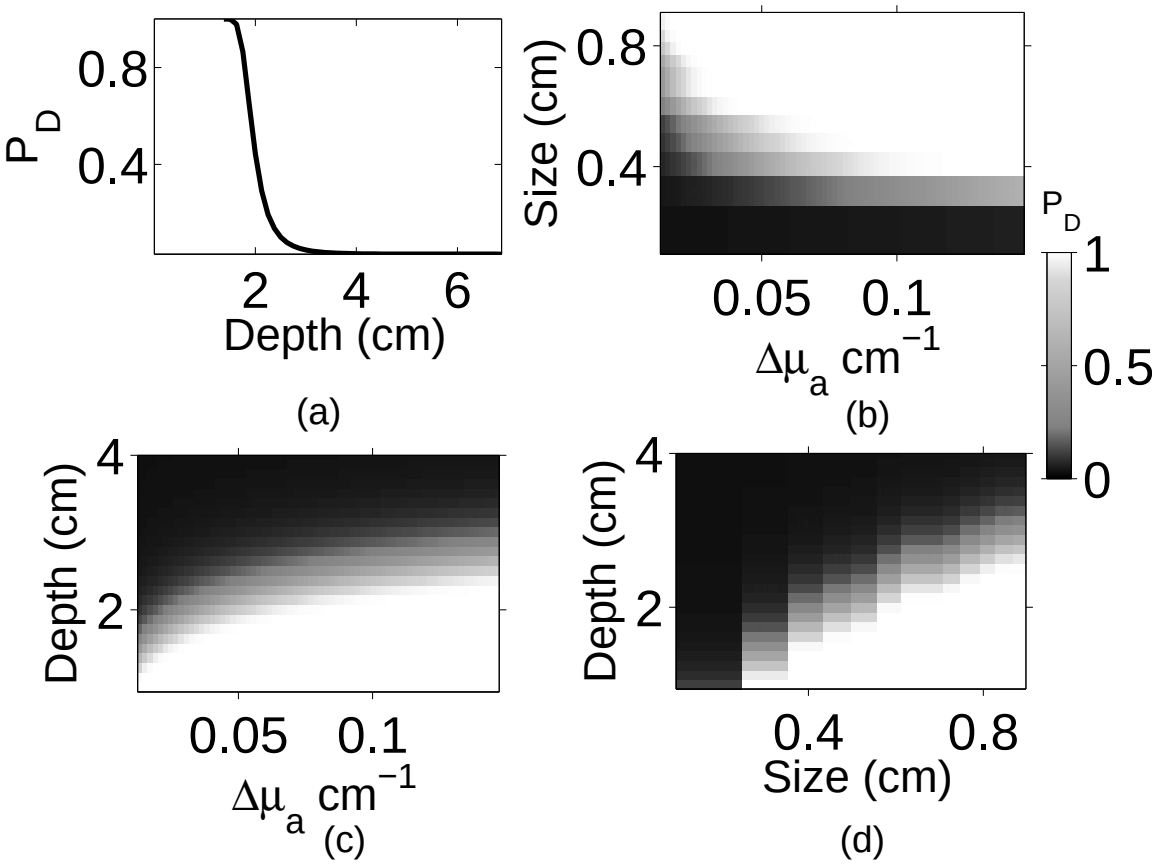


Fig. 3.3. Influence of inhomogeneity depth, size and optical contrast ($\Delta\mu_a$) on P_D for the geometry and parameters shown in Fig. 3.1, with $P_F = 0.03$ and an average SNR of 40 dB. (a) P_D as a function of depth, for an inhomogeneity having: diameter 0.625 cm, $\mu_a = 0.12$ cm^{-1}, and $D = 0.03$ cm. (b) P_D as a function of size and $\Delta\mu_a$, with $d = 1.5$ cm. (c) P_D as a function of depth and $\Delta\mu_a$, with a 0.625 cm diameter inhomogeneity. (d) P_D as a function of depth and size, with $\Delta\mu_a = 0.1$ cm^{-1}.

these agree nicely with the analytical result. An instrument should have SD spacings that encompass the optimum sensitivity, which for the two cases we consider are given by the peaks in Fig. 3.4.

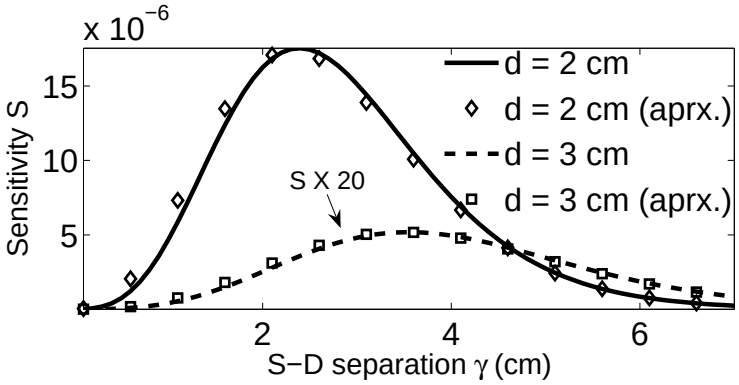


Fig. 3.4. Detection sensitivity as a function of S-D distance for two inhomogeneity depths. The background optical parameters are: $\mu_{a0} = 0.1$ cm^{-1}, $D_0 = 0.03$ cm, which give $k = 0.9$ cm^{-1}. The sensitivity for inhomogeneity depth 3 cm is magnified 20 times. The points are the approximate solution from (3.9).

3.5 Conclusions

In this chapter, we presented a statistical framework for fast localization and detection of an absorbing inhomogeneity in a scattering medium as an alternative to volumetric reconstruction. Extension to a diffusion coefficient inhomogeneity would follow the same general procedure. With a known inhomogeneous background, the forward model could be calculated numerically, and the Fréchet matrix elements stored beforehand. The same approach can be used to localize and detect the sparse hemodynamic response in the rapidly developing field of functional optical imaging of brain activites [58, 59] .

4. COVARIANCE ESTIMATION FOR HIGH DIMENSIONAL DATA VECTORS USING THE SPARSE MATRIX TRANSFORM

4.1 Introduction

Many problems in statistical pattern recognition and analysis require the classification and analysis of high dimensional data vectors. However, covariance estimation for high dimensional vectors is a classically difficult problem because the number of coefficients in the covariance grows as the dimension squared [60, 61]. In a typical application, one measures n samples of a p dimensional vector. If $n < p$, then the sample covariance matrix will be singular with $p-n$ eigenvalues equal to zero. This is usually referred to as "small n, large p" inference problem. This problem, sometimes also referred to as the curse of dimensionality [62, 63], presents a classic dilemma in statistical pattern analysis and machine learning.

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 identity matrix or the diagonal of the sample covariance [64–67].

More recently, a number of methods have been proposed for regularizing the estimate by making either the covariance or its inverse sparse [68, 69]. For example, the graphical lasso method enforces sparsity by imposing an L_1 norm constraint on the inverse covariance [69]. Banding or thresholding have also been used to obtain a sparse estimate of the covariance [68, 70]. Some other methods apply L_1 sparsity con-

straints to the eigenvector transform (i.e. PCA transform) itself, and are collectively referred to as sparse PCA [71–74].

In this chapter, we propose a new approach to covariance estimation, which is based on constrained maximum likelihood (ML) estimation of the covariance [27, 75]. In particular, the covariance is constrained to have an eigen decomposition which can be represented as a sparse matrix transform (SMT) [17, 76]. The SMT is formed by a product of pairwise coordinate rotations known as Givens rotations [18]. Using this framework, the covariance can be efficiently estimated using greedy minimization 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.

In order to validate our model, we perform experiments using standard hyperspectral data and face image sets. We compare against both traditional shrinkage estimates and recently proposed graphical lasso estimates. Our experiments show that, for these examples, the SMT covariance estimate is consistently more accurate for a variety of different classes and sample sizes. In addition, the SMT method has a number of other advantages. The SMT estimate yields a sparse transformation for the implementation of eigen-transform. Therefore, the resulting eigen transform can be computed with very little computation (i.e. $\ll p^2$ operations). Moreover, it seems to be particularly good when estimating small eigenvalues and their associated eigenvectors. Also, the cross-validation procedure used to estimate the SMT model order requires little additional computation, and thus can be done very efficiently.

4.2 Covariance Estimation for High Dimensional Vectors

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

$$Y = [y_1, y_2, \cdots, y_n] \ . \tag{4.1}$$

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

$$S = \frac{1}{n} Y Y^t \ , \tag{4.2}$$

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.

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 [64–66]. 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) [65, 66] and the diagonal entries of S, $\mathrm{diag}(S)$ [65, 67]. 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 [68,69]. For example, the graphical lasso method enforces sparsity by imposing an L_1 norm constraint on the inverse covariance [69], and is therefore a good representative of the general class of L_1 based methods. Banding or thresholding can also be used to obtain a sparse estimate of the covariance [68, 70].

4.2.1 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) [17, 76]. To do this, we first decompose R as

$$R = E\Lambda E^t \ , \tag{4.3}$$

where E is the orthonormal matrix of eigenvectors 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. 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

$$P_R(Y) = \frac{1}{(2\pi)^{\frac{np}{2}}} |R|^{-\frac{n}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\{Y^t R^{-1} Y\}\right\} \ . \tag{4.4}$$

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

$$\log P_{(E,\Lambda)}(Y) = -\frac{n}{2}\mathrm{tr}\{\mathrm{diag}(E^t S E)\Lambda^{-1}\} - \frac{n}{2}\log|\Lambda| - \frac{np}{2}\log(2\pi) \ , \tag{4.5}$$

where $R = E\Lambda E^t$ is specified by the orthonormal eigenvector matrix E and diagonal eigenvalue matrix Λ. Jointly maximizing the likelihood with respect to E and Λ then results in the ML estimates of E and Λ given by (see Appendix F)

$$\hat{E} = \arg\min_{E\in\Omega} \left\{\left|\mathrm{diag}(E^t S E)\right|\right\} \tag{4.6}$$

$$\hat{\Lambda} = \mathrm{diag}(\hat{E}^t S \hat{E}) \ , \tag{4.7}$$

where Ω is the set of allowed orthonormal transforms, and $|\cdot|$ represents 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 (4.6), and then computing the eigenvalue estimates from (4.7).

An interesting special case occurs when S has full rank and Ω is the set of all orthonormal transforms. In this case, equations (4.6) and (4.7) are solved by selecting E and Λ as the eigenvector matrix and eigenvalue matrix of S, respectively (see Appendix G). 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.

4.2.2 ML Estimation of Eigenvectors Using SMT Model

The ML estimate of E can be improved if the feasible set of eigenvector transforms, 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 the 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 [17, 76]. 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

$$E = \prod_{k=1}^{K} E_k = E_1 E_2 \cdots E_K \ , \tag{4.8}$$

where every sparse matrix, E_k, is a Givens rotation operating on a pair of coordinate indices (i_k, j_k) [18]. Every Givens rotation E_k is an orthonormal rotation in the plane of the two coordinates, i_k and j_k, which has the form

$$E_k = I + \Theta(i_k, j_k, \theta_k) \ , \tag{4.9}$$

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

$$[\Theta]_{ij} = \begin{cases} \cos(\theta_k) - 1 & \text{if } i = j = i_k \text{ or } i = j = j_k \\ \sin(\theta_k) & \text{if } i = i_k \text{ and } j = j_k \\ -\sin(\theta_k) & \text{if } i = j_k \text{ and } j = i_k \\ 0 & \text{otherwise} \end{cases} . \tag{4.10}$$

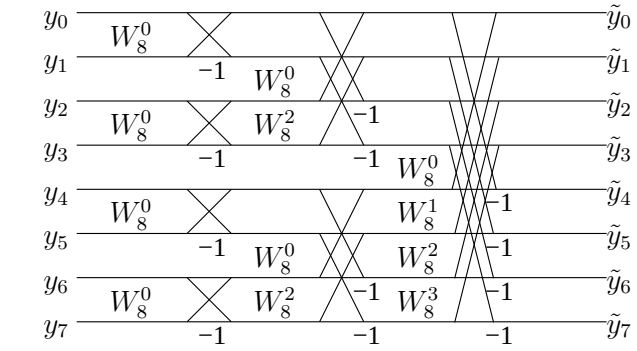
Figure 4.1(b) shows the flow diagram for the application of an SMT to a data vector y. Notice that each 2D rotation, E_k, plays a role analogous to a "butterfly" used in a traditional fast Fourier transform (FFT) [36] in Fig. 4.1(a). However, unlike an FFT, the organization of the butterflies in an SMT is unstructured, and 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 can be adjusted for the specific characteristics of the data, therefore, the SMT is a transform which is adaptive to 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 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 [25, 77]. The SMT also includes the recently proposed class of treelets [26], which uses less than p 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 H). Therefore, by varying the number of Givens rotations K, we can increase or decrease the set of orthonormal transforms that the SMT can represent.

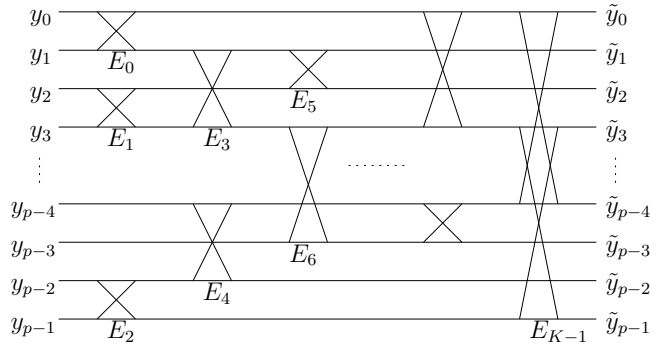
Using the SMT model constraint, the ML estimate of E is given by

$$\hat{E} = \arg \min_{E = \prod_{k=1}^{K} E_k} \left| \text{diag}(E^t S E) \right| . \tag{4.11}$$

Unfortunately, evaluating the constrained ML estimate of (4.11) requires the solution of an optimization problem with a non-convex constraint. So evaluation of the



(a) FFT



(b) SMT

Fig. 4.1. (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.

globally optimal solutions is difficult. Therefore, our approach will be to use greedy minimization to compute a locally optimal solution to (4.11). 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.

$$E_k^* = \arg\min_{E_k} \left|\text{diag}\left(E_k^t S_k E_k\right)\right| \tag{4.12}$$

$$S_{k+1} = E_k^{*t} S_k E_k^* \ . \tag{4.13}$$

The resulting values of E_k^* are the butterflies of the SMT.

The problem remains of how to compute the solution to (4.12). In fact, this can be done quite easily by first determining the two coordinates, i_k and j_k, that are most correlated,

$$(i_k, j_k) \leftarrow \arg\min_{(i,j)} \left(1 - \frac{[S_k]_{ij}^2}{[S_k]_{ii}[S_k]_{jj}}\right) \ . \tag{4.14}$$

It can be shown that this coordinate pair, (i_k, j_k), can most reduce the cost in (4.12) among all possible coordinate pairs (see Appendix I). Once i_k and j_k are determined, we apply the Givens rotation E_k^* to minimize the cost in (4.12), which is given by

$$E_k^* = I + \Theta(i_k, j_k, \theta_k) \ , \tag{4.15}$$

where [1]

$$\theta_k = \frac{1}{2}\text{atan}(-2[S_k]_{i_k j_k}, [S_k]_{i_k i_k} - [S_k]_{j_k j_k}) \ . \tag{4.16}$$

By iterating Eq (4.12) and (4.13) K times, we obtain the constrained ML estimate of E given by

$$\hat{E} = \prod_{k=1}^{K} E_k^* \ . \tag{4.17}$$

[1]Here we use $\text{atan}(y, x) = \text{atan}(y/x)$ 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.

4.2.3 Model Order

The model order, K, can be determined efficiently using a simple cross validation procedure. We can partition the data into t subsets, $\{Y_1, Y_2, \ldots, Y_t\}$, and K is chosen to maximize the average log-likelihood of the left-out subset given the estimated covariance using the other $t-1$ subsets. For example, let $(\hat{E}_{1,k}, \hat{\Lambda}_{1,k})$ denote the order-k SMT covariance estimate based on the $n-1$ subsets except Y_1. The log-likelihood of Y_1 given $(\hat{E}_{1,k}, \hat{\Lambda}_{1,k})$ can be computed as follows

$$\log P_{(\hat{E}_{1,k},\hat{\Lambda}_{1,k})}(Y_1) = -\frac{1}{2}\mathrm{tr}\left\{\mathrm{diag}(\hat{E}_{1,k}^t S_1 \hat{E}_{1,k})\hat{\Lambda}_{1,k}^{-1}\right\} - \frac{1}{2}\log\left|\hat{\Lambda}_{1,k}\right| - \frac{p}{2}\log(2\pi)\ . \quad (4.18)$$

Notice that the log-likelihood in (4.18) can be recursively evaluated on-the-fly for different k starting from $k=1$. Since each $\hat{E}_{1,k}$ is a sparse operator, this recursive evaluation requires little additional computation.

The log-likelihood in Eq (4.18) is evaluated until each subset is used as left-out subset once. The model order K is then chosen to maximize the average log-likelihood over all the subsets,

$$K^* = \arg\max_k \frac{1}{t}\sum_{i=1}^{t} \log P_{(\hat{E}_{i,k},\hat{\Lambda}_{i,k})}(Y_i)\ .$$

Once K^* is determined, the SMT covariance estimate is re-computed using all the data and the estimated model order.

4.2.4 Numerical Stable SMT Covariance Estimator

If there are two perfectly or almost perfectly correlated components, then the variance of the small component will be zero in numerical precision after the pair-wise decorrelation. This can happen to some synthesized data and cause potential numerical instability in the following criteria selection step according to (4.14). To avoid it, we can adjust the sample covariance as $\tilde{S} = S + \delta I$ with a small positive

$\delta > 0$. Then, the constrained ML estimation of the eigenvectors and eigenvalues of the covariance can be written as

$$\left(\hat{E}, \hat{\Lambda}\right) = \arg \max_{(E \in \Omega, \Lambda)} \left\{ -\frac{n}{2} \mathrm{tr}\{\mathrm{diag}(E^t \tilde{S} E)\Lambda^{-1}\} - \frac{n}{2} \log|\Lambda| - \frac{np}{2} \log(2\pi) \right\} . \quad (4.19)$$

This is exactly the same optimization problem as (4.5) and thus has the same solutions as (4.6) and (4.7) with S replaced by $\tilde{S}$. To obtain $\hat{E}$ in (4.19), it is equivalent to adjust the selection criteria as follows

$$(i_k, j_k) \leftarrow \arg \min_{(i,j)} \left(1 - \frac{[S_k]_{ij}^2}{([S_k]_{ii} + \delta) \cdot ([S_k]_{jj} + \delta)} \right) . \quad (4.20)$$

Notice that all the resulting SMT estimate of eigenvalues will be larger than δ. The value of δ can be decided according to the noise level. Therefore, by this way we regularize both the eigenvalues and eigenvectors. For the numerical experiments in Section 4.5, however, $\delta = 0$ is always used.

4.3 Algorithm Implementation and Complexity Analysis

An explicit implementation of the greedy algorithm of the SMT covariance estimation has a computation complexity of $O(np^2 + Kp^2)$, where the $O(np^2)$ is required for initial construction of the sample covariance S, and $O(p^2)$ is required for exhaustively searching the minimum of the selection criteria in Eq (4.14) at K steps. It also requires a memory of $O(p^2)$ for storing S. However, the greedy algorithm can be implemented much more efficiently. After initial search at step $k = 1$, we can store the minimal criteria value and its corresponding column index for each row in two column vectors. Notice that every E_k only modifies two rows and columns of S at each k-th step. Therefore, at the end of each iteration, these two vectors can be updated with another $O(p)$ operations, and the minimum of the selection criteria can be also found in $O(p)$ operations for each $k > 1$. Therefore, the computation complexity of $O(np^2 + Kp)$ can be achieved. Moreover, if p is very large and $p \gg n$, it is actually not necessary to construct the large sample covariance matrix S. Instead, we can directly apply the greedy algorithm on the data Y, and hence dramatically

reduce the memory requirement. This can be important for some applications with very high data dimensions, e.g. image analysis.

Next, we provide the implementation of the SMT covariance estimation in detail. Specifically, two different cases are discussed: i) $n > p$ or $n \approx p$; and ii) $n \ll p$. Table 4.1 shows the computation and memory complexity of the corresponding algorithms.

4.3.1 Case I: $n > p$ or $n \approx p$

Here, p is not so large, and it is not a problem to compute and store the sample covariance matrix S. For this case, the greedy algorithm for the SMT covariance estimation can be explicitly implemented as desribed in Section 4.2.2 with the revised search strategy desribed above. The algorithm implementation is illustrated in Fig. 4.2(a). It has a computation complexity of $O(np^2 + pK)$ with $O(np^2)$ required for the construction of the sample covariance matrix and $O(pK)$ for the greedy optimization after the initialization. It has a memory requirement of $O(p^2)$ for storing the sample covariance matrix.

4.3.2 Case II: $n \ll p$

Here, the data dimension p may be very large, and the sample number n can be much smaller than p. Therefore, the memory required for storing the sample covariance matrix, $O(p^2)$, can be prohibitively high. In this case, we develop an algorithm implementation that directly operates on the data Y instead of the sample covariance S, which is illustrated in Fig. 4.2(b). The idea is to compute the required correlation coefficients of Y on-the-fly. This algorithm has a computation complexity of $O(np^2 + npK)$ with $O(np^2)$ required for initial search of the criteria minimum and $O(npK)$ for the greedy optimization after the initialization. It has a memory requirement of $O(np)$ for storing the data matrix Y. Compared to the first case, this

Table 4.1
Complexity of greedy algorithm I and II for the SMT covariance estimation. 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.

Method	n versus p	Comp. Compl.	Mem. Compl.
Algo. I	$n > p$ or $n \approx p$	$O(np^2 + pK)$	$O(p^2)$
Algo. II	$n \ll p$	$O(np^2 + npK)$	$O(np)$

implementation has a trade-off between the computation and memory requirement, which is justified by the condition $n \ll p$.

4.4 Properties of SMT Covariance Estimator and Its Extensions

4.4.1 Properties of SMT Covariance Estimator

Throughout this chapter, we assume the data sample $Y = [y_1, y_2, \cdots, y_n]$ is a p-dimensional second order i.i.d. process with zero mean and positive definite covariance R. Let $\hat{R} = \hat{E}\hat{\Lambda}\hat{E}^t$ be the SMT estimator as described in the previous section. The SMT covariance estimator has some interesting properties. First, $\hat{R}$ is clearly symmetric. Also, it has the following properties.

Property 4.4.1 *If $\hat{R}$ is the unique order-K SMT covariance estimate of 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 Eq (4.14) has a unique minimum at each step $k \leq K$. This is true for a physical process with a probability of 1. The proof of Property 2 is given in Appendix J. This property shows that the SMT covariance estimator is permutation invariant. In other words, the SMT estimate does not depend on the ordering of the data. Therefore, the SMT can potentially be used to process the data

$S \leftarrow YY^t/n$
$\Lambda \leftarrow \mathrm{diag}\,(S)$
$R \leftarrow \Lambda^{-1/2} S \Lambda^{-1/2}$
$[MaxJ(i), MaxR(i)] \leftarrow \left[\arg\max_{j<i} \{|\, R(i,j)\,|\}, \max_{j<i} \{|\, R(i,j)\,|\}\right]$, for $i = 1:p$
For $k = 1:K$ {

$$
\begin{aligned}
i_k &\leftarrow \arg\max_i (MaxR) \\
j_k &\leftarrow MaxJ(i_k) \\
\theta_k &\leftarrow \frac{1}{2}\mathrm{atan2}(-2S_{i_k,j_k}, S_{i_k i_k} - S_{j_k j_k}) \\
E_k &\leftarrow I + \Theta(i_k, j_k, \theta_k) \\
S &\leftarrow E_k^t S E_k \\
R(i,:) &\leftarrow S(i,:)/\sqrt{\mathrm{diag}(S)\cdot S(i,i)},\ \text{if}\ (i == i_k \text{ or } j_k) \\
R(:,i) &\leftarrow R(i,:)^t,\ \text{if}\ (i == i_k \text{ or } j_k) \\
\text{if}\ (i == i_k \text{ or } j_k) \quad &\text{or} \quad (MaxJ(i) == i_k \text{ or } j_k)\ , \\
[MaxJ(i), MaxR(i)] &\leftarrow \left[\arg\max_j \{|\, R(i,j)\,|\}, \max_j \{|\, R(i,j)\,|\}\right]
\end{aligned}
$$

}
$\Lambda \leftarrow \mathrm{diag}\,(S)$
$E \leftarrow \prod_{k=1}^K E_k$

(a)

$[MaxJ(i), MaxR(i)] \leftarrow \left[\arg\max_{j<i} \left\{\frac{|Y(i,:)^t Y(j,:)|}{\|Y(i,:)\|\|Y(j,:)\|}\right\}, \max_{j<i} \left\{\frac{|Y(i,:)^t Y(j,:)|}{\|Y(i,:)\|\|Y(j,:)\|}\right\}\right]$, for $i = 1:p$
For $k = 1:K$ {

$$
\begin{aligned}
i_k &\leftarrow \arg\max_i (MaxR) \\
j_k &\leftarrow MaxJ(i_k) \\
\theta_k &\leftarrow \frac{1}{2}\mathrm{atan2}(-2S_{i_k,j_k}, S_{i_k i_k} - S_{j_k j_k}) \\
E_k &\leftarrow I + \Theta(i_k, j_k, \theta_k) \\
Y &\leftarrow E_k^t Y \\
\text{if}\ (i == i_k \text{ or } j_k) \quad &\text{or} \quad (MaxJ(i) == i_k \text{ or } j_k)\ , \\
[MaxJ(i), MaxR(i)] &\leftarrow \left[\arg\max_j \left\{\frac{|\, Y(i,:)^t Y(j,:)\,|}{\|\, Y(i,:)\,\|\|\, Y(j,:)\,\|}\right\}, \max_j \left\{\frac{|\, Y(i,:)^t Y(j,:)\,|}{\|\, Y(i,:)\,\|\|\, Y(j,:)\,\|}\right\}\right]
\end{aligned}
$$

}
$\Lambda_{ii} \leftarrow Y(i,:)^t Y(i,:)/n$, for $i = 1:p$
$E \leftarrow \prod_{k=1}^K E_k$

(b)

Fig. 4.2. Pseudo-code of the greedy algorithms for the SMT covariance estimation. Notations of the operators follows the style of Matlab. (a) Algorithm I where $n > p$ or $n \approx p$. In this case, the sample covariance is explicitly constructed. (b) Algorithm II where $n \ll p$. In this case, the sample covariance is computed on-the-fly to save memory.

sets whose ordering does not have explicit meaning, such as text data, financial data, distributed sensor networks.

Property 4.4.2 *The computation of the eigen transformation $\hat{E}^t y$ is of $O(K)$.*

The eigen transform using the SMT can be efficiently computed by applying the K Givens rotations in sequence. Every Givens rotation requires at most 4 multiplies (actually only 2 multiplies [76]). Therefore, the eigen transformation $\hat{E}^t y$ has a complexity of $O(K)$. As we found in our experiments, usually K is a small multiple of p, which means the SMT eigen transform has a complexity of $O(p)$, the same order as the FFT and fast wavelet transform. The low computation complexity of the SMT makes it attractive for applications with very high data dimension p, such as eigen-image analysis.

Property 4.4.3 *The SMT covariance estimator $\hat{R}$ is generally positive definite, even for the limited sample size $1 < n < p$.*

The eigenvalues of the SMT covariance estimator are given by $\text{diag}\left(\tilde{Y}\tilde{Y}^t\right)$ where $\tilde{Y} = \hat{E}Y$. So the SMT covariance estimator is always non-negative definite. In fact, we know if $K = 0$ (i.e. $\hat{E} = I$), then $\hat{R} = \text{diag}\,(S)$, which is diagonal and positive definite. As K increases, which usually happens as the sample size n increases, the SMT estimator will approach the sample covariance. If K goes to infinity, then the SMT estimator eventually becomes the sample covariance, which makes the SMT equivalent to PCA. In general, the SMT provides a covariance estimator between the diagonal and the sample covariance depending on the sample size.

4.4.2 SMT Shrinkage Estimator

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

$$\hat{R}_s = \alpha\hat{R} + (1-\alpha)S \ , \tag{4.21}$$

where α is the shrinkage intensity.

The value of α can be efficiently computed using leave-one-out likelihood (LOOL) cross validation [65] in the SMT transformed domain. Let S_i be the sample covariance excluding y_i,

$$S_i = \frac{1}{n-1}\sum_{\substack{j=1 \\ j\neq i}}^{n} y_j y_j^t = \frac{n}{n-1}S - \frac{1}{n-1}y_i y_i^t \ , \tag{4.22}$$

and $\hat{R}_{s|i} = \alpha\hat{R} + (1-\alpha)S_i$ be the corresponding SMT-S covariance estimator. Notice that

$$P(y_i \mid \hat{R}_{s|i}) = P(\hat{E}y_i \mid \hat{E}\hat{R}_{s|i}\hat{E}^t) = P(\tilde{y}_i \mid \alpha\hat{\Lambda}_i + (1-\alpha)\tilde{S}_i) \ , \tag{4.23}$$

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

Let

$$G = \alpha\hat{\Lambda} + (1-\alpha)\frac{n}{n-1}\tilde{S} \ . \tag{4.24}$$

Then the log-likelihood of y_i given $\hat{R}_{s|i}$ in Eq(4.23) can be efficiently computed as

$$\log P\left(y_i \mid \hat{R}_{s|i}\right) = \log P\left(\tilde{y}_i \mid G - \beta\tilde{y}_i\tilde{y}_i^t\right) \tag{4.25}$$

$$= -\frac{1}{2}\log\{\mid G \mid (1-\beta d_i)\} - \frac{1}{2}\left(\frac{d_i}{1-\beta d_i}\right) - \frac{p}{2} \ , \tag{4.26}$$

where $\beta = \frac{1-\alpha}{n-1}$ and $d_i = \tilde{y}_i G^{-1}\tilde{y}_i^t$. This saves a large amount of computation since G^{-1} and $\mid G \mid$ now need to be computed only once for all y_i. The value of α that leads to maximum average LOOL is chosen as the final shrinkage intensity; and then the SMT-S covariance estimator $\hat{R}_s$ is re-computed using all the samples and the final shrinkage intensity.

4.5 Experimental Results

The effectiveness of the SMT covariance estimation depends on how well the SMT model can capture the behavior of real data vectors. Therefore in this section, we compare the performance of the SMT covariance estimators to commonly used shrinkage estimator and recently proposed graphical lasso estimator. We do this

comparison using standard hyperspectral remotely sensed data and face image sets as our high dimensional data vectors.

4.5.1 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 [65,67]. In this case, the shrinkage estimator is given by

$$\hat{R} = \alpha \text{diag}\,(S) + (1-\alpha)\,S\ . \tag{4.27}$$

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 [65].

An alternative estimator is the graphic lasso (glasso) estimate recently proposed in [69] which is an L_1-regularized maximum likelihood estimate, such that

$$\hat{R} = \arg\max_{R\in\Psi} \left\{ \log P(Y \mid R) - \rho \parallel R^{-1} \parallel_1 \right\}\ , \tag{4.28}$$

where Ψ denotes the set of $p \times p$ positive definite matrices and ρ the 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 R code for glasso that is publically available online [78]. ρ is chosen using the cross validation that maximizes the average log-likelihood of the left-out subset. Glasso 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 the optimization problem in Eq (4.28) for the glasso estimate to be solved for every different ρ value, which is is very expensive.

4.5.2 SMT Covariance Estimation for Hyperspectral Data Classification

The hyperspectral data we use is available with the recently published book [79]. Figure 4.3(a) shows a simulated color IR view of an airborne hyperspectral data flight-

line over the Washington DC Mall. The sensor system measured the pixel response in 191 effective bands 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. The data set also provides ground truth pixels for five classes designated as grass, water, street, roof, and tree. In Fig. 4.3(a), the ground-truth pixels of the grass class are outlined with a white rectangle. Figure 4.3(b) shows the spectrum of the grass pixels, and Fig. 4.3(c) shows multivariate Gaussian vectors that were generated using the measured sample covariance for the grass class.

For each class, we computed the "true" covariance by using all the ground truth pixels to calculate the sample covariance. The covariance is 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, street, roof, and tree are 1928, 1224, 3579, 416, and 388, respectively. In each case, the number of ground truth pixels was much larger than 191, so the true covariance matrices are nonsingular, and accurately represent the covariance of the hyperspectral data for that class. In each case, 3-fold cross validation is used to choose the regularization parameter for SMT and glasso, and LOOL cross validation is used to choose the shrinkage intensity for SMT-S and the shrinkage method. Figure 4.4 is an example of the plot of the average log-likelihood as a function of the number of Givens rotations K in cross validation.

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 the true covariance for each of the five classes. Next we estimated the covariance using the four methods, the shrinkage estimator, glasso, SMT and SMT shrinkage estimation. 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 true covariance using the Kullback-Leibler (KL) distance (Appendix K) [80]. The KL distance is a measure of the error between the estimated and true distribution. Figure 4.5(a), (b) and (c) show plots of the KL distances as a function of sample size for the four estimators. The error bars indicate the standard deviation of the KL distance due to random variation in the sample statistics. Notice that the SMT shrinkage (SMT-S) estimator is consistently the best of the four.

Figure 4.6(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. Notice that the SMT estimators generate good estimates especially for the small eigenvalues.

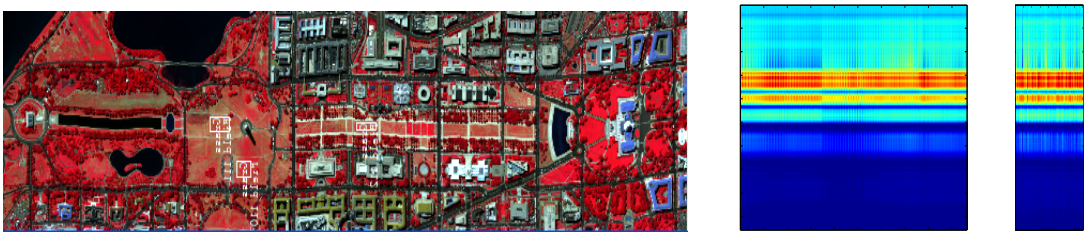
Table 4.2 compares the computational complexity, CPU time and model order for the four estimators with and without cross validation. The CPU time and model order were measured using the average results of 10 repeated experiments for the Gaussian case of the grass class with $n = 80$. Notice that even with the cross validation, the SMT and SMT-S estimators are much faster than glasso without cross-validation. This is due to the fact that the SMT transform is a sparse operator. In this example, the SMT uses an average of $K = 495$ rotations, which is equal to $K/p = 495/191 = 2.59$ rotations per spectral sample.

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 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[2] are shown in Fig. 4.5(d), (e) and (f); and Figure 4.6(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 shows that the SMT estimators are robust to non-Gaussian distributions.



(a) (b) (c)

Fig. 4.3. (a) Simulated color IR view of an airborne hyperspectral data over the Washington DC Mall [79]. (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.

4.5.3 SMT Covariance Estimation for Eigen Image Analysis

Eigen image analysis is an important problem in statistical image processing and pattern recognition. For example, eigenface is a well-known technique in face recognition and face image compression [82]. Eigen image analysis using the SMT estimator has several advantages, as demonstrated next.

Figure 4.7 shows how the SMT can be used to efficiently perform eigen-image analysis. First the SMT covariance estimation is used to estimate the covariance

[2]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.

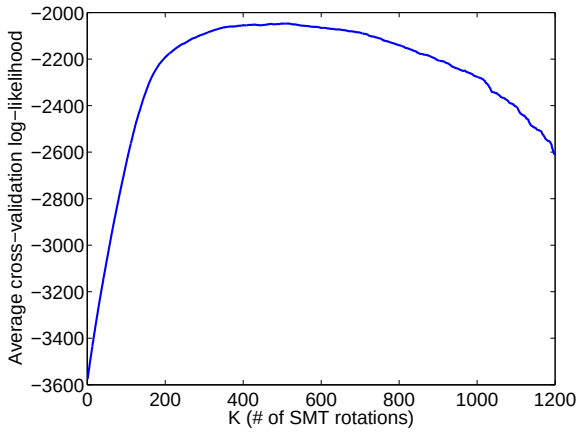
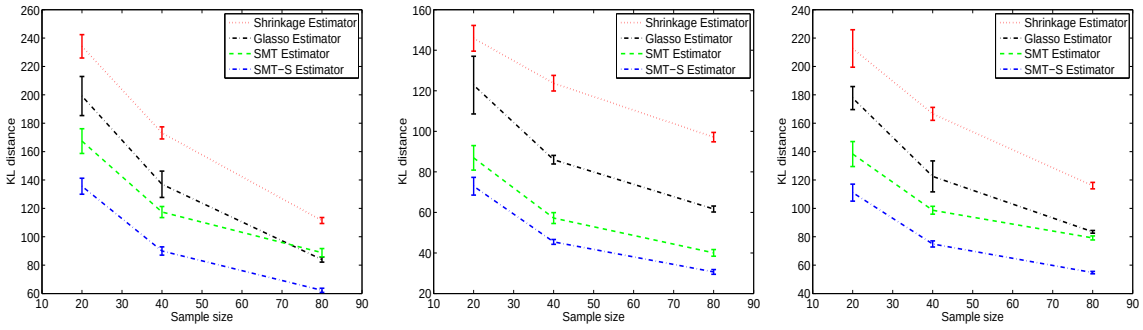


Fig. 4.4. Plot of the average log-likelihood as a function of the number of Givens rotations K in cross validation. The value of K that achieves the highest average log-likelihood is choosen as the number of rotations in the final SMT covariance estimator. $K = 495$ in this example.

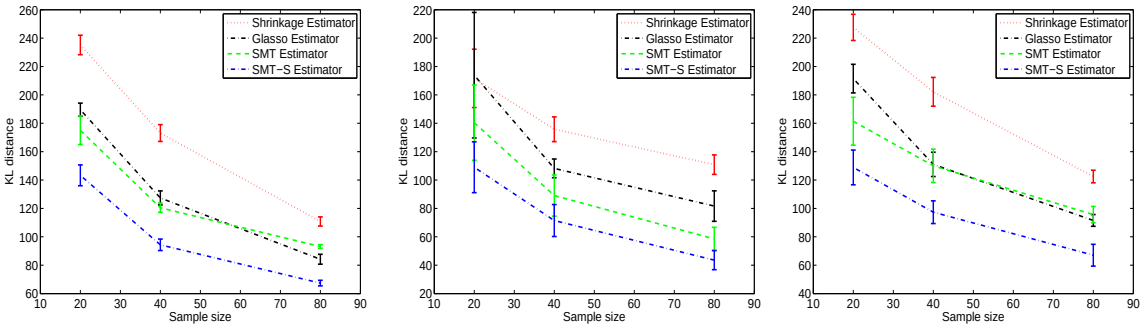
Table 4.2

Comparison of computational complexity, CPU time and model order for various covariance estimators with and without cross validation. The complexity does not include the computation of the sample covariance. Here, the CPU time and model order were measured as the average results for the Gaussian case of the grass class with $n = 80$. m – number of different cross validation values of the regularization parameter, t – number of splitted subsets in cross validation, and i – number of iterations used in glasso. c.v. stands for cross validation.

	Complexity		CPU time (sec.)		Parameter	Model order
	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	$\rho = 0.0005$	4939
SMT	$p^2 + Kp$	$t(p^2 + Kp)$	1.6	6.5	$K = 495$	495
SMT-S	$p^2 + Kp$	$m(p^3 + np^2)$	1.6	7.2	$(K, \alpha) = (495, 0.6)$	-

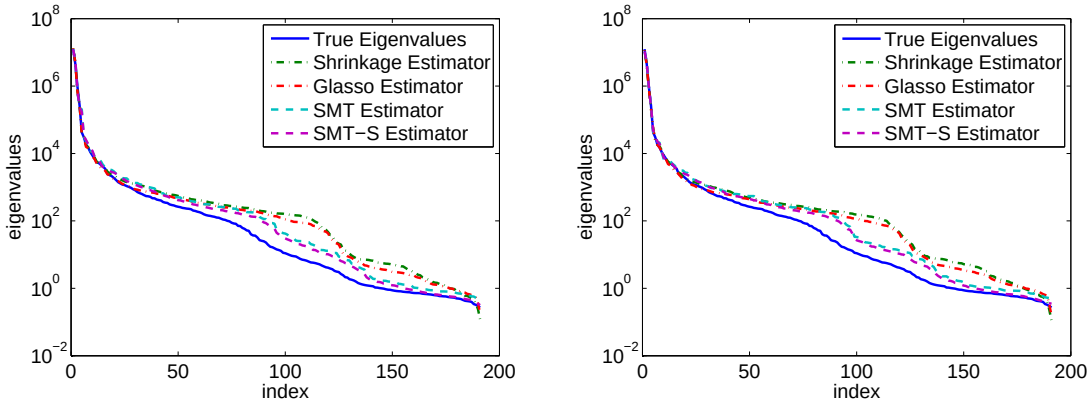


(a) Grass (b) Water (c) Street



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

Fig. 4.5. Kullback-Leibler distance from true distribution versus sample size for various classes: (a) (b) (c) Gaussian case (d) (e) (f) non-Gaussian case.



(a) (b)

Fig. 4.6. The distribution of estimated eigenvalues for the grass class with $n = 80$: (a) Gaussian case (b) Non-Gaussian case.

from n sample images, as in Fig. 4.7(a). In this case, every column of Y is a 2D face image. The SMT estimator produces a full set of eigenfaces even from the limited image samples. With the fast transform property of the SMT, one can either compute the eigen-image decomposition of a single image (see Fig. 4.7(b)), or using the adjoint transform, one can compute individual eigen images (see Fig. 4.7(c)). Notice it is not practical to store all the eigen images for the other methods. Since the SMT is a sparse operator, the SMT eigenface analysis could be readily scaled to mega-pixel images.

Here, we apply the SMT covariance estimation to a face image dataset from the ORL Face Database [83,84], with the images are 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. 4.8(a). First, we subtracted the sample mean from these images, and used these mean-subtracted images as our data sample for covariance estimation. We compared the SMT estimators with other covariance estimators in terms of both the accuracy and the visual quality. Especially, we included the covariance estimators using the diagonals of the sample covariance, i.e. $\hat{R} = \mathrm{diag}(S)$. This estimator models each pixel as an independent random variable, and has been widely used in image processing.

Eigenfaces

First we compute the eigenfaces of this face image set based on these different covariance estimators. 3-fold cross validation is used to choose the regularization parameter for the SMT and glasso, and LOOL cross validation is used to choose the shrinkage intensity for the SMT-S and the shrinkage method. Figure 4.8(b) to (f) show the first 80 estimated eigenfaces (i.e. columns of $\hat{E}$) using the different methods. Compared to the eigenfaces resulted from the diagonal and shrinkage estimators, the SMT eigenfaces clearly show much more visual structure. These spatial structure

could be better served as features in computer models. Also, the sparse feature of the SMT eigenfaces makes them very attractive in practical applications. Figure 4.9(a) shows the plot of average log-likelihood of the face images as a function of the number of Givens rotations K in 3-fold cross validation. The value of K that achieves the highest average log-likleihood is choosen as the model order in the final SMT covariance estimator, which is 974 in this example. Figure 4.9(b) shows the values of the regularization parameters for different estimators chosen by cross validation.

Automated Generation of Face Image Samples

In order to better illustrate the advantage of the SMT covariance estimates, we generated random face image samples using these different covariance estimates. Specifically, the face image samples were generated under the Gaussian distribution

$$y \sim N(\bar{y}, \hat{R}) \ , \tag{4.29}$$

where $\bar{y}$ is the sample mean of the training images, $\hat{R}$ is the covariance estimate. The generated sample images are shown in Fig. 4.10(a)–(e). Intuitively, the generated face samples using the SMT estimates have better visual quality as face images compared to the diagonal and shrinkage models, and are therefore good models for face images. The face image samples generated by glasso, the L_1 regularized sparse estimator, look good as well. This is because pixels far away from each other are commonly believed to have little correlation, and therefore can be reasonablly well modeld by sparse inverse covariance etimators such as glasso and Markov random field.

Cross Validated Log-Likelihood

Besides visual assessment of the different models, here we also provide some numerical evidence. 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. More specifically, we split the 80 face

images into 3 subsets (24,23,23). We estimated the covariance matrix using two of the subsets with an inner 3-fold cross validation used to chose the regularization parameter. Then the log-likelihood of the left-out subset is calculated based on the estimated covariance. This procedure was repeated until each subset was used as the left-out subset once, and then the average cross-validated log-likelihood was calculated. Figure 4.11(a) shows the average cross-validated log-likelihood of the face images using the SMT covariance estimators, as compared to the diagonal, shrinkage and glasso 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 highest likelihood. In Fig. 4.11(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. 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 very poor model. The SMT-S has a huge increase in likelihood ratio of $e^{167.3}$ compared to the shrinkage estimator. We believe that these enormous improvements in data fitting should result in an improved ability for object detection and recognition. We are currently investigating face recognition using SMT covariance estimation, and the results are encouraging.

4.6 Conclusion

In this chapter, we have proposed a novel method for covariance estimation of high dimensional data. The new method is based on constrained maximum likelihood (ML) estimation in which the eigenvector transformation is constrained to be the composition of K Givens rotations. This model seems to capture the essential behavior of the data 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

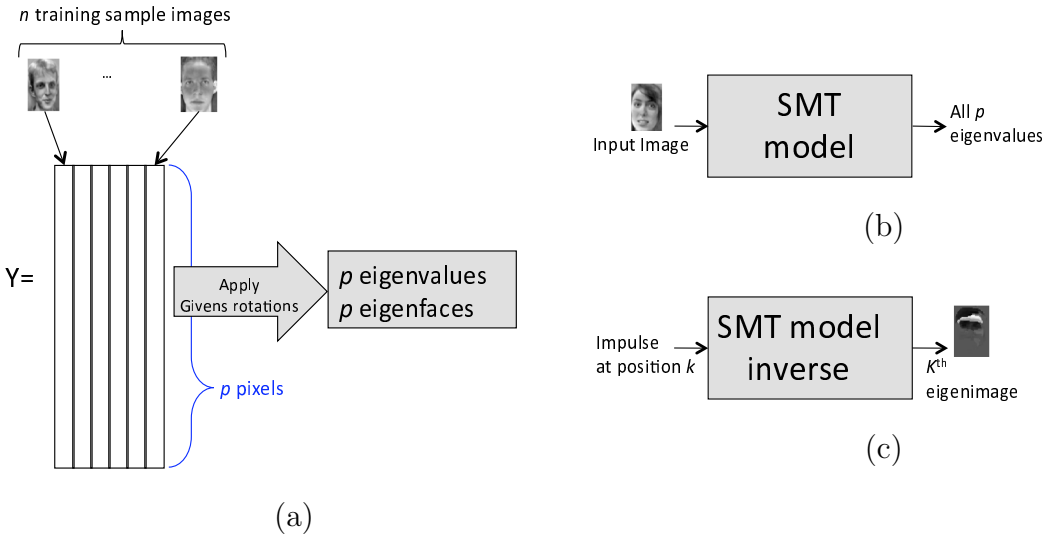


Fig. 4.7. This figure illustrates how the SMT covariance estimation can be used for eigen-image analysis. (a) A set of n images can be used to estimate the associated SMT. (b) The resulting SMT can be used to analyze a single input image, or (c) the transpose (i.e. inverse) of the SMT can be used to 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.

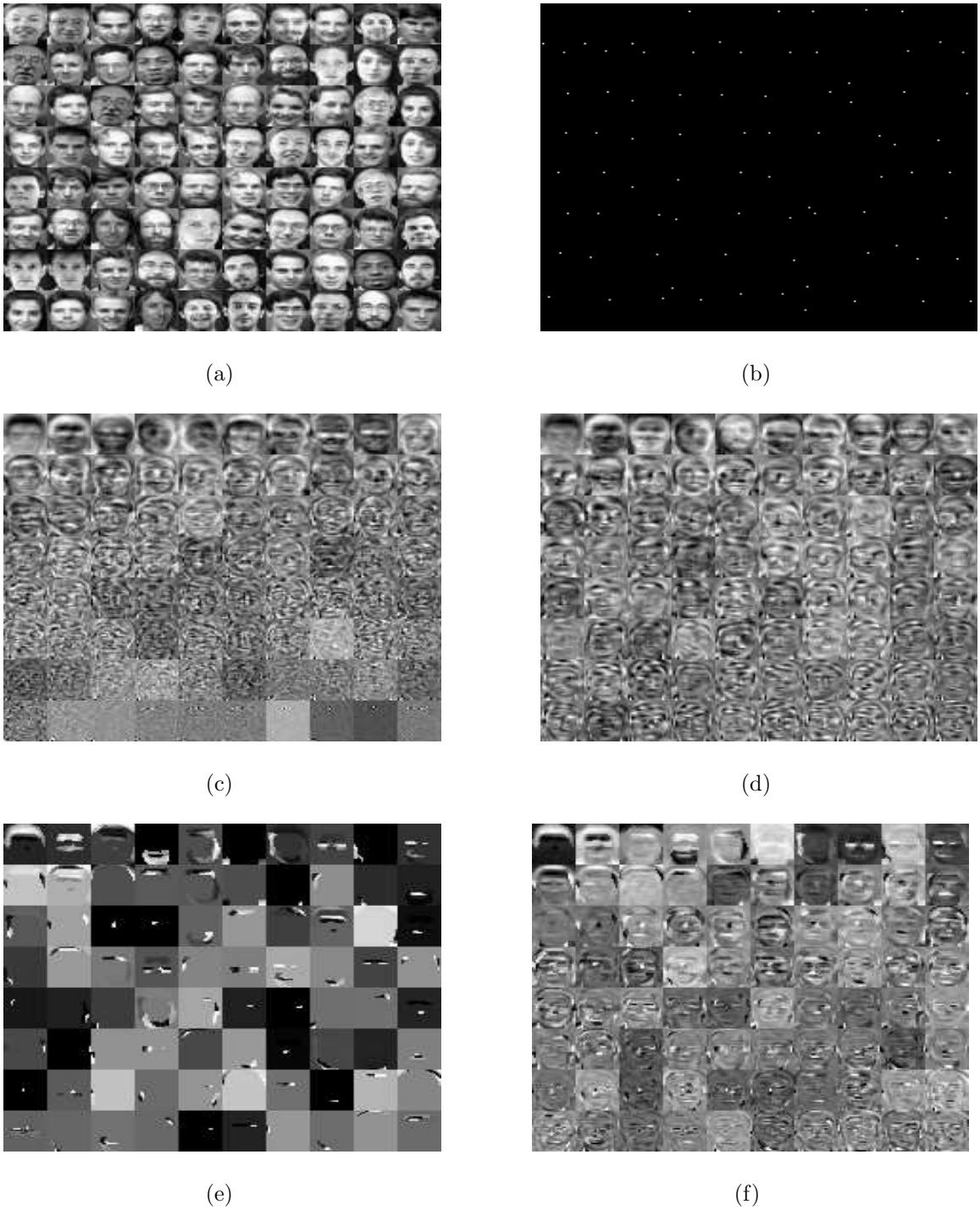
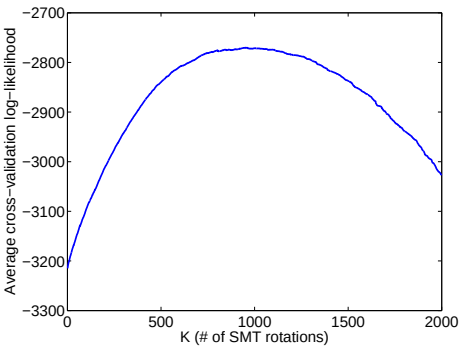


Fig. 4.8. Experimental results of eigen-image analysis for $n = 80$ and 28×23 thumbnail face images from the face image database [83]. (a) Example face image samples. First 80 eigen-images for each of the following methods: (b) Diagonal covariance estimate (i.e. independent pixels); (c) Shrinkage of sample covariance to diagonal; (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.

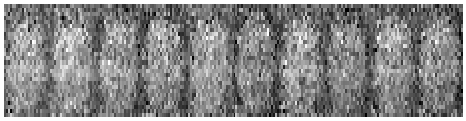


(a)

Method	Parameter
Diagonal	$K = 0$ or $\alpha = 1$
Shrinkage	$\alpha = 0.28$
glasso	$\rho = 0.08$
SMT	$K = 974$
SMT-S	$(K, \alpha) = (974, 0.8)$

(b)

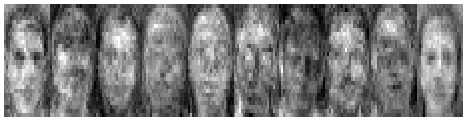
Fig. 4.9. Plot of the average log-likelihood as a function of the number of Givens rotations K in cross validation. The value of K that achieves the highest average log-likelihood is choosen as the number of rotations in the final SMT covariance estimator. $K = 974$ in this example. (b) The values of the regularization parameters that were chosen by cross validation for different covariance estimation methods.



(a) diagonal



(b) shrinkage



(c) glasso

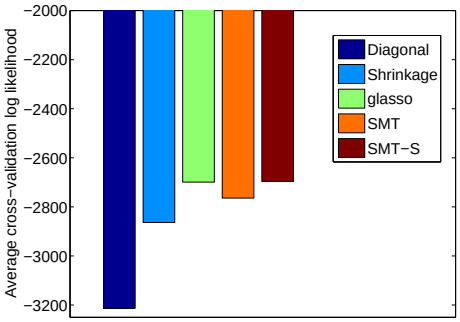


(d) SMT



(e) SMT-S

Fig. 4.10. Generated face image samples under the Gaussian distribution with the sample mean and different covariance estimates: (a) Diagonal (b) Shrinkage (c) Glasso (d) SMT (e) SMT-S.



(a)

Method	Log likelihood	Δ
Diagonal	-3213.3	-349.7
Shrinkage	-2863.6	0
glasso	-2699.1	164.5
SMT	-2764.2	99.4
SMT-S	-2696.3	167.3

(b)

Fig. 4.11. (a) The graph shows the average cross-validated log-likelihood of the face images using the diagonal, shrinkage, glasso, SMT and SMT-S covariance estimators. (b) The table shows the cross-validated log-likelihood for each estimator. 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).

a closed form global optimum. However, we show that a recursive local optimization procedure is simple, intuitive, and yields good results.

We also demonstrate that the proposed SMT covariance estimation methods substantially reduce the error in the covariance estimate as compared to current state-of-the-art estimates for a standard hyperspectral data and face image sets.

5. WEAK SIGNAL DETECTION IN HYPERSPECTRAL IMAGERY USING THE SPARSE MATRIX TRANSFORM (SMT) COVARIANCE ESTIMATION

5.1 Introduction

The covariance matrix is the cornerstone of multivariate statistical analysis. From radar [85] and remote sensing [79] to high finance [86], algorithms for the detection and analysis of signals require the estimation of a covariance matrix, often as a way to characterize the background clutter. For applications in hyperspectral imagery – where the covariance matrix is large, the estimation of that matrix from a limited number of samples is especially challenging. In this chapter, we will focus on signal detecion applications in hyperspectral imagery, and demonstrate how the SMT covariance estimation can be used to improve the performance of traditional detectors.

Many detection algorithms in hyperspectral image analysis, from well-characterized gaseous [87] and solid targets to deliberately uncharacterized anomalies and anomalous changes [88–91], depend on accurately estimating the covariance matrix of the background. In practice, the background covariance is estimated from samples in the image, and imprecision in this estimate can lead to a loss of detection power. The sample covariance is the most natural estimator, but particularly when the number of samples n (*e.g.,* number of pixels) is not much larger than the dimension p (*e.g.,* number of spectral channels), this is not necessarily the best estimate. Sliding window methods with the RX anomaly detector [88, 92], and segmentation methods for which a different covariance matrix is computed for different background classes [93], are just two examples where only a small number of samples is available for each covariance matrix that needs to be estimated. To mitigate the effect of undersampling, regularization is particularly important, and various kinds of regularization

have been proposed [65–68]. Here, we will consider the two recently developed regularizing schemes for covariance matrix estimation based on the sparse matrix transform (SMT) proposed in last chapter [27, 75].

Previously, the effectiveness of the SMT estimator was demonstrated in terms of eigenvalues and Kullback-Leibler distances between Gaussian distributions based on true and approximate covariance matrices. In this chapter, we investigate the performance of the adaptive matched filter, which depends on a covariance matrix estimate, when different regularizers are used. This work extends previous work by others investigating different approaches for regularizing the adaptive matched filter [92, 94, 95].

5.2 Review of the SMT Covariance Estimation

Given a p-dimensional Gaussian distribution with zero mean and covariance matrix $R \in \mathbb{R}^{p\times p}$, the likelihood of observing n samples, organized into a data matrix $X = [\mathbf{x}_1\mathbf{x}_2 \dots \mathbf{x}_n] \in \mathbb{R}^{p\times n}$, is given by

$$\mathcal{L}(R; X) = \frac{|R|^{-n/2}}{(2\pi)^{np/2}} \exp\left[-\frac{1}{2}\text{trace}\left(X^T R^{-1} X\right)\right]. \tag{5.1}$$

If the covariance is decomposed as the product $R = E\Lambda E^T$ where E is the orthogonal eigenvector matrix and Λ is the diagonal matrix of eigenvalues, then one can jointly maximize the likelihood with respect to E and Λ, which results in the maximum likelihood (ML) estimates [75]

$$\hat{E} = \arg\min_{E\in\Omega} \left\{\left|\text{diag}(E^T S E)\right|\right\} \tag{5.2}$$

$$\hat{\Lambda} = \text{diag}\left(\hat{E}^T S \hat{E}\right), \tag{5.3}$$

where $S = \left\langle \mathbf{x}\mathbf{x}^T \right\rangle = \frac{1}{n} XX^T$ is the sample covariance, and Ω is the set of allowed orthogonal transforms. Then $\hat{R} = \hat{E}\hat{\Lambda}\hat{E}^T$ is the ML estimate of the covariance.

Note that if S has full rank and Ω is the set of all orthogonal matrices, then the ML estimate of the covariance is given by the sample covariance: $\hat{R} = S$. However, if

$n < p$, then the sample covariance is singular and usually results in a poor estimate of the true covariance.

The sparse matrix transform (SMT) is proposed as a way to improve the covariance estimate by restricting the set Ω to a class of sparse eigenvector matrices E. The most sparse nontrivial orthogonal transform is the *Givens rotation* [18], which corresponds to a rotation by an angle θ in the plane of the i and j axes; specifically, it is given by $E = I + \Theta(i, j, \theta)$ where

$$\Theta(i,j,\theta)_{mn} = \begin{cases} \cos(\theta) - 1 & \text{if } m = n = i \text{ or } m = n = j \\ \sin(\theta) & \text{if } m = i \text{ and } n = j \\ -\sin(\theta) & \text{if } m = j \text{ and } n = i \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

Let E_k denote a Givens rotation, and note that a product of orthogonal rotations $E_1 E_2 \cdots E_k$ is still orthogonal. Let Ω_K be the set of orthogonal matrices that can be expressed as a product of K Givens rotations. The SMT covariance estimate is then given by Eq. (5.2) and Eq. (5.3) with $\Omega = \Omega_K$

$$\hat{E} = \arg\min_{E \in \Omega_K} \left\{ \left| \text{diag}(E^T S E) \right| \right\} \tag{5.5}$$

$$\hat{\Lambda} = \text{diag}\left(\hat{E}^T S \hat{E} \right). \tag{5.6}$$

Actually, the effective Ω is more restrictive than this, since we do not optimize over *all* possible products of K rotations, but instead greedily choose Givens rotations one at a time.

5.3 Matched Filter Criterion

One practical use for a covariance estimate in hyperspectral imagery is the detection of signals using a matched filter. The aim of the matched filter is to discriminate between pixels that are background clutter, and pixels that include signal [87]. If $\mathbf{x} \in \mathbb{R}^p$ is the observed spectrum of a background pixel, we use $\tilde{\mathbf{x}}$ for the spectrum

of that pixel if signal has been added. For the linear adaptive matched filter (AMF), the effect of the signal is modeled as an additive perturbation

$$\tilde{\mathbf{x}} = \mathbf{x} + \epsilon \mathbf{t} \ , \tag{5.7}$$

where ϵ is the signal intensity, and $\mathbf{t}$ is the signal's spectrum. For example, for a gaseous plume, this would correspond to the chemical spectrum of the gas.

A filter is a vector of coefficients $\mathbf{q} \in \mathbb{R}^p$, and the filter output for a given pixel is the scalar value $\mathbf{q}^T\mathbf{x}$. Generally, $\mathbf{q}^T\tilde{\mathbf{x}}$ is different than $\mathbf{q}^T\mathbf{x}$ since $\tilde{\mathbf{x}}$ contains signal. The goal is to find a $\mathbf{q}$ such that $\mathbf{q}^T\tilde{\mathbf{x}}$ is most distinguishable from $\mathbf{q}^T\mathbf{x}$.

The signal-to-clutter ratio for a filter $\mathbf{q}$ is given by

$$\mathrm{SCR} = \frac{(\mathbf{q}^T\mathbf{t})^2}{\langle\, (\mathbf{q}^T\mathbf{x})^2 \,\rangle} = \frac{(\mathbf{q}^T\mathbf{t})^2}{\mathbf{q}^T \langle\, \mathbf{x}\mathbf{x}^T \,\rangle\, \mathbf{q}} = \frac{(\mathbf{q}^T\mathbf{t})^2}{\mathbf{q}^T R \mathbf{q}} \ . \tag{5.8}$$

The *matched filter* is the filter that optimizes the SCR, and it is given, up to a constant multiplier, by $\mathbf{q} = R^{-1}\mathbf{t}$ [87]. As we can see, the matched filter is formed using the covariance of the background data. Using this $\mathbf{q}$ in Eq. (5.8), we get the optimal SCR:

$$\mathrm{SCR}_o = \frac{(\mathbf{t}^T R^{-1}\mathbf{t})^2}{\mathbf{t}^T R^{-1} R R^{-1}\mathbf{t}} = \mathbf{t}^T R^{-1}\mathbf{t} \ . \tag{5.9}$$

However, the true R is usually unkonw in practice. So we approximate the matched filter using an approximate covariance estimate $\hat{R}$. In this case, $\hat{\mathbf{q}} = \hat{R}^{-1}\mathbf{t}$ gives

$$\mathrm{SCR} = \frac{(\mathbf{t}^T \hat{R}^{-1}\mathbf{t})^2}{\mathbf{t}^T \hat{R}^{-1} R \hat{R}^{-1}\mathbf{t}} \ , \tag{5.10}$$

and the SCRR is the ratio

$$\frac{\mathrm{SCR}}{\mathrm{SCR}_o} = \frac{(\mathbf{t}^T \hat{R}^{-1}\mathbf{t})^2}{(\mathbf{t}^T \hat{R}^{-1} R \hat{R}^{-1}\mathbf{t})(\mathbf{t}^T R^{-1}\mathbf{t})}. \tag{5.11}$$

If $\hat{R} = R$, then SCRR $= 1$, but in general SCRR ≤ 1. The performance of this detector depends on the estimate of the covariance matrix of the background data; and the better the covariance is estimated, the more effective the detector.

5.4 Numerical Experiments

The analysis here is based on two hyperspectral datasets: one is the 224-channel AVIRIS image of the Florida coastline that was used in Ref. [91] (see Fig. 5.1); and one is the 191-channel image of the Washington DC mall that was used in Ref. [79] (see Fig. 4.3). For the Florida image, we used all 75,000 of the pixels in the image to estimate the "true" covariance; for the Washington data, we limited ourselves to the 1224 pixels that were labeled as "water." We have performed comparisons with a number of other data sets, and observed similar results.

We compare SMT and SMT-S to other regularization schemes. Shrinkage estimators are a widely used class of estimators which regularize the covariance matrix by shrinking it toward some target structures. Shrinkage estimators generally have the form $\hat{R} = \alpha D + (1 - \alpha)S$ where D is some positive definite matrix. Two popular choices of D are the scaled identity matrix, $\mathrm{trace}(S)/p \cdot I$ (called "Shrinkage-trI" in the plots), and the diagonal entries of S, $\mathrm{diag}(S)$ (called "Shrinkage-D" in the plots), and the corresponding shrinkage estimators are used for comparison. SMT-S is the shrinkage estimator with D being the SMT estimate.

The parameter K required by SMT can be efficiently determined by a simple cross-validation procedure. Specifically, we partition the samples into three subsets, and choose K to maximize the average likelihood of the left-out subset given the estimated covariance using the other two subsets. This cross-validation requires little additional computation since every E_k is a sparse operator. As the number of samples grows (*i.e.,* $n \approx 10p$), the optimal K for SMT can be very large. In experiments, we considered K up to $p(p-1)/2$.

For SMT-S, we found that we could more agressively limit the number of Givens rotations (we used $K \leq 10p$) and still obtain effective performance by choosing the shrinkage coefficient α to maximize the cross-validated likelihood.

In Fig. 5.2, the performance of different covariance estimators is compared using the SCRR statistic. In Fig. 5.2(a,b,d,e), we see that SMT and the SMT-S algorithms

outperformed the other regularization schemes over a range of values of n, with the most dramatic improvement at the smallest sample sizes. This behavior was *not* seen in Fig. 5.2(c,f), however, and we discuss this in the following section.



Fig. 5.1. Broadband image of the 224-channel hyperspectral AVIRIS data used in the experiments. The image is 150×500 pixels, and was obtained from flight $f960323t01p02_r04_sc01$ over the Florida coast.

5.4.1 Structure in the Covariance Matrix

It is widely appreciated that real hyperspectral data is not fully modeled by a multivariate Gaussian distribution [96, 97]. But in addition to this structure *beyond* the Gaussian, there also appears to be structure *within* the covariance matrix.

We hypothesize that this structure is exploited by SMT and that that explains how SMT can outperform the other regularizers. We tested this hypothesis by randomly rotating the covariance matrices. A random orthogonal matrix Q (obtained from a QR decomposition of a matrix whose elements were independently chosen from a Gaussian distribution) was used to rotate the matrices: that is, $R' = QRQ^T$. Fig. 5.2(c,f) shows the performance of different covariance estimators applied to randomly rotated covariance matrices. Indeed, these panels show that the rotationally invariant Shrinkage-trI is the estimator with the best performance.

The performance of the sample covariance was similar for all of these cases. This is consistent with the theoretical result, due to Reed *et al.* [85], that for $n \gg p$, the expected value of SCRR is given by $1 - p/n$. (For $n < p$, the sample covariance is singular and the matched filter is undefined.)

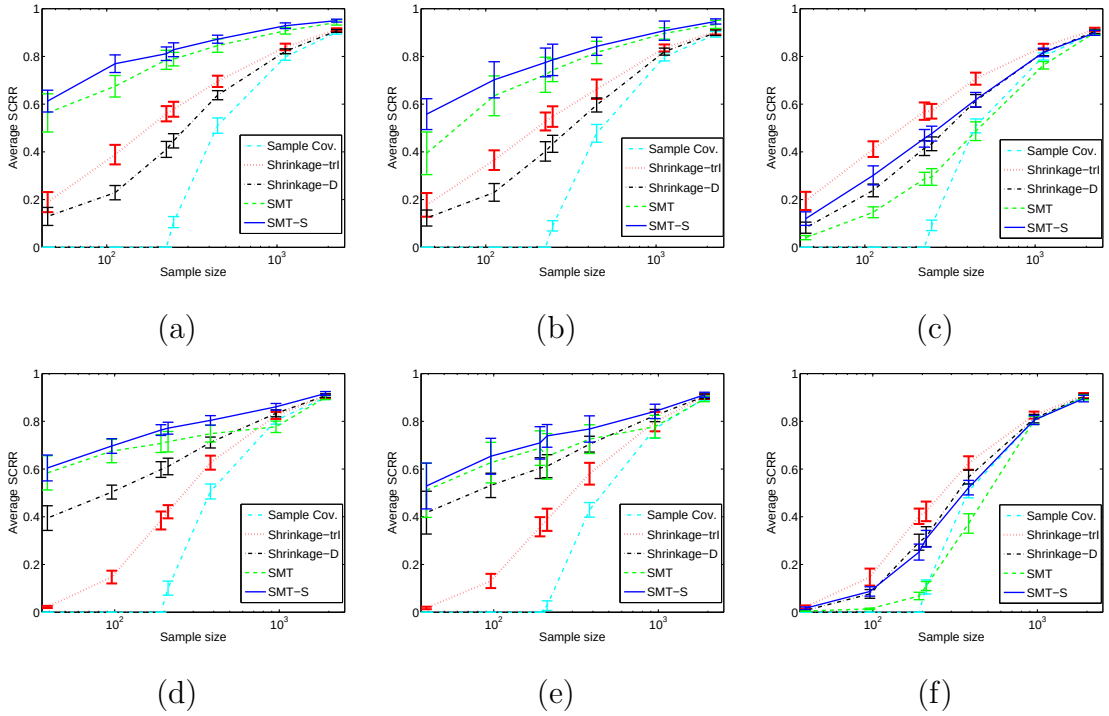


Fig. 5.2. Average of SCRR as a function of sample size for (a,b,c) Florida image and (d,e,f) Washington image. In all cases, the target signals are randomly generated from a Gaussian distribution, and the error bars are based on runs with 30 trials. (a,d) Gaussian samples are generated from the "true" covariance matrices for these two images. (b,e) Non-Gaussian samples are drawn at random with replacement from the image data itself. (c,f) Gaussian samples generated from randomly rotated covariance matrices. All plots are based on 30 trials, and each trial used a different rotation (for the randomly rotated covariances) and a different target $\mathbf{t}$.

One appeal of algorithms based on the covariance matrix is that they are often rotationally invariant. If we rotate our data $\mathbf{x}$ via some linear transform $\hat{\mathbf{x}} = L\mathbf{x}$, then the analysis on $\hat{\mathbf{x}}$ uses a different covariance matrix $\hat{R} = LRL^T$, but rotationally invariant algorithms will detect signal at the same pixels and achieve the same performance.

Rotationally invariant algorithms are attractive, but there are properties of the data which are manifestly not rotationally invariant. For instance, radiance or reflectance is always non-negative but arbitrary linear combinations can produce negative values. This is not a "problem" in the sense that the algorithms which exploit this data do not require non-negative values; but it does point to unexploited structure in the data.

This structure can be illustrated by considering the covariance matrix R from a hyperspectral image. We can express this matrix in terms of a product of eigenvectors and eigenvalues, $R = E\Lambda E^T$, and then observe an image of the eigenvector matrix E. Fig. 5.3 illustrates such images, based on the Florida data and on the Washington data. We see that the eigenvector images, in Fig. 5.3(b,f) are sparse, particularly compared to their randomly rotated counterparts in Fig. 5.3(c,g). The histograms in Fig. 5.3(d,h), with their sharp peaks at zero, further emphasize this sparsity.

We suggest that the behavior seen in Fig. 5.3 may be relatively generic. We consider some simulated data where each pixel is independently chosen from a uniform distribution so that the mth channel has distribution in the range $[0, r_m]$ where r_m is itself a number chosen in the range $[0, 1]$. This property of different spectral channels having different characteristic amplitudes matches the intuition that different spectral wavelengths with in generally have different physical properties. This is an example of a property that would not hold under a random rotation of the data. Fig. 5.4 shows that although the rough alignment of channel number with principal component (seen in Fig. 5.3(b,f) is not observed in this simulated case, we do see a sparse eigenvector matrix and a sharp histogram of eigenvector values.

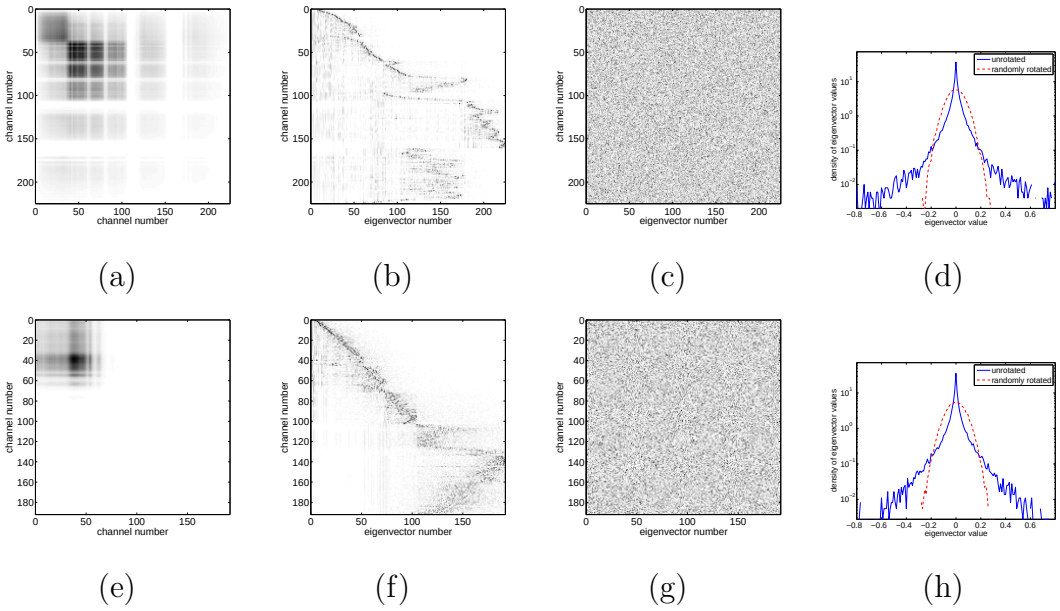


Fig. 5.3. Non-rotationally-invariant structure in the covariance matrix of real hyperspectral data is evident in the image of eigenvectors for (a,b,c,d) Florida data and for (e,f,g,h) Washington data. In (a,e) the covariance matrix is shown with larger values of R_{ij} plotted darker; in (b,f) the matrix E of eigenvectors of R is shown with larger values of the absolute value $|E_{ij}|$ shown darker; in (c,g) the eigenvectors are shown for a randomly rotated covariance matrix; and in (d,h) a histogram of eigenvector values is shown for both the original and the randomly rotated covariance matrix.

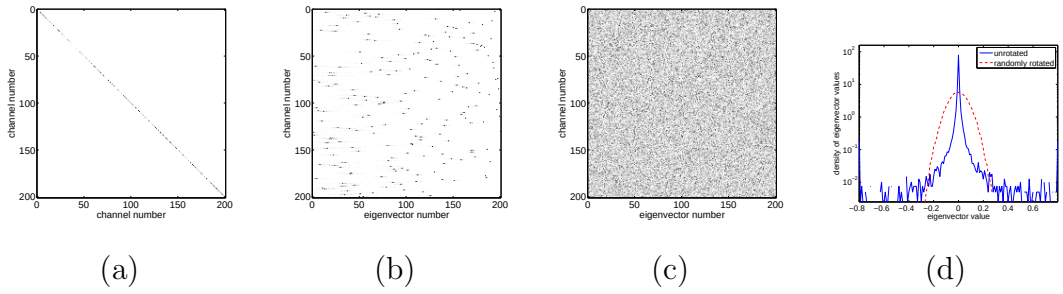


Fig. 5.4. Same as Fig. 5.3 but for simulated data. $p = 200$ channels, and 75000 pixels. The data were generated in such a way that each pixel is independently chosen from a uniform distribution so that the mth channel has distribution in the range $[0, r_m]$ where r_m is itself a number chosen in the range $[0, 1]$.

5.5 Conclusion

In this chapter, we describe the sparse matrix transform (SMT) covariance estimation and investigate its utility for weak signal detection in hyperspectral imagery. Experiments shows that using the SMT to estimate the covariance matrix in the adaptive matched filter leads to consistently higher signal-to-clutter ratios than other regulization methods. This improvement may be due to the fact that the SMT covariance estimates are able to capture non-rotationally-invariant structures that exist in the hyperspectral data.

6. HIGH DIMENSIONAL REGRESSION USING THE SPARSE MATRIX TRANSFORM (SMT)

6.1 Introduction

Regression from high dimensional observation vectors is particularly difficult when training data is limited. Traditional regression methods that use the sample covariance, such as the ordinary least squares (OLS) approach, perform poorly in this situation. This is because, if the sample number n is less than data dimension p, then the sample covariance is singular, with at least $p-n$ of the smallest eigenvalues estimated to be zero. In this case, the sample covariance does not accurately characterize any signal that falls in the subspaces of the observations corresponding to the smallest eigenvalues of the observations.

In the past decades, regression methods that adopt regularization have been introduced, such as ridge regression [98], subset selection, and Lasso [99]. More recently, there has also been increasing interest in replacing the sample covariance with some sparse estimates of the true covariance or its inverse for high dimensional regression problems [100, 101].

In this chapter, we propose a novel regression approach that first decorrelates the high dimensional observation vector using the sparse matrix transform (SMT) estimate of the covariance [27, 75]. To improve the prediction accuracy, model selection is then performed by regularizing the regression in the domain of the decorrelated data. In particular, we explore the use of both Lasso and shrinkage methods for this regularized regression step. While the technique we propose can be used with other estimates of the covariance, we have found that SMT covariance estimation results in relatively good estimates particularly when $n < p$ [27, 81]. The SMT covariance esti-

mate achieves this improved accuracy by imposing a constraint that the eigenvector transformation should be formed by a product of Givens rotations [18].

Our numerical results demonstrate that the both the SMT-Lasso and SMT-Shrinkage regression methods can significantly improve the prediction performance when $n < p$, and that, for our experiments, the SMT-Lasso method yields better results than the SMT-Shrinkage method, but at the cost of greater computational cost.

6.2 Regression Model

Without loss of generality, let $y \in \Re^{n\times 1}$ be a vector of n i.i.d. zero-mean Gaussian random variables which we would like to estimate. Our observations are $X \in \Re^{n\times p}$, a matrix containing n independent zero mean Gaussian random row vectors, each of dimension p. The minimum mean square error (MSEE) estimate of y given X has the form

$$\hat{y} = Xb \tag{6.1}$$

where b is a vector of regression coefficients given by

$$b = R_x^{-1}\rho \ , \tag{6.2}$$

where R_x is the covariance of the observations X, and ρ is the correlation between the observations X and y given by

$$\begin{aligned} R_x &= \frac{1}{n}\mathrm{E}[X^t X] \\ \rho &= \frac{1}{n}\mathrm{E}[X^t y] \ . \end{aligned}$$

Of course, in practice R_x and ρ are often unknown, so that b must be estimated from training data (y, X). This problem has been widely studied over the years, and most recently has become of particular interest in the challenging case when $n < p$. The traditional method for solving the regression problem is ordinary least squares (OLS). However, OLS becomes ill-posed when $n < p$, so ridge regression [98], and the Lasso [99] have been proposed as alternatives. The methods of partial least squares (PLS) have also been widely used for this problem [102].

6.3 SMT Regression for High Dimensional Data

Our approach will be to estimate y based on the assumption that we can accurately estimate the covariance R_x. Our approach is motivated by a variety of new methods for estimating high dimensional covariance matrices through the use of sparsity constraints [27,68,69]. In particular, we will use the SMT covariance estimation method, which has been shown to produce substantially more accurate covariance estimates for certain physical data through the introduction of a covariance model [27, 75, 81]. Importantly, the SMT covariance estimate can accurately produce all the eigenvalues of the covariance even when $n < p$, and the resulting estimate is typically full rank.

Perhaps surprisingly, we find that even when the exact value of the covariance is used in Eq (6.2), the resulting regression coefficients may yield estimates that are inferior to established methods. Intuitively, this is because the correlation ρ must also be accurately determined. Therefore, having an accurate estimate of R_x does not insure success.

Therefore, our proposed regression approach is based on two steps. In the first step, we decorrelate the observations using the estimate of R_x. In the second step, we estimate the regression coefficients in this decorrelated domain. We propose three possible methods for estimating these regression coefficients. The first method, which we refer to as SMT-Lasso, applies the Lasso regression method in the decorrelated domain. The second method, which we refer to as SMT-Shrinkage, shrinks the regression coefficients toward zero; and the third method, which we refer to as SMT-Subset Selection, simply selects the coordinates which are most correlated with the y.

For all the three methods, we use the SMT covariance estimate to decorrelate the observation data in the first step. Let $\hat{R}_x$ be the covariance estimate, and let the eigen decomposition of the covariance estimate be given by

$$\hat{R}_x = \hat{E}\hat{\Lambda}\hat{E}^t \ , \tag{6.3}$$

where $\hat{E}$ is the orthonormal matrix of eigenvectors and $\hat{\Lambda}$ is a diagonal matrix of eigenvalues. Using these estimated quantities, we can approximately decorrelate and whiten the observed data using the transformation

$$\tilde{X} = X\hat{E}\hat{\Lambda}^{-\frac{1}{2}} \ . \tag{6.4}$$

For the SMT covariance estimate, the entries of the diagonal matrix $\hat{\Lambda}$ are generally positive, so the matrix is invertible.

Using the whitened observations $\tilde{X}$, the estimate of y can now be expressed as

$$\hat{y} = \tilde{X}\beta \ . \tag{6.5}$$

Since $\tilde{X}$ is a linear transformation of the observations X, it does not change the OLS estimate. However, it can change other regression results based on nonlinear estimators of the regression coefficients.

An important special case occurs if the observations are perfectly whitened. In this case, $\frac{1}{n}E\left[\tilde{X}^t\tilde{X}\right] = I$, and the regression parameters for MMSE estimation are given by

$$\beta = \frac{1}{n}\mathrm{E}[\tilde{X}^t y] \ . \tag{6.6}$$

The question remains of how to compute an effective estimate of β. For this purpose, we propose three methods.

6.3.1 SMT-Lasso

The first method we propose for estimating β, which we call SMT-Lasso, is based on the use of least squares minimization with an L_1 norm constrain on β [99]. The SMT-Lasso estimate is given by

$$\hat{\beta} = \arg\min_{\beta}\left\{\parallel y - \tilde{X}\beta \parallel^2 + \lambda \parallel \beta \parallel_1\right\} \ . \tag{6.7}$$

Notice that the solution to (6.7) depends on the specific whitening transformation used in (6.4) because the L_1 norm is not invariant to orthonormal rotations. Therefore, SMT-Lasso will produce a different solution from conventional Lasso performed

in the native coordinates of X. Since the columns of the matrix $\tilde{X}$ are not exactly orthogonal, the SMT-Lasso regression coefficients are computed as the solution to a quadratic programing problem. As with conventional Lasso, this optimization problem can be computed using a variety of efficient techniques [99, 103].

6.3.2 SMT-Shrinkage

The second method we propose for estimating β, which we call SMT-Shrinkage, is based on the approximation that the columns of $\tilde{X}$ are orthogonal, and have an L_2 norm of n. More specifically, we assume that

$$\frac{1}{n}\tilde{X}^t\tilde{X} \approx I \ . \tag{6.8}$$

In fact, the form of the SMT covariance estimate ensures that $\frac{1}{n}\text{diag}\left(\tilde{X}^t\tilde{X}\right) = I$. So the columns of $\tilde{X}$ have a constant norm of n. In addition, the columns are approximately orthogonal because SMT covariance estimation attempts to minimize correlation between columns subject to a constraint on the required number of sparse rotations.

Using this approximation, the solution to (6.7) can be computed by first solving the unconstrained optimization problem to yield

$$\hat{\beta} = \frac{1}{n}\tilde{X}^t y \ , \tag{6.9}$$

and then applying the soft shrinkage operator to yield

$$\hat{\beta}_{\gamma i} = \text{sign}(\hat{\beta}_i)(\mid \hat{\beta}_i \mid -\gamma)^+ \ , \text{for } i = 1, 2, \ldots, p \ , \tag{6.10}$$

where the operator $(\cdot)^+$ returns the positive portion of the argument. Notice that this soft shrinkage operator has the same form of wavelet shrinkage for noise removal as propose in [104].

6.3.3 SMT-Subset Selection

The third method we propose for estimating β, which we call SMT-Subset Selection, is similar to SMT-Shrinkage, but it selects a subset of decorrelated components for the regression, rather than shrinking the components toward zero.

For SMT-Subset selection, we first compute $\hat{\beta}$ of (6.9). We then apply the following operation to each component of $\hat{\beta}$

$$\hat{\beta}_{\gamma i} = \begin{cases} \hat{\beta}_i & \text{if } \mid \hat{\beta}_i \mid > \gamma \\ 0 & \text{otherwise} \end{cases} . \tag{6.11}$$

Notice that this operation selects the components that have the largest correlation with the observations to be predicted.

The regularization parameter value, such as λ in Eq (6.7), or γ in Eq (6.10) and Eq (6.11), can be estimated using cross validation or Stein's unbiased risk estimator (SURE) theory [105, 106].

6.4 Alternative Regression Methods

This section lists out some alternative methods for regression. We will compare our proposed regression approaches with these methods in the numerical simulations.

6.4.1 Ordinary Least Squares Regression

In practice we do not usually know the true covariance and the correlation coefficient, therefore, the sample covariance $S = \frac{1}{n} X^t X$ and sample correlation $\frac{X^t y}{n}$ are generally used to estimate the regression coefficient vector b, which is called ordinary least squares (OLS) regression. OLS can be formulated as the unconstrained optimization problem

$$\hat{b} = \arg\min_b \parallel y - Xb \parallel^2 , \tag{6.12}$$

and the solution is given by

$$\hat{b} = (X^t X)^{-1} \left(X^t y \right) . \tag{6.13}$$

We know the OLS estimator is unbiased, and it has lowest variance among all linear unbiased estimators [98]. However, its variance can be very high for high-dimensional data, compared to regularized estimators that are usually biased. Especially, if $p > n$, the sample covariance is singular and non-invertible.

6.4.2 Ridge Regression

Ridge regression imposes a L_2 penalty on the magnitude of the regression coefficients. It can be formulated as the constrained optimization problem

$$\hat{b} = \arg\min_{b} \left\{ \| y - Xb \|^2 + \lambda \| b \|^2 \right\} \ . \tag{6.14}$$

6.4.3 Traditional Lasso Regression

Traditional Lasso regression is usually performed in the native coordinates of X with L_1 regularization on the regression coefficients [99]

$$\hat{b} = \arg\min_{b} \left\{ \| y - Xb \|^2 + \lambda \| b \|_1 \right\} \ . \tag{6.15}$$

6.5 Numerical Experiments

In this section, we compare the accuracy of various regression methods using the following model for $X \in \Re^{n \times p}$

$$X = y \cdot \tau^t + W \ , \tag{6.16}$$

where $\tau \in \Re^{p \times 1}$ is a deterministic but unknown signal, $y \in \Re^{n \times 1}$ is a vector of n independent Gaussian random variables to be estimated, and each row of $W \in \Re^{n \times p}$ is an independent p-dimensional Gaussian random vector of correlated clutter or noise. Without loss of generality, we assume that W, y, and X are all zero mean, and that elements of y have unit variance.

For all numerical experiments, the assumed clutter covariance, $R_w = \frac{1}{n}\mathrm{E}[W^t W]$, is computed from the real hyperspectral data shown in Fig. 6.1 with dimension $p = 191$



(a) (b) (c)

Fig. 6.1. (a) Simulated color IR view of an airborne hyperspectral data over the Washington DC Mall [79]. (b) Ground-truth pixel spectrum of grass. (c) Ground-truth pixel spectrum of water.

[79]. We use two covariance matrices in our experiments corresponding to "grass" and "water" classes, and for these two classes, $\frac{1}{p}\text{trace}\{R_w\}$ is equal to 9.68×10^4 and 2.83×10^4, respectively. Accordingly, we know that

$$R_x = R_w + \tau \cdot \tau^t \ . \tag{6.17}$$

All comparisons are done in terms of the signal-to-noise ratio,

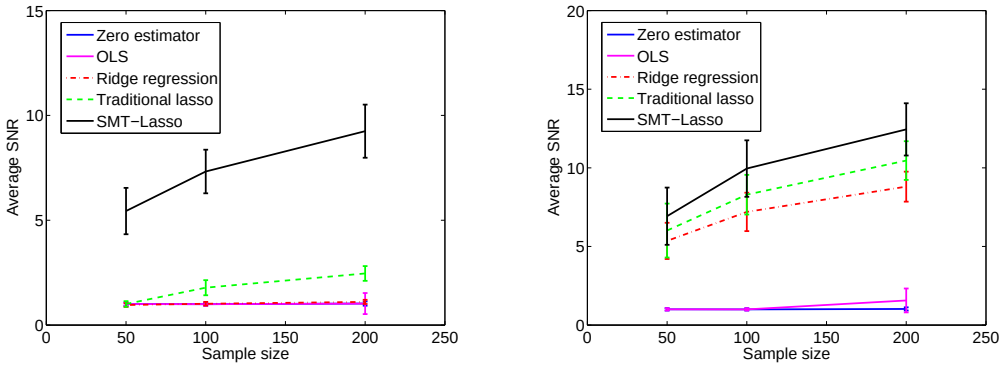
$$SNR = \frac{\| \ y \ \|^2}{\| \ y - X\hat{b} \ \|^2} \ , \tag{6.18}$$

which is computed for a set of 300 data vectors that are generated independently of the training data. Also, in each case, the SNR is averaged over 30 simulations, each of which uses a different realization of the cluster and signal.

6.5.1 When τ Is a Small Eigenvector

Here, we investigate the case when the signal, τ, falls in a subspace corresponding to small eigenvalues of the clutter/noise. To do this, we choose τ to be the 170-th eigenvector of R_w (with eigenvalues sorted from largest to smallest). Furthermore, the data is scaled so that $||\tau||^2 = 3^2$. The experiments are run for $n = 50, 100$, and 200, and the results are compared with the zero estimator ($\hat{b} = 0$), OLS, ridge regression, and traditional Lasso regression.

Figure 6.2 shows the plots of the average SNR as a function of the sample size n for both the "grass" and "water" covariance matrices. In Fig. 6.2(a) we can see the



(a) grass (b) water

Fig. 6.2. Plots of average SNR when τ is the 170-th eigenvector of R_w. Notice that SMT-Lasso regression results in the highest SNR in the range of $n < p$. (a) Clutter W is generated using hyperspectral grass data. (b) Clutter W is generated using hyperspectral water data.

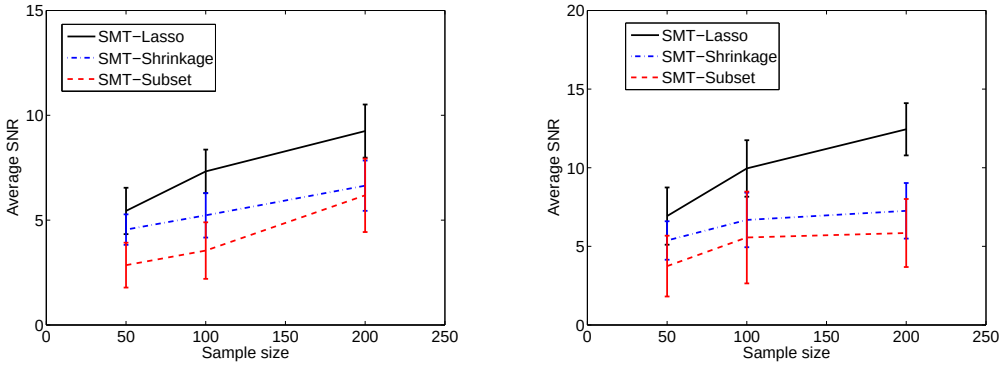
traditonal reguralized regression methods perform poorly; however, they do improve the prediction accuracy in some cases as shown in Fig. 6.2(b). Notice that the SMT-Lasso regression results in the highest SNRs in the whole range of $n < p$ for both cases. This is because the SMT covariance estimate is able to represent the signal that exists in small eigenvector subspaces of the clutter.

6.5.2 When τ Is a Random Signal

Here, τ is generated as a Gaussian random vector with zero mean and covariance I. The amplitude of τ is then scaled so that $\| \tau \|^2 = 4^2$.

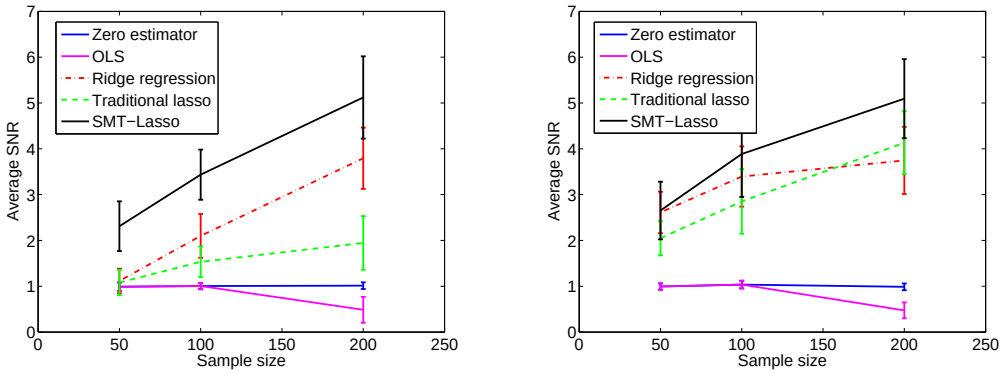
Figure 6.4 shows the plots of the average SNRs as a function of the sample size n for the "grass" and "water" classes. Notice again that the SMT-Lasso regression results in consistently higher SNR in the range of $n < p$ compared to the other methods.

Figure 6.3 and 6.5 shows the SNR for each of the methods SMT-Lasso, SMT-Shrinkage, and SMT-Subset selection. SMT-Lasso performs best but is more computationally expensive than the others.



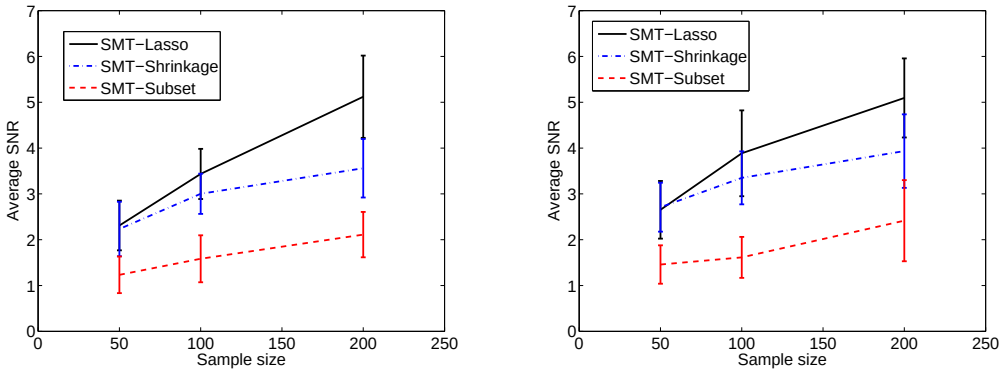
(a) grass (b) water

Fig. 6.3. SMT-Lasso versus SMT-Shrinkage versus SMT-Subset when τ is the 170-th eigenvector of R_w. SMT-Lasso works best, but are much more computationally expensive. (a) Clutter W is generated using hyperspectral grass data. (b) Clutter W is generated using hyperspectral water data.



(a) grass (b) water

Fig. 6.4. Plots of average SNR when τ is a random Gaussian signal. Notice that SMT-Lasso regression results in consistently higher SNR in the range of $n < p$ compared to the other regression methods. (a) Clutter W is generated using hyperspectral grass data. (b) Clutter W is generated using hyperspectral water data.



(a) grass (b) water

Fig. 6.5. SMT-Lasso versus SMT-Shrinkage versus SMT-Subset when τ is a random Gaussian signal. SMT-Lasso works best, but is more computationally expensive. (a) Clutter W is generated using hyperspectral grass data. (b) Clutter W is generated using hyperspectral water data.

6.6 Conclusions

In this chapter, we proposed a novel regression approach for high dimensional data. In this approach, the SMT covariance estimate is computed and used to decorrelate the data, and then different model selection methods are used to obtain good estimates of regression coefficients in the decorrelated data domain. Numerical examples show that the proposed approach can significantly improve the SNR of the regression models, especially for 'small n, large p' problems.

LIST OF REFERENCES

LIST OF REFERENCES

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[2] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 674–693, July 1989.

[3] E. E. Fenimore and T. M. Cannon, "Coded aperture imaging with uniformly redundant arrays," *Applied Optics*, vol. 17, no. 3, p. 337, 1978.

[4] Y. Bresler, M. Gastpar, and R. Venkataramani, "Image compression on-the-fly by universal sampling in fourier imaging systems," in *Proceedings of 1999 IEEE Information Theory Workop on Detection, Estimation, Classification, and Imaging*, February 1999.

[5] E. J. Candes, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, (Madrid, Spain), August 2006.

[6] D. L. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, pp. 1289– 1306, April 2006.

[7] C. F. Van Loan, *Computational frameworks for the fast Fourier transform*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1992.

[8] P. A. Regalia and S. K. Mitra, "Kronecker products, unitary matrices and signal processing applications," *SIAM Review*, vol. 31, pp. 586–613, December 1989.

[9] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, USA: Academic Press, 1998.

[10] T. Hebert and R. Leahy, "A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors," *IEEE Trans. on Medical Imaging*, vol. 8, pp. 194–202, June 1989.

[11] K. Sauer and C. A. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Trans. on Signal Processing*, vol. 41, pp. 534–548, February 1993.

[12] J.-B. Thibault, K. D. Sauer, C. A. Bouman, and J. Hsieh, "A three-dimensional statistical approach to improved image quality for multislice helical CT," *Medical Physics*, vol. 34, no. 11, pp. 4526–4544, 2007.

[13] S. Geman and D. E. McClure, "Statistical methods for tomographic image reconstruction," *Bulletin of the International Statistical Institute*, pp. LII–4:5–21, 1987.

[14] E. Levitan and G. T. Herman, "A maximum a posteriori probability expectation maximization algorithm for image reconstruction in emission tomography," *IEEE Trans. on Medical Imaging*, vol. 6, no. 3, pp. 185–192, 1987.

[15] P. J. Green, "Bayesian reconstructions from emission tomography data using a modified EM algorithm," *IEEE Trans. on Medical Imaging*, vol. 9, no. 1, pp. 84–93, 1990.

[16] G. Cao, C. A. Bouman, and K. J. Webb, "Fast and efficient stored matrix techniques for optical tomography," in *Proceedings of the 40th Asilomar Conference on Signals, Systems, and Computers*, October 2006.

[17] G. Cao, C. A. Bouman, and K. J. Webb, "Fast reconstruction algorithms for optical tomography using sparse matrix representations," in *Proceedings of 2007 IEEE International Symposium on Biomedical Imaging*, April 2007.

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

[19] H. Rutishauser, "The Jacobi method for real symmetric matrices," *Numerische Mathematik*, vol. 9, pp. 1–10, 1966.

[20] P. A. Regalia, "An adaptive unit norm filter with applications to signal analysisand Karhunen-Loeve transformations," *IEEE Trans. on Circuits and Systems*, vol. 37, pp. 646–649, May 1990.

[21] P. Comon, "Independent component analysis: A new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[22] N. Delfosse and P. Loubaton, "Adaptive blind separation of independent sources: a deflation approach," *Signal Process.*, vol. 45, no. 1, pp. 59–83, 1995.

[23] E. Moreau and J.-C. Pesquet, "Generalized contrasts for multichannel blind deconvolution oflinear systems," *IEEE Signal Processing Letters*, vol. 4, pp. 182–183, Jun. 1997.

[24] A. Hyvarinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, pp. 411–430, May-Jun. 2000.

[25] 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.

[26] 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.

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

[28] A. P. Gibson, J. C. Hebden, and S. R. Arridge, "Recent advances in diffuse optical imaging," *Phys. Med. Biol*, vol. 50, pp. R1–R43, February 2005.

[29] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems*. New York: Winston and Sons, 1977.

[30] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation, Compression, and Standards*. New York: Plenum Press, 1995.

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

[32] D. Tretter and C. A. Bouman, "Optimal transforms for multispectral and multilayer image coding," *IEEE Trans. on Image Processing*, vol. 4, pp. 296–308, March 1995.

[33] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, pp. 205–220, April 1992.

[34] P. Dragotti, G. Poggi, and A. Ragozini, "Compression of multispectral images by three-dimensional SPIHT algorithm," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 38, pp. 416–428, January 2000.

[35] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[36] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, pp. 297–301, April 1965.

[37] D. K. Hoffman, R. C. Raffenetti, and K. Ruedenberg, "Generalization of euler angles to n-dimensional orthogonal matrices," *Journal of Mathematical Physics*, vol. 13, no. 4, pp. 528–533, 1972.

[38] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, Inc., 1991.

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

[40] J. Eriksson and V. Koivunen, "Characteristic-function-based independent component analysis," *Signal Processing*, vol. 83, no. 10, pp. 2195–2208, 2003.

[41] D. A. Boas, D. H. Brooks, E. L. Miller, C. A. DiMarzio, M. Kilmer, R. J. Gaudette, and Q. Zhang, "Imaging the body with diffuse optical tomography," *IEEE Signal Processing Magazine*, vol. 18, pp. 57–75, November 2001.

[42] A. B. Milstein, S. Oh, K. J. Webb, C. A. Bouman, Q. Zhang, D. A. Boas, and R. P. Millane, "Fluorescence optical diffusion tomography," *Applied Optics*, vol. 42, pp. 3081–3094, June 2003.

[43] M. S. Patterson, J. D. Moultan, B. C. Wilson, K. W. Berndtand, and J. R. Lakowicz, "Frequency-domain reflectance for the determination of the scattering and absorption properties of tissue," *Applied Optics*, vol. 30, pp. 4474–4476, 1991.

[44] M. A. O'Leary, D. A. Boas, B. Chance, and A. G. Yodh, "Refraction of diffuse photon density waves," *Physical Review Letters*, vol. 69, pp. 2658–2661, 1992.

[45] Q. Zhang, T. J. Brukilacchio, A. Li, J. J. S. T. Chaves, E. Hillman, T. Wu, M. Chorlton, E. Rafferty, R. H. Moore, D. B. Kopans, and D. A. Boas, "Coregistered tomographic x-ray and optical breast imaging: initial results," *J. Biomed. Opt.*, vol. 10, pp. 024033,1–9, March/April 2005.

[46] L. C. Enfield, A. P. Gibson, N. L. Everdell, D. T. Delpy, M. Schweiger, S. R. Arridge, C. Richardson, M. Keshtgar, M. Douek, and J. C. Hebden, "Three-dimensional time-resolved optical mammography of the uncompressed breast," *Applied Optics*, vol. 46, no. 17, pp. 3628–3638, 2007.

[47] J. C. Ye, K. J. Webb, C. A. Bouman, and R. P. Millane, "Optical diffusion tomography using iterative coordinate descent optimization in a Bayesian framework," *J. Optical Society America A*, vol. 16, pp. 2400–2412, October 1999.

[48] C. A. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Trans. on Image Processing*, vol. 2, pp. 296–310, July 1993.

[49] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier Analysis and Applications*, vol. 4, pp. 247–269, May 1998.

[50] R. Roy, A. B. Thompson, A. Godavarty, and E. M. Sevick-Muraca, "Tomographic fluorescence imaging in tissue phantoms: A novel reconstruction algorithm and imaging geometry," *IEEE Trans. on Medical Imaging*, vol. 24, pp. 137–154, February 2005.

[51] J. Wei, G. Cao, C. Bouman, and J. Allebach, "Fast space-varying convolution and its application in stray light reduction," in *Proceedings of the SPIE/IS&T Conference on Computational Imaging VII*, January 2009.

[52] M. G. Erickson, J. S. Reynolds, and K. J. Webb, "Comparison of sensitivity for single and dual interfering source configurations in optical diffusion imaging," *J. Optical Society America A*, vol. 14, pp. 3080–3092, November 1997.

[53] Y. Chen, G. Zheng, Z. H. Zhang, D. Blessington, M. Zhang, H. Li, Q. Liu, L. Zhou, X. Intes, S. Achilefu, and B. Chance, "Metabolism-enhanced tumor localization by fluorescence imaging: *in vivo* animal studies," *Optics Letters*, vol. 28, no. 21, pp. 2070–2072, 2003.

[54] I. Gannot, A. Garashi, G. Gannot, V. Chernomordik, and A. Gandjbakhche, "*In vivo* quantitative three-dimensional localization of tumor labeled with exogenous specific fluorescence markers," *Applied Optics*, vol. 42, pp. 3073–3080, June 2003.

[55] A. B. Milstein, M. D. Kennedy, P. S. Low, C. A. Bouman, and K. J. Webb, "Detection and localization of a fluorescing mouse tumor in a turbid medium," *Applied Optics*, vol. 44, pp. 2300–2310, April 2005.

[56] G. Cao, V. Gaind, C. A. Bouman, and K. J. Webb, "Localization of an absorbing inhomogeneity in a scattering medium in a statistical framework," *Opt. Lett.*, vol. 32, no. 20, pp. 3026–3028, 2007.

[57] A. B. Milstein, S. Oh, J. S. Reynolds, K. J. Webb, C. A. Bouman, and R. P. Millane, "Three-dimensional Bayesian optical diffusion tomography with experimental data," *Optics Letters*, vol. 27, pp. 95–97, 2002.

[58] A. Y. Bluestone, G. Abdoulaev, C. H. Schmitz, R. L. Barbour, and A. H. Hielscher, "Three-dimensional optical tomography of hemodynamics in the human head," *Opt. Express*, vol. 9, pp. 272–286, Sept. 10 2001.

[59] J. P. Culver, A. M. Siegel, J. J. Stott, and D. A. Boas, "Volumetric diffuse optical tomography of brain activity," *Optics Letters*, vol. 28, pp. 2061–2063, 2003.

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

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

[62] 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.

[63] 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.

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

[65] 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.

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

[67] 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.

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

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

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

[71] 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.

[72] 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.

[73] 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.

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

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

[76] G. Cao, C. Bouman, and K. Webb, "Noniterative map reconstruction using sparse matrix representations," *IEEE Transactions on Image Processing*, vol. 18, pp. 2085–2099, Sept. 2009.

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

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

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

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

[81] 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.

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

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

[84] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pp. 138–142, Dec 1994.

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

[86] 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.

[87] J. Theiler, B. R. Foy, and A. M. Fraser, "Beyond the adaptive matched filter: nonlinear detectors for weak signals in high-dimensional clutter," in *Proceedings of the SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIII*, vol. 6565, (Orlando FL), April 9 2007.

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

[89] A. Schaum and A. Stocker, "Long-interval chronochrome target detection," *Proc. 1997 International Symposium on Spectral Sensing Research*, 1998.

[90] A. Schaum and A. Stocker, "Linear chromodynamics models for hyperspectral target detection," *Proc. IEEE Aerospace Conference*, pp. 1879–1885, 2003.

[91] J. Theiler, "Quantitative comparison of quadratic covariance-based anomalous change detectors," *Applied Optics*, vol. 47, pp. F12–F26, 2008.

[92] N. M. Nasrabadi, "Regularization for spectral matched filter and RX anomaly detector," *Proc. SPIE*, vol. 6966, p. 696604, 2008.

[93] C. C. Funk, J. Theiler, D. A. Roberts, and C. C. Borel, "Clustering to improve matched filter detection of weak gas plumes in hyperspectral imagery," *IEEE Trans. Geoscience and Remote Sensing*, vol. 39, pp. 1410–1420, 2001.

[94] P. V. Villeneuve, H. A. Fry, J. Theiler, B. W. Smith, and A. D. Stocker, "Improved matched-filter detection techniques," *Proc. SPIE*, vol. 3753, pp. 278–285, 1999.

[95] A. De Maio, "Fast converging adaptive matched filter and adaptice cosine/coherence estimator," *Signal Processing*, vol. 82, pp. 1417–1423, 2002.

[96] D. Manolakis, D. Marden, J. Kerekes, and G. Shaw, "On the statistics of hyperspectral imaging data," *Proc. SPIE*, vol. 4381, pp. 308–316, 2001.

[97] J. Theiler, B. R. Foy, and A. M. Fraser, "Characterizing non-Gaussian clutter and detecting weak gaseous plumes in hyperspectral imagery," in *Proceedings of the SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI*, vol. 5806, (Orlando FL), March 28 2005.

[98] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2001.

[99] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[100] X. J. Jeng, Z. J. Daye, and Y. Zhu, "Sparse covariance thresholding for high-dimensional variable selection with the lasso," Technical Report TR 08-07, Purdue University, Statistics Department, 2008.

[101] D. M. Witten and R. Tibshirani, "Covariance-regularized regression and classification for high dimensional problems," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 3, pp. 615–636, 2009.

[102] D. Haaland and E. Thomas, "Partial least-squares methods for spectral analyses," *Analytical Chemistry*, 1988.

[103] S. Alliney and S. Ruzinsky, "An algorithm for the minimization of mixed l(1) and l(2) norms with application to bayesian estimation," *IEEE Transactions on Signal Processing*, vol. 42, no. 3, pp. 618–627, 1994.

[104] D. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, pp. 1200–1224, 1995.

[105] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The Annals of Statistics*, vol. 9, no. 6, pp. 1135–1151, 1981.

[106] M. Ulfarsson and V. Solo, "Dimension estimation in noisy pca with sure and random matrix theory," *Signal Processing, IEEE Transactions on*, vol. 56, pp. 5804–5816, Dec. 2008.

APPENDICES

A. COMPUTATION OF H

Here we describe the method used for computing the inverse matrix H. Let h_i be the i^{th} column of H, and let e_i be the unit measurement vector which is 1 for entry i and zero for all other entries. Then, h_i can be computed as the reconstructed image given the measurement e_i.

$$h_i = \arg\min_x \parallel e_i - Ax \parallel_\Lambda^2 + x^t S x \tag{A.1}$$

Thus, iterative methods such as the conjugate gradient (CG) method can be used to solve this problem. Actually, it is more sensible to compute $\tilde{H}$ directly instead of H. The i^{th} column of $\tilde{H}$ can be computed as

$$\tilde{h}_i = \arg\min_x \parallel \tilde{e}_i - Ax \parallel_\Lambda^2 + x^t S x \ , \tag{A.2}$$

where $\tilde{e}_i = E\Lambda_y^{1/2} e_i$. It would perhaps be desirable to directly compute the sparse representation $\check{H}$, but this computation requires Φ which depends on the inverse matrix H.

B. FAST COMPUTATION OF SMT

Here we show that each butterfly of the SMT can be computed with 2 multiplies. First, we show that for a pair-wise sparse transform matrix T_k and a diagonal matrix S_k, there exists another pair-wise sparse transform matrix $\tilde{T}_k$ and diagonal matrix $\tilde{S}_k$ such that

$$T_k \cdot S_k = \tilde{S}_k \cdot \tilde{T}_k \ . \tag{B.1}$$

Without loss of generality, (B.1) can be easily verified that in the 2D case. First assume either $s_1 t_{11}$ or $s_2 t_{22}$ is not zero, then we have

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \cdot \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} = \begin{bmatrix} \tilde{s}_1 & 0 \\ 0 & \tilde{s}_2 \end{bmatrix} \cdot \begin{bmatrix} 1 & b \\ a & 1 \end{bmatrix} \ , \tag{B.2}$$

where

$$\begin{cases} \tilde{s}_1 = s_1 t_{11} \\ \tilde{s}_2 = s_2 t_{22} \end{cases} \text{ and } \begin{cases} a = \frac{s_1 t_{21}}{s_2 t_{22}} \\ b = \frac{s_2 t_{12}}{s_1 t_{11}} \end{cases} . \tag{B.3}$$

If either $s_1 t_{11}$ or $s_2 t_{22}$ is zero, e.g. $s_1 t_{11} = 0$, then we have

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \cdot \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & s_2 t_{22} \end{bmatrix} \cdot \begin{bmatrix} 0 & s_2 t_{12} \\ \frac{s_1 t_{21}}{s_2 t_{22}} & 1 \end{bmatrix} \ . \tag{B.4}$$

If both $s_1 t_{11}$ and $s_2 t_{22}$ are zero, then we have

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \cdot \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & s_2 t_{12} \\ s_1 t_{21} & 0 \end{bmatrix} \ . \tag{B.5}$$

Starting from $S_0 = I$ and iterating (B.1) with $S_{k+1} = \tilde{S}_k$, we can obtain

$$\check{y} = \left(\prod_{k=K-1}^{0} T_k \right) y = \tilde{S}_{K-1} \left(\prod_{k=K-1}^{0} \tilde{T}_k \right) y \ . \tag{B.6}$$

Notice that each multiplication by $\tilde{T}_k$ requires two multiples, so the evaluation of $\check{y}$ using the SMT requires a total of $2K + M$ multiplies. This implementation is similar to the fast Givens transformation of [31].

C. OPTIMALITY OF T_{OPT} FOR THE SMT COST FUNCTION

Here we show that the exact transform T_{opt} in (2.21) is the solution to the minimization of (2.40). First we prove some properties of symmetric, positive definite matrices, e.g., any covariance matrix.

Theorem C.0.1 *If $R \in \mathbb{R}^{n\times n}$ is a symmetric, positive definite matrix, then we have*

$$\mid \mathrm{diag}(R) \mid \geq \mid R \mid \ . \tag{C.1}$$

The equality holds if and only if R is diagonal.

Proof We know there exists a unique low triangular matrix $G \in \mathbb{R}^{n\times n}$, such that

$$R = GG^t \ , \tag{C.2}$$

which is called Cholesky factorization [31]. Therefore, $\mid R \mid = \mid G \mid^2 = \prod_{i=1}^{n} G_{ii}^2$. Clearly, we have $R_{ii} = \sum_{j=1}^{n} G_{ij}^2 \geq G_{ii}^2$, for $i = 1, 2, \ldots, n$. This gives

$$\mid \mathrm{diag}(R) \mid \geq \prod_{i=1}^{n} G_{ii}^2 = \mid R \mid \ . \tag{C.3}$$

The equality holds if and only if $R_{ii} = G_{ii}^2$ for $i = 1, 2, \ldots, n$, which is equivalent to the fact that R is diagonal. ■

Now we show that $T_{opt} = \Phi^t \Lambda_y^{-\frac{1}{2}} E^t$ minimizes the cost function (2.40) among all nonsigular matrices, i.e.

$$T_{opt} = \arg\min_{T} \mid \mathrm{diag}\left(\left(T^{-1}\right)^t R_H T^{-1}\right) \mid \cdot \mid \mathrm{diag}\left(T R_y T^t\right) \mid \ . \tag{C.4}$$

Proof Since the covariance matrices R_H and R_y are symmetric and positive definite, $(T^{-1})^t R_H T^{-1}$ and $T R_y T^t$ are also symmetric and positive definite for any nonsigular matrix $T \in \mathbb{R}^{n\times n}$. Therefore, from Theorem C.0.1 we have

$$\begin{aligned} &| \operatorname{diag}\left((T^{-1})^t R_H T^{-1}\right) | \cdot | \operatorname{diag}\left(T R_y T^t\right) | \\ \geq &| (T^{-1})^t R_H T^{-1} | \cdot | T R_y T^t | = | R_H | \cdot | R_y | \ . \end{aligned} \tag{C.5}$$

The equality in (C.5) holds if and only if both $(T^{-1})^t R_H T^{-1}$ and $T R_y T^t$ are diagonal. In another words, the cost function is minimized if and only if R_H and R_y are simultaneously diagonized by T. Now, let $T_{opt} = \Phi^t \Lambda_y^{-\frac{1}{2}} E^t$. This leads to

$$T_{opt} R_y T_{opt}^t = I \tag{C.6}$$

$$\left(T_{opt}^{-1}\right)^t R_H T_{opt}^{-1} = \Lambda_{\tilde{H}} \ , \tag{C.7}$$

where $\Lambda_{\tilde{H}}$ is defined in (2.18). Therefore, both of the transformed covariance matrices of R_y and R_H using T_{opt} are diagonal, and hence the cost function is minimized. By substituting the expression of T_{opt} into the SMT cost function (2.40), we can also explicitly show

$$\begin{aligned} &| \operatorname{diag}\left(\left(T_{opt}^{-1}\right)^t R_H T_{opt}^{-1}\right) | \cdot | \operatorname{diag}\left(T_{opt} R_y T_{opt}^t\right) | \\ = &| R_H | \cdot | R_y | \ . \end{aligned} \tag{C.8}$$

■

D. OPTIMAL GIVENS ROTATION

In this appendix, we find the solution to the optimization problem of (2.41) for a specified coordinate index 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 C, we know that T^* minimizes the cost function (2.41) if and only if it simultaneously diagonalizes both R_y and R_H. To do this, let $T^* = B\Lambda A$, and let

$$R_y = \begin{bmatrix} \sigma_{y11}^2 & \sigma_{y12} \\ \sigma_{y21} & \sigma_{y22}^2 \end{bmatrix} \text{ and } R_H = \begin{bmatrix} \sigma_{h11}^2 & \sigma_{h12} \\ \sigma_{h21} & \sigma_{h22}^2 \end{bmatrix} . \tag{D.1}$$

A Givens rotation $A = I + \Theta(1, 2, \theta)$ with $\theta = \frac{1}{2}\text{atan}(2\sigma_{y12}, \sigma_{y11}^2 - \sigma_{y22}^2)$ diagonalizes R_y, i.e.

$$AR_yA^t = \begin{bmatrix} \sigma_{y11}'^2 & 0 \\ 0 & \sigma_{y22}'^2 \end{bmatrix} , \tag{D.2}$$

where

$$\begin{aligned} \sigma_{y11}'^2 &= \frac{1}{2}\left(\sigma_{y11}^2 + \sigma_{y22}^2 + \sqrt{(\sigma_{y11}^2 - \sigma_{y22}^2)^2 + 4\sigma_{y12}^2}\right) \\ \sigma_{y22}'^2 &= \frac{1}{2}\left(\sigma_{y11}^2 + \sigma_{y22}^2 - \sqrt{(\sigma_{y11}^2 - \sigma_{y22}^2)^2 + 4\sigma_{y12}^2}\right) . \end{aligned}$$

Notice that for the special case that $\sigma_{y11}^2 = \sigma_{y22}^2 = 1$, we can always choose $\theta = \pi/4$, which leads to

$$\sigma_{y11}'^2 = 1 + \sigma_{y12} \tag{D.3}$$

$$\sigma_{y22}'^2 = 1 - \sigma_{y12} . \tag{D.4}$$

If we define $\Lambda = \begin{bmatrix} \frac{1}{\sigma'_{y11}} & 0 \\ 0 & \frac{1}{\sigma'_{y22}} \end{bmatrix}$, then this leads to

$$R_{y'} = \Lambda A R_y A^t \Lambda = I \tag{D.5}$$

$$R_{H'} = \Lambda^{-1} A R_H A^t \Lambda^{-1} = \begin{bmatrix} \sigma'^2_{h11} & \sigma'_{h12} \\ \sigma'_{h21} & \sigma'^2_{h22} \end{bmatrix} , \tag{D.6}$$

where

$$\begin{aligned} \sigma'^2_{h11} &= \sigma'^2_{y11} \left[\tfrac{1+\cos 2\theta}{2} \sigma^2_{h11} + \sigma_{h12} \sin 2\theta + \tfrac{1-\cos 2\theta}{2} \sigma^2_{h22} \right] \\ \sigma'^2_{h22} &= \sigma'^2_{y22} \left[\tfrac{1-\cos 2\theta}{2} \sigma^2_{h11} - \sigma_{h12} \sin 2\theta + \tfrac{1+\cos 2\theta}{2} \sigma^2_{h22} \right] \\ \sigma'_{h12} &= \sigma'_{y11} \sigma'_{y22} \left[\tfrac{1}{2} (\sigma^2_{h22} - \sigma^2_{h11}) \sin 2\theta + \sigma_{h12} \cos 2\theta \right] , \end{aligned}$$

and $\sigma'_{h21} = \sigma'_{h12}$. Then, a Givens rotation $B = I + \Theta(1, 2, \phi)$ with $\phi = \frac{1}{2}\text{atan}(2\sigma'_{h12}, \sigma'^2_{h11} - \sigma'^2_{h22})$ makes $B R_{H'} B^t$ diagonal while maintaining $B R_{y'} B^t = I$. For the special case that $\sigma^2_{y11} = \sigma^2_{y22} = 1$, the simplified expression for B is given in (2.48).

So we see that both $T^* R_y T^{*t}$ and $\left(T^{*-1}\right)^t R_H T^{*-1}$ are diagonal with $T^* = B\Lambda A$ given above. Hence, T^* must minimize the cost function of (2.41). Based on (C.8), we know that the ratio of the cost function before and after the transform of T^* is given as

$$\begin{aligned} \frac{C(T^*, R_H, R_y)}{C(I, R_H, R_y)} &= \frac{\mid R_H \mid \cdot \mid R_y \mid}{\mid \text{diag}\,(R_H) \mid \cdot \mid \text{diag}\,(R_y) \mid} \\ &= \left(1 - \frac{\sigma^2_{y12}}{\sigma^2_{y11}\sigma^2_{y22}}\right) \left(1 - \frac{\sigma^2_{h12}}{\sigma^2_{h11}\sigma^2_{h22}}\right) . \end{aligned} \tag{D.7}$$

E. RUN-LENGTH CODING

Here we give the details of the run-length encoder [30] used in numerical experiments. After quantization of $\check{H}$, the run-length encoder codes the first voxel position of each non-zero run along every column, the run length and the non-zero voxel values. Specifically, for the ODT example, a 32-bit integer is used to store the voxel position, and an 8-bit integer is used to store the run length. To identify the column index of every run, we assign one extra bit to show weather it belongs to the same column as the previous run. An 8-bit or 16-bit integer is used to store every quantized non-zero voxel value depending on its magnitude, and an extra bit is assigned to label the choice. For the FODT example, the run-length encoder is almost the same as in the ODT example except that an 16-bit integer is used for storing the first voxel position of each non-zero run.

For encoding the SMT (i.e. T_k's), we use an 16-bit integer for storing index i_k and j_k, respectively, and an 8-byte double float for each of the two non-zero off-diagonal entries.

F. 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

$$\begin{aligned} P_{(E,\Lambda)}(Y) &= \frac{1}{(2\pi)^{\frac{np}{2}}} |R|^{-\frac{n}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\{Y^t R^{-1} Y\}\right\} && \text{(F.1)} \\ &= \frac{1}{(2\pi)^{\frac{np}{2}}} |\Lambda|^{-\frac{n}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\{Y^t E\Lambda^{-1} E^t Y\}\right\} && \text{(F.2)} \\ &= \frac{1}{(2\pi)^{\frac{np}{2}}} |\Lambda|^{-\frac{n}{2}} \exp\left\{-\frac{1}{2}\mathrm{tr}\{E^t Y Y^t E\Lambda^{-1}\}\right\} && \text{(F.3)} \\ &= \frac{1}{(2\pi)^{\frac{np}{2}}} |\Lambda|^{-\frac{n}{2}} \exp\left\{-\frac{n}{2}\mathrm{tr}\{E^t S E\Lambda^{-1}\}\right\} && \text{(F.4)} \\ &= \frac{1}{(2\pi)^{\frac{np}{2}}} |\Lambda|^{-\frac{n}{2}} \exp\left\{-\frac{n}{2}\mathrm{tr}\{\mathrm{diag}(E^t S E)\Lambda^{-1}\}\right\} \ . && \text{(F.5)} \end{aligned}$$

Taking the logarithm yields

$$\log P_{(E,\Lambda)}(Y) = -\frac{n}{2}\mathrm{tr}\{\mathrm{diag}(E^t S E)\Lambda^{-1}\} - \frac{n}{2}\log|\Lambda| - \frac{np}{2}\log(2\pi) \ . \qquad \text{(F.6)}$$

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

$$\begin{aligned} (\hat{E}, \hat{\Lambda}) &= \arg\max_{(E,\Lambda)} \log P_{(E,\Lambda)}(Y) && \text{(F.7)} \\ &= \arg\max_{E} \max_{\Lambda} \log P_{(E,\Lambda)}(Y) \ . && \text{(F.8)} \end{aligned}$$

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

$$\hat{\Lambda} = \mathrm{diag}(E^t S E) \ . \qquad \text{(F.9)}$$

Therefore, the ML estimation of E is given by

$$\begin{aligned} \hat{E} &= \arg\max_{E\in\Omega} \log P_{(E,\hat{\Lambda}(E))}(Y) && \text{(F.10)} \\ &= \arg\max_{E\in\Omega} \left\{ -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log\left|\mathrm{diag}(E^t S E)\right| - \frac{np}{2} \right\} && \text{(F.11)} \\ &= \arg\min_{E\in\Omega} \left\{ \left|\mathrm{diag}(E^t S E)\right| \right\} \ , && \text{(F.12)} \end{aligned}$$

where Ω is the set of allowed orthonormal transforms. So minimization of $|\mathrm{diag}(E^t S E)|$ leads to the ML estimate of E, and hence the ML estimate of Λ which is given by

$$\hat{\Lambda} = \mathrm{diag}(\hat{E}^t S \hat{E}) \ . \qquad \text{(F.13)}$$

G. UNCONSTRAINED ML ESTIMATE

Proposition G.0.1 *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 (4.6), so that*

$$\left|\mathrm{diag}(E^{*t} S E^*)\right| = \min_{E \in \Omega} \left\{ \left|\mathrm{diag}(E^t S E)\right| \right\} \ . \tag{G.1}$$

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

$$|\mathrm{diag}(S)| \geq |S| \ . \tag{G.2}$$

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

$$S = GG^t \ , \tag{G.3}$$

which is called the Cholesky factorization [31]. 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

$$|\mathrm{diag}(S)| \geq \prod_{i=1}^p G_{ii}^2 = |S| \ . \tag{G.4}$$

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

$$\left|\mathrm{diag}(E^t S E)\right| \geq \left|E^t S E\right| = |S| \ . \tag{G.5}$$

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

$$\left|\mathrm{diag}(E^{*t} S E^*)\right| = |\Lambda_S| = |S| \ . \tag{G.6}$$

Therefore, E^* is the solution of global minimization of (4.6) if the sample covariance S is non-singular. ■

H. EXACT SMT FACTORIZATION OF ORTHONORMAL TRANSFORMS

We know the Givens QR factorization can be used to find a decomposition of an $p \times p$ matrix into $\binom{p}{2}$ Givens rotations [31]. 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

$$A = QR \ ,$$

where every G_k is a Givens rotation and R is upper triangular. Since A and Q are orthonormal, R must be orthonormal. Therefore, the columns of R must be orthogonal. 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 $A = Q$.

I. GREEDY ALGORITHM FOR THE SMT DESIGN

In this appendix, we will find the solution to the greedy optimization problem of (4.12) 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 G, we know that E minimizes the cost function (4.12) if and only if E is the matrix of eigenvectors of S, i.e. $S = E\Lambda_S E^t$. Next we obtain an expression for E in terms of a Givens rotation. Let

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} , \tag{I.1}$$

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

$$E^t S E = \begin{bmatrix} s'_{11} & 0 \\ 0 & s'_{22} \end{bmatrix} , \tag{I.2}$$

where

$$s'_{11} = \frac{1}{2}\left(s_{11} + s_{22} + \sqrt{(s_{11} - s_{22})^2 + 4s_{12}^2}\right) \tag{I.3}$$

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

This shows that E of the given form is the matrix of eigenvectors of S. Hence E must minimize the cost function of (4.12). Based on (G.6), we know that the ratio of the cost function before and after the transform of E is given as

$$\frac{|\text{diag}\,(E^t S E)|}{|\text{diag}\,(S)|} = \frac{|S|}{|\text{diag}\,(S)|} = 1 - \frac{s_{12}^2}{s_{11}s_{22}} . \tag{I.5}$$

J. PROOF OF PERMUTATION INVARIANCE OF THE SMT ESTIMATOR

Property 4.4.1: If $\hat{R} = \hat{E}\hat{\Lambda}\hat{E}^t$ is the unique order-K SMT covariance estimate of 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 Eq (4.14) has a unique minimum at each step $k \leq K$. Let $\tilde{R} = \tilde{E}\tilde{\Lambda}\tilde{E}^t$ be the order-K SMT covariance estimator of the permuted data PY. By the construction of E, we know $\tilde{E} = P\hat{E}P^t$, and $\hat{E}^t S\hat{E} = P^t\tilde{E}^t\tilde{S}\tilde{E}P$ where $\tilde{S} = PYY^tP^t/n$, for any Y and P under the assumption of uniqueness. Thus we have

$$\begin{aligned} \tilde{\Lambda} &= \text{diag}(\tilde{E}^t\tilde{S}\tilde{E}) \\ &= \text{diag}(P\hat{E}^tS\hat{E}P^t) \\ &= P\text{diag}(\hat{E}^tS\hat{E})P^t \\ &= P\hat{\Lambda}P^t \ . \end{aligned} \tag{J.1}$$

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

$$\tilde{R} = \tilde{E}\tilde{\Lambda}\tilde{E}^t = P\hat{E}\hat{\Lambda}\hat{E}^tP^t = P\hat{R}P^t \ . \tag{J.2}$$

K. KULLBACK-LEIBLER DISTANCE

The Kullback-Leibler (KL) distance for two distributions $P_\theta(y)$ and $P_{\hat{\theta}}(y)$ is defined as [80]

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

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

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

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

VITA

VITA

Guangzhi Cao received his B.S. and M.S. degrees in Electrical Engineering from Zhejiang University, China, in 2002 and 2004, respectively. He is currently working towards his Ph.D. degree in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), Optical Society of America (OSA), and the Eta Kappa Nu (HKN) Honor Society. His current research interests include statistical signal and image processing, inverse problems, machine learning and computer vision.