

Noniterative MAP Reconstruction Using Sparse Matrix Representations

Guangzhi Cao, *Student Member, IEEE*, Charles A. Bouman, *Fellow, IEEE*, and Kevin J. Webb, *Fellow, IEEE*

Abstract—We present a method for noniterative maximum *a posteriori* (MAP) tomographic reconstruction which is based on the use of sparse matrix representations. Our approach is to precompute 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 fully populated (i.e., not sparse). 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 transformations, which we call a sparse-matrix transform (SMT). The SMT is a generalization of the classical FFT in that it uses butterflies to compute an orthonormal transform; but unlike an FFT, the SMT uses the butterflies in an irregular pattern, and is numerically designed to best approximate the desired transforms. We demonstrate the potential of the noniterative MAP reconstruction with examples from optical tomography. The method requires offline computation to encode the inverse transform. However, once these offline computations are completed, the noniterative MAP algorithm is shown to reduce both storage and computation by well over two orders of magnitude, as compared to a linear iterative reconstruction methods.

Index Terms—Inverse problems, matrix source coding, noniterative reconstruction, optical tomography, sparse matrix representation.

I. 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

Manuscript received February 06, 2008; revised April 25, 2009. This work was supported by the National Science Foundation under Contract CCR-0431024. First published June 23, 2009; current version published August 14, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter C. Doerschuk.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-2035 USA (e-mail: gcao@purdue.edu; bouman@purdue.edu; webb@purdue.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2009.2023724

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 statistical 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 noniterative MAP approach requires an offline computation in which the inverse transform matrix is compressed and encoded. However, once this offline computation is complete, the online 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 noniterative 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 noniterative MAP algorithm reduces online storage and computation by well over 2 orders of magnitude, as compared to a traditional iterative reconstruction method.

II. NONITERATIVE 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

$$y = Ax + w \quad (1)$$

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 noninvertible 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

$$\hat{x} = \arg \max_x \{\log p(y|x) + \log p(x)\} \quad (2)$$

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

$$\hat{x} = \arg \min_x \|y - Ax\|_{\Lambda}^2 + x^t S x. \quad (3)$$

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 (3) may be computed in closed form as

$$\hat{x} = (A^t \Lambda A + S)^{-1} A^t \Lambda y. \quad (4)$$

Therefore, if we precompute the inverse matrix

$$H \triangleq (A^t \Lambda A + S)^{-1} A^t \Lambda \quad (5)$$

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

$$\hat{x} = Hy. \quad (6)$$

The noniterative computation of the MAP estimate in (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, noniterative computation of the MAP estimate is rarely used because the matrix H can be both enormous and nonsparse 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 (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 (6) may be efficiently computed, and so the matrix H may be efficiently stored.

III. LOSSY SOURCE CODING OF THE INVERSE MATRIX

A. Encoding Framework

For convenience, we assume that both the columns of H and the measurements y have zero mean.¹ For the 3-D tomographic reconstruction problem, the columns of H are 3-D 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

$$H = [H] + \delta H \quad (7)$$

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

$$\delta \hat{x} = \delta H y. \quad (8)$$

B. 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 1:

$$\mathbb{E} [\|\delta \hat{x}\|^2 | \delta H] = \|\delta H\|_{R_y}^2 \quad (9)$$

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

Proof:

$$\begin{aligned} \mathbb{E} [\|\delta \hat{x}\|^2 | \delta H] &= \mathbb{E} [y^t \delta H^t \delta H y | \delta H] \\ &= \mathbb{E} [\text{trace}\{\delta H y y^t \delta H^t\} | \delta H] \\ &= \text{trace}\{\delta H \mathbb{E}[y y^t] \delta H^t\} \\ &= \text{trace}\{\delta H R_y \delta H^t\}. \end{aligned} \quad (10)$$

From this result, we have the following immediate corollary. ■

Corollary 1: If $R_y = I$, then

$$\mathbb{E} [\|\delta \hat{x}\|^2 | \delta H] = \|\delta H\|^2. \quad (11)$$

Corollary 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

¹Let 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 nonzero mean.

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

$$R_y = E \Lambda_y E^t \quad (12)$$

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

$$\tilde{H} = H E \Lambda_y^{1/2} \quad (13)$$

$$\tilde{y} = \Lambda_y^{-1/2} E^t y. \quad (14)$$

Notice that with these definitions $\mathbb{E}[\tilde{y}\tilde{y}^t] = I$, and $\hat{x} = \tilde{H}\tilde{y}$. As in the case of (8), the distortion in \hat{x} due to quantization of \tilde{H} may be written as

$$\delta \hat{x} = \delta \tilde{H} \tilde{y} \quad (15)$$

where $\delta \tilde{H}$ denotes the quantization error in \tilde{H} . Using the result of Corollary 1 and the fact that \tilde{y} is whitened, we then know that

$$\mathbb{E}[\|\delta \hat{x}\|^2 | \delta \tilde{H}] = \|\delta \tilde{H}\|^2. \quad (16)$$

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

C. 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

$$\check{H} = W^t \tilde{H} \Phi \quad (17)$$

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^t 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^t to be a 3-D orthonormal wavelet transform. We do this because the columns of \tilde{H} are 3-D images, and wavelet transforms are known to approximate the 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 \check{H} for our 3-D reconstruction problems.

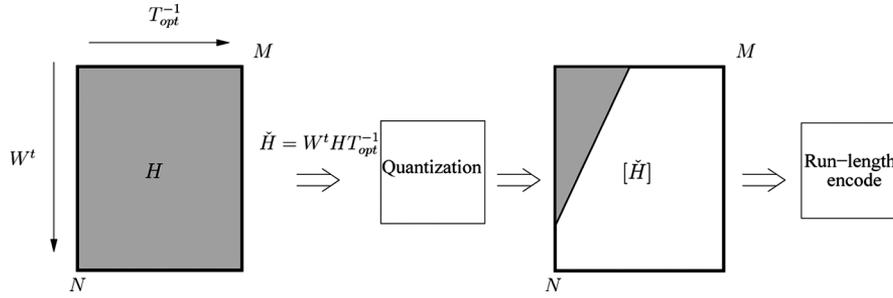


Fig. 1. Illustration of matrix source coding procedure. \tilde{H} is a transformed representation of H using both KL and wavelet transforms. The gray regions represent the effective nonzero entries in the matrices. Notice that the two transforms concentrate the energy in \tilde{H} toward the upper left hand corner, so that quantization results in a very sparse matrix. This procedure is performed offline, so that online reconstruction can be very fast.

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

$$R_{\tilde{H}} = \Phi \Lambda_{\tilde{H}} \Phi^t \quad (18)$$

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

$$\tilde{H} = W^t H T_{\text{opt}}^{-1} \quad (19)$$

$$\tilde{y} = T_{\text{opt}} y \quad (20)$$

where T_{opt} is defined as

$$T_{\text{opt}} = \Phi^t \Lambda_y^{-1/2} E^t \quad (21)$$

with E , Λ_y and Φ given in (12) and (18), respectively. Using \tilde{H} and \tilde{y} , the reconstruction is computed as

$$\hat{x} = W \tilde{H} \tilde{y}. \quad (22)$$

Finally, the sparse representation \tilde{H} is quantized and encoded. Since Φ is orthonormal, the vector \tilde{y} has covariance $E[\tilde{y} \tilde{y}^t] = I$, and by Corollary 1 minimum MSE quantization of \tilde{H} will achieve minimum MSE reconstruction of \hat{x} . Since the objective is to achieve minimum MSE quantization of \tilde{H} , we quantize each entry of \tilde{H} with the same quantization step size and denote the resulting quantized matrix as $[\tilde{H}]$. 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 $[\tilde{H}]$, 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 \tilde{H} in order to quantize and encode it. Appendix A discusses some details of how this can be done.

In summary, the noniterative reconstruction procedure requires two steps. In the first offline step, matrix source coding is used to compute the sparse matrix $[\tilde{H}]$. In the second online step, the approximate reconstruction is computed via the relationship

$$\hat{x} \approx W [\tilde{H}] T_{\text{opt}} y. \quad (23)$$

By changing the quantization step size, we can control the accuracy of this approximation, but at the cost of longer reconstruct-

tion times and greater storage for $[\tilde{H}]$. Of course, the question arises of how much computation will be required for the evaluations of the matrix-vector product $T_{\text{opt}} y$. This question will be directly addressed in Section IV through the introduction of the SMT transform.

Importantly, matrix source coding is done offline as a pre-computation step, but the operations of (23) are done online during reconstruction. Fig. 1 illustrates the procedure for the offline step of matrix source coding, and Fig. 2(a) lists a pseudo-code procedure for its implementation. The gray regions of Fig. 1 graphically illustrate nonzero 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 $[\tilde{H}]$ into the upper lefthand region.

Fig. 2(b) lists the pseudo-code for the online reconstruction of (23). Notice, that since the matrix $[\tilde{H}]$ is very sparse, the computation required to evaluate $[\tilde{H}] \tilde{y}$ is dramatically reduced. Also, notice that the inverse wavelet transform is only applied once, after \tilde{x} is computed in order to reduce computation.

IV. SPARSE MATRIX TRANSFORM

Step 1 of the online reconstruction procedure in Fig. 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

$$T_{\text{opt}} \approx \prod_{k=K-1}^0 T_k = T_{K-1} T_{K-2} \cdots T_0 \quad (24)$$

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

Off-line Processing: Lossy matrix source coding of H

- 1) Measurement whitening:

$$(E, \Lambda_y) \leftarrow \text{EigenDecomposition}(R_y)$$

$$\tilde{H} \leftarrow HE\Lambda_y^{-\frac{1}{2}}$$

- 2) Decorrelation of the columns of
- \tilde{H}
- :

$$R_{\tilde{H}} \leftarrow \frac{1}{N} \tilde{H}^t \tilde{H}$$

$$(\Phi, \Lambda_{\tilde{H}}) \leftarrow \text{EigenDecomposition}(R_{\tilde{H}})$$

$$\tilde{H} \leftarrow \tilde{H}\Phi$$

$$T_{opt} \leftarrow \Phi^t \Lambda_y^{-\frac{1}{2}} E^t$$

- 3) Wavelet transform of each column:

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

- 4) Quantization and coding of
- \tilde{H}
- :

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

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

(a)

On-line Processing: Evaluation of $H\tilde{y}$

- 1) Measurement data transform:

$$\tilde{y} \leftarrow T_{opt} y$$

- 2) Decoding of
- $[\tilde{H}]$

- 3) Reconstruction of image in the wavelet domain:

$$\tilde{x} \leftarrow [\tilde{H}]\tilde{y}$$

- 4) Final reconstructed image after the inverse wavelet transform:

$$\hat{x} \leftarrow W\tilde{x}$$

(b)

Fig. 2. Implementation of noniterative MAP reconstruction. (a) The offline processing algorithm for the matrix source coding of the inverse matrix H . (b) The online processing to compute the reconstruction from the matrix-vector product.

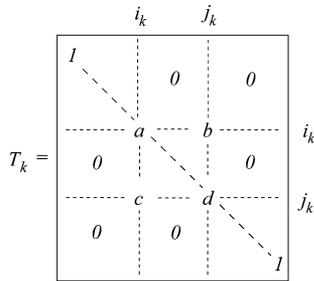


Fig. 3. Structure of a pair-wise sparse transform T_k . Here, all the unlabeled diagonal elements are 1's, and all the unlabeled off-diagonal elements are 0's. A_k , B_k and T_k have similar structures.

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 (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. 3. In general, any such pair-wise transform T_k can be represented in the form

$$T_k = B_k \Lambda_k A_k \quad (25)$$

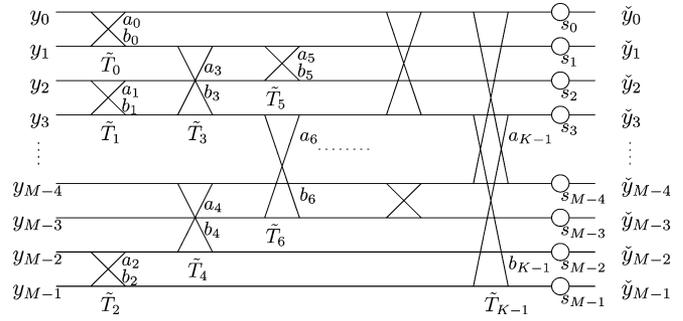


Fig. 4. Structure of SMT implementation. Every \tilde{T}_k is a “butterfly” that can be computed using 2 multiplies. In addition, multiplications by normalization factors is required in the end for a total of $2K + M$ multiplies when K butterflies are used in an M -point transform. The irregular structure of the SMT makes it a generalization of the FFT and allows it to be used to accurately approximate a general orthonormal transform.

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 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

$$A_k = I + \Theta(i_k, j_k, \theta_k) \quad (26)$$

$$B_k = I + \Theta(i_k, j_k, \phi_k) \quad (27)$$

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

$$[\Theta]_{ij} = \begin{cases} \cos(\theta) - 1, & \text{if } i = j = m \text{ or } i = j = n \\ \sin(\theta), & \text{if } i = m \text{ and } j = n \\ -\sin(\theta), & \text{if } i = n \text{ and } j = m \\ 0, & \text{otherwise.} \end{cases} \quad (28)$$

Given the form of (26) and (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 (28). However, we can do better than this. In Appendix B, we show that the SMT of (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 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 $K = (M/2) \log_2 M$ 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.

²Note also that (25) represents the singular value decomposition of T_k .

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 nontraditional or data dependent orthonormal transforms, such as the KL transform.

A. Cost Function for SMT Design

In order to accurately approximate T_{opt} by the SMT transform of (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 (19) and (20) as

$$\check{H} = W^t H T^{-1} \Lambda^{1/2} \quad (29)$$

$$\check{y} = \Lambda^{-1/2} T y \quad (30)$$

where $T = \prod_{k=K-1}^0 T_k$ is the SMT transform of (24), and $\Lambda = \text{diag}(T R_y T^t)$. First notice, that Λ is defined so that the variance of each component of \check{y} is 1. Second notice, that when $T = T_{\text{opt}}$, then (29) and (30) reduce to (19) and (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 $T \neq T_{\text{opt}}$ 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 (29) and (30), the covariance matrices for \check{y} and \check{H} are given by

$$R_{\check{y}} = E[\check{y} \check{y}^t] = \Lambda^{-1/2} T R_y T^t \Lambda^{-1/2} \quad (31)$$

$$R_{\check{H}} = \frac{1}{N} \check{H}^t \check{H} = \Lambda^{1/2} (T^{-1})^t R_H T^{-1} \Lambda^{1/2} \quad (32)$$

where $R_H = H^t H / N$. If T is a good approximation to T_{opt} , then \check{y} will be approximately white, and by Corollary 1 we have that $E[|\delta \hat{x}|^2 | \delta \check{H}] \approx \|\delta \check{H}\|^2$, 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 $E[\|\delta \check{H}\|^2]$. To do this, we will derive a simplified expression for the distortion. In information theory, we know that if $X_i \sim N(0, \sigma_i^2)$, $i = 0, 1, 2, \dots, M-1$ are independent Gaussian random variables, then the rate and distortion functions for encoding the X_i 's is given by [38]

$$R(\lambda) = \sum_{i=0}^{M-1} \frac{1}{2} \max \left\{ 0, \log_2 \left(\frac{\sigma_i^2}{\lambda} \right) \right\} \quad (33)$$

$$D(\lambda) = \sum_{i=0}^{M-1} \min \{ \sigma_i^2, \lambda \} \quad (34)$$

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

$$R(\lambda) = N \sum_{i=0}^{M-1} \frac{1}{2} \max \left\{ 0, \log_2 \left(\frac{R_{\check{H}ii}}{\lambda} \right) \right\} \quad (35)$$

$$D(\lambda) = N \sum_{i=0}^{M-1} \min \{ R_{\check{H}ii}, \lambda \} \quad (36)$$

where $D(\lambda) = E[\|\delta \check{H}\|^2]$. If the distortion is sufficiently low so that $\lambda < \min(R_{\check{H}ii})$, then (36) reduces to $D(\lambda) = NM\lambda$. In this case, the rate in (35) can be expressed as the following function of distortion:

$$R(D) = \frac{N}{2} \left\{ \log_2 |\text{diag}(R_{\check{H}})| - M \log_2 \left(\frac{D}{NM} \right) \right\}. \quad (37)$$

Therefore, minimization of $|\text{diag}(R_{\check{H}})|$ 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

$$C(T, R_H, R_y) = |\text{diag}(R_{\check{H}})|. \quad (38)$$

Substituting in the expression for $R_{\check{H}}$ from (32) and the definition $\Lambda = \text{diag}(T R_y T^t)$ into (38) yields

$$C(T, R_H, R_y) = |\text{diag} \left((T^{-1})^t R_H T^{-1} \right)| \times |\text{diag}(T R_y T^t)|. \quad (39)$$

The cost function of (39) is also justified by the fact that the exact transform T_{opt} of (21) achieves its global minimum (see Appendix C). Our goal is then to find a sparse matrix transform T that minimizes the cost function $C(T, R_H, R_y)$.

B. SMT Design Using Greedy Minimization

In this subsection, we show how the SMT may be designed through greedy minimization of the cost function of (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

$$T^* = \arg \min_{T = \prod_{k=K-1}^0 T_k} \left\{ |\text{diag} \left((T^{-1})^t R_H T^{-1} \right)| \times |\text{diag}(T R_y T^t)| \right\} \quad (40)$$

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:

$$T_k^* \leftarrow \arg \min_{T_k} \left\{ |\text{diag} \left((T_k^{-1})^t R_H T_k^{-1} \right)| \times |\text{diag}(T_k R_y T_k^t)| \right\} \quad (41)$$

$$R_H \leftarrow (T_k^{*-1})^t R_H T_k^{*-1} \quad (42)$$

$$R_y \leftarrow T_k^* R_y (T_k^*)^t \quad (43)$$

where “ \leftarrow ” indicates assignment of a value in pseudocode. For a specified coordinate pair (i, j) , the cost function in (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 (41) for this case and also the ratio of the minimized cost function to its original value which is given by

$$\left(1 - \frac{R_{yij}^2}{R_{yii}R_{yjj}}\right) \left(1 - \frac{R_{Hij}^2}{R_{Hii}R_{Hjj}}\right) \quad (44)$$

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 (41) most among all possible pairs. The coordinate pair with the greatest cost reduction is then

$$(i_k, j_k) \leftarrow \arg \min_{(i,j)} \left\{ \left(1 - \frac{R_{yij}^2}{R_{yii}R_{yjj}}\right) \left(1 - \frac{R_{Hij}^2}{R_{Hii}R_{Hjj}}\right) \right\}. \quad (45)$$

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. 5. Then as shown in Appendix D, A_k is given by

$$A_k = I + \Theta(i_k, j_k, \theta_k), \quad \text{where } \theta_k = \frac{\pi}{4} \quad (46)$$

Λ_k is given by

$$[\Lambda_k]_{ij} = \begin{cases} \frac{1}{\sqrt{1+R_{yik}j_k}}, & \text{if } i = j = i_k \\ \frac{1}{\sqrt{1-R_{yik}j_k}}, & \text{if } i = j = j_k \\ 1, & \text{if } i = j \neq i_k \text{ and } i = j \neq j_k \\ 0, & \text{if } i \neq j; \end{cases} \quad (47)$$

and

$$B_k = I + \Theta(i_k, j_k, \phi_k) \quad (48)$$

where

$$\phi_k = \frac{1}{2} \operatorname{atan} \left((R_{Hj_kj_k} - R_{Hi_ki_k}) \sqrt{1 - R_{yik}^2} \right. \\ \left. (R_{Hi_ki_k} + R_{Hj_kj_k}) R_{yikj_k} + 2R_{Hi_kj_k} \right) \quad (49)$$

$\Lambda_y \leftarrow \operatorname{diag}(R_y)$
 $R \leftarrow \Lambda_y^{-1/2} R_y \Lambda_y^{-1/2}$
 $C \leftarrow \Lambda_y^{1/2} R_H \Lambda_y^{1/2}$
 For $k = 0 : K - 1$ {

$$(i_k, j_k) \leftarrow \arg \min_{i < j} \left\{ (1 - R_{ij}^2) \cdot \left(1 - \frac{C_{ij}^2}{C_{ii}C_{jj}}\right) \right\}$$

$$\theta_k \leftarrow \pi/4$$

$$A_k \leftarrow I + \Theta(i_k, j_k, \theta_k)$$

$$\Lambda_k \leftarrow I$$

$$[\Lambda_k]_{i_k, i_k} \leftarrow 1/\sqrt{1 + R_{i_k j_k}}$$

$$[\Lambda_k]_{j_k, j_k} \leftarrow 1/\sqrt{1 - R_{i_k j_k}}$$

$$\phi_k \leftarrow \frac{1}{2} \operatorname{atan} \left((C_{j_k j_k} - C_{i_k i_k}) \sqrt{1 - R_{i_k j_k}^2} \right. \\ \left. (C_{j_k j_k} + C_{i_k i_k}) R_{i_k j_k} + 2C_{i_k j_k} \right)$$

$$B_k \leftarrow I + \Theta(i_k, j_k, \phi_k)$$

$$T_k \leftarrow B_k \Lambda_k A_k$$

$$R \leftarrow T_k R T_k^t$$

$$C \leftarrow (T_k^{-1})^t C T_k^{-1}$$

}

$$T \leftarrow (\prod_{k=K-1}^0 T_k) \Lambda_y^{-\frac{1}{2}}$$

Fig. 5. Pseudo-code implementation of the greedy algorithm used for the SMT design.

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

$$T^* = \prod_{k=K-1}^0 B_k \Lambda_k A_k = \prod_{k=K-1}^0 T_k^*. \quad (50)$$

Fig. 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 two coordinates, i_k and j_k , that minimize the criteria of (45) with each iteration. However, this operation 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 .

C. 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 a constant multiple of M in many applications, so the resulting

³Here, we use $\operatorname{atan}(y, x) = \operatorname{atan}(y/x)$ 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.

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].

V. 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.

A. ODT Example ($M \ll N$)

1) *Description of Experiment:* Fig. 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 nine 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 3-D box with a size of $16 \times 16 \times 6$ cm³. 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 optical 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

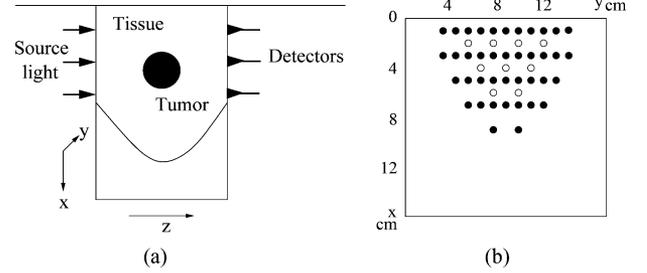


Fig. 6. Measurement geometry for optical breast imaging. (a) Imaging geometry. (b) Source-detector probe configuration. The open circles indicate the source fiber locations and the solid circles indicate the detector fiber location. Source fibers and detector fibers are connected to the left and right plates, respectively, and are on 1-cm grid (adapted from [45]).

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 $R_y = AE[xx^t]A^t = AA^t$, 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 III. 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 \tilde{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.

2) *Discussion of Experimental Results:* Fig. 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

$$\text{NRMSE} = \sqrt{\frac{\| [H]y - Hy \|^2}{\| Hy \|^2}}. \quad (51)$$

Fig. 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. 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 I compares the computational complexity of the three methods: iterative MAP using conjugate gradient; non-iterative MAP with no compression; and noniterative MAP

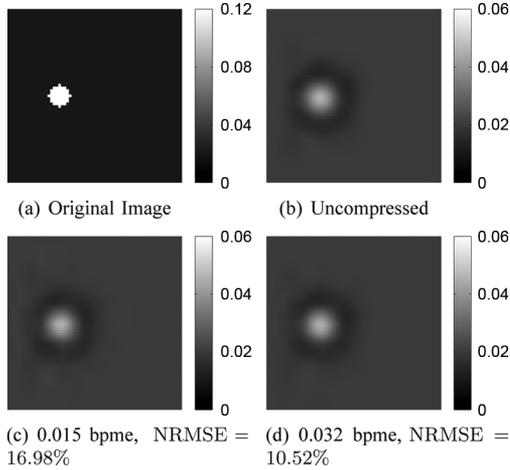


Fig. 7. Reconstructed images of $\mu_a(r)$ at $z = 3$ cm using the compressed H matrix based on KL transform (used both for data whitening and matrix decorrelation). The compression ratios in (c) and (d) are 4267:1 and 1982:1, respectively. Here “bpme” stands for bits per matrix entry.

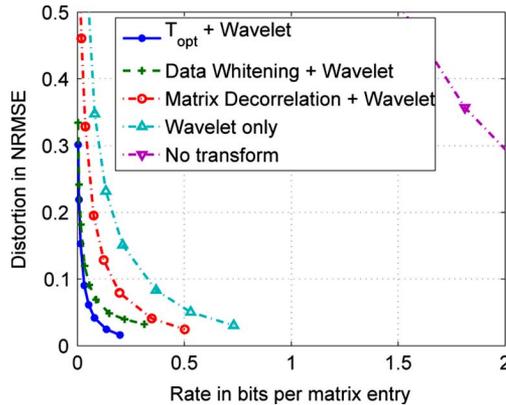


Fig. 8. Distortion versus rate for compression using KL transforms for data whitening and matrix decorrelation for the ODT example. Notice here simply whitening the data yields close distortion-rate performance to the theoretically optimal KL transforms. The performance drops significantly with the other three methods that do not perform data whitening.

with the KLT compression. The noniterative 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 $[\tilde{H}]$ (0.4 Mbyte) and the storage of the required transform T_{opt} (4.0 Mbyte). From the table we can see that both the online computation time and storage are dramatically reduced using the compressed inverse matrix.

B. FODT Example (M is Close to N)

1) *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 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. 9, where a $6 \text{ cm} \times 6 \text{ cm}$ 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. 9, 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 $\mu_{a_x} = \mu_{a_m} = 0.02 \text{ cm}^{-1}$ and $D_x = D_m = 0.03 \text{ cm}$, where the subscripts x and m represent the wavelengths λ_x and λ_m , respectively, and the bulk fluorescence yield was set to $\eta\mu_{a_f} = 0 \text{ cm}^{-1}$. 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 $\mu_{a_x} = 0.12 \text{ cm}^{-1}$, $\mu_{a_m} = 0.02 \text{ cm}^{-1}$, $D_x = D_m = 0.03 \text{ cm}^{-1}$, and $\eta\mu_{a_f} = 0.05 \text{ cm}^{-1}$. The size of the imaging domain is $8 \times 8 \times 4 \text{ cm}^3$, 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 $\mu_{a_x} = \mu_{a_m} = 0.02 \text{ cm}^{-1}$ and $D_x = D_m = 0.03 \text{ cm}$ 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 \tilde{H} was quantized and encoded using a run-length coder (see Appendix E for details).

2) *Discussion of Experimental Results:* Fig. 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

TABLE I

COMPARISON OF ONLINE AND OFFLINE 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 NONITERATIVE MAP RECONSTRUCTION REQUIRES MUCH LOWER ONLINE COMPUTATION AND MEMORY THAN THE ITERATIVE RECONSTRUCTION METHOD. HOWEVER, IT REQUIRES GREATER OFFLINE 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	$\frac{NM}{c} + M^2$	$0.4 + 4.0$ $([\tilde{H}] + T_{opt})$

	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

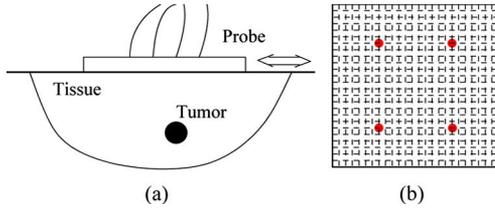


Fig. 9. Measurement geometry for an FODT example. (a) A graphic depiction of the imaging geometry. (b) An illustration of the source-detector probe where the solid circles indicate the locations of sources and the rectangular grid represents the CCD sensor array.

transform are given in Fig. 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. Fig. 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 II gives a detailed comparison of noniterative 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

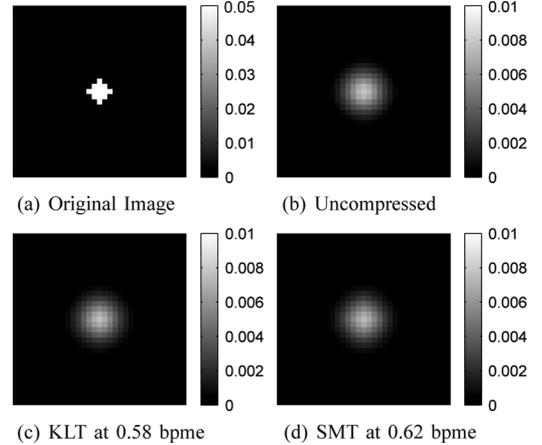


Fig. 10. Reconstructed images of $\eta\mu_{a_f}(r)$ at the depth of 2 cm using different compression methods. The compression ratios in (c) and (d) are 110:1 and 103:1, and the NRMSE's are 9.96% and 10.24%, respectively.

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 $[\tilde{H}]$ and the storage of the required transform T_{opt} or T as shown explicitly in the table. Notice that the SMT reduces the online computation by over a factor of 2 and reduces online 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.

C. Discussion

From the numerical examples, we see that noniterative MAP reconstruction can dramatically reduce the computation and memory usage for online reconstruction. However, this dramatic reduction requires the offline precomputation and

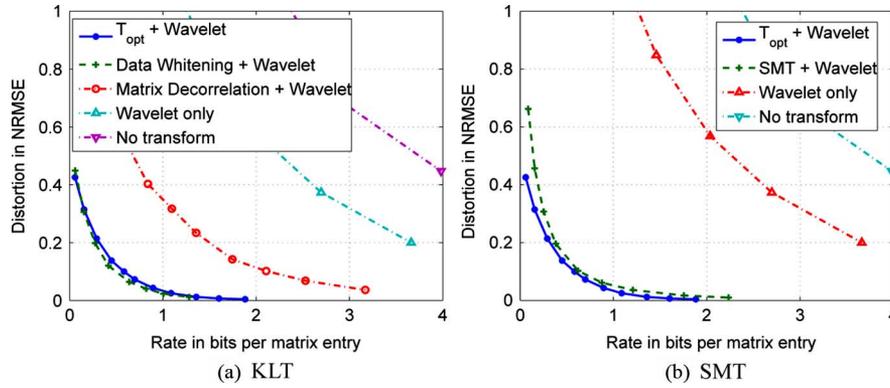


Fig. 11. Distortion versus rate for the FODT example. (a) Distortion versus rate for compression using the KL transforms for data whitening and matrix decorrelation. (b) Distortion versus rate for compression using the sparse matrix transform (SMT). $M \log_2(M)$ SMT butterflies were used to whiten the 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 II

COMPARISON OF ONLINE AND OFFLINE 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 $K = M \log_2(M) = 28220$, 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 NONITERATIVE MAP RECONSTRUCTION REQUIRES MUCH LESS ONLINE COMPUTATION AND MEMORY THAN THE ITERATIVE RECONSTRUCTION METHOD. HOWEVER, IT REQUIRES GREATER OFFLINE 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})	$\frac{NM}{c_1} + M^2$	$3.2 + 47.7$ ($[H] + T_{opt}$)
Non-Iterative MAP with SMT Compression	$\frac{NM}{c_2} + N + K$	0.03	$\frac{NM}{c_2} + K$	$3.5 + 0.5$ ($[H] + T$)

	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

encoding of the inverse transform. In our experiments, computation of the inverse matrix dominated offline 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 noniterative 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 com-

pared 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].

VI. CONCLUSION

In this paper, we presented a noniterative 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.

APPENDIX 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

$$h_i = \arg \min_x \|e_i - Ax\|_{\Lambda}^2 + x^t Sx.$$

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

$$\tilde{h}_i = \arg \min_x \|\tilde{e}_i - Ax\|_{\Lambda}^2 + x^t Sx$$

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

APPENDIX 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

$$T_k \cdot S_k = \tilde{S}_k \cdot \tilde{T}_k. \quad (52)$$

Without loss of generality, (52) can be easily verified that in the 2-D case. First assume either $s_1 t_{11}$ or $s_2 t_{22}$ is not zero, then we have

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \cdot \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} = \begin{bmatrix} \tilde{s}_1 & 0 \\ 0 & \tilde{s}_2 \end{bmatrix} \cdot \begin{bmatrix} 1 & b \\ a & 1 \end{bmatrix} \quad (53)$$

where

$$\begin{cases} \tilde{s}_1 = s_1 t_{11} \\ \tilde{s}_2 = s_2 t_{22} \end{cases} \quad \text{and} \quad \begin{cases} a = \frac{s_1 t_{21}}{s_2 t_{22}} \\ b = \frac{s_2 t_{12}}{s_1 t_{11}}. \end{cases} \quad (54)$$

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

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \cdot \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & s_2 t_{22} \end{bmatrix} \cdot \begin{bmatrix} 0 & s_2 t_{12} \\ \frac{s_1 t_{21}}{s_2 t_{22}} & 1 \end{bmatrix}. \quad (55)$$

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

$$\begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \cdot \begin{bmatrix} s_1 & 0 \\ 0 & s_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & s_2 t_{12} \\ s_1 t_{21} & 0 \end{bmatrix}. \quad (56)$$

Starting from $S_0 = I$ and iterating (52) with $S_{k+1} = \tilde{S}_k$, we can obtain

$$\tilde{y} = \left(\prod_{k=K-1}^0 T_k \right) y = \tilde{S}_{K-1} \left(\prod_{k=K-1}^0 \tilde{T}_k \right) y. \quad (57)$$

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

APPENDIX C OPTIMALITY OF T_{opt} FOR THE SMT COST FUNCTION

Here, we show that the exact transform T_{opt} in (21) is the solution to the minimization of (40). First we prove some properties of symmetric, positive definite matrices, e.g., any covariance matrix.

Theorem 2: If $R \in \mathbb{R}^{n \times n}$ is a symmetric, positive definite matrix, then we have

$$|\text{diag}(R)| \geq |R|. \quad (58)$$

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

$$R = GG^t \quad (59)$$

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

$$|\text{diag}(R)| \geq \prod_{i=1}^n G_{ii}^2 = |R|. \quad (60)$$

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

Now we show that $T_{\text{opt}} = \Phi^t \Lambda_y^{-1/2} E^t$ minimizes the cost function (40) among all nonsingular matrices, i.e.,

$$T_{\text{opt}} = \arg \min_T |\text{diag}((T^{-1})^t R_H T^{-1})| \cdot |\text{diag}(T R_y T^t)|. \quad (61)$$

Proof: Since the covariance matrices R_H and R_y are symmetric and definite positive, $(T^{-1})^t R_H T^{-1}$ and $T R_y T^t$ are also symmetric and definite positive for any nonsingular matrix $T \in \mathbb{R}^{n \times n}$. Therefore, from Theorem 2 we have

$$\begin{aligned} & |\text{diag}((T^{-1})^t R_H T^{-1})| \cdot |\text{diag}(T R_y T^t)| \\ & \geq |(T^{-1})^t R_H T^{-1}| \cdot |T R_y T^t| = |R_H| \cdot |R_y|. \end{aligned} \quad (62)$$

The equality in (62) 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 diagonalized by T . Now, let $T_{\text{opt}} = \Phi^t \Lambda_y^{-1/2} E^t$. This leads to

$$T_{\text{opt}} R_y T_{\text{opt}}^t = I \quad (63)$$

$$(T_{\text{opt}}^{-1})^t R_H T_{\text{opt}}^{-1} = \Lambda_{\tilde{H}} \quad (64)$$

where $\Lambda_{\tilde{H}}$ is defined in (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 (40), we can also explicitly show

$$\begin{aligned} & |\text{diag}((T_{\text{opt}}^{-1})^t R_H T_{\text{opt}}^{-1})| \cdot |\text{diag}(T_{\text{opt}} R_y T_{\text{opt}}^t)| \\ & = |R_H| \cdot |R_y|. \end{aligned} \quad (65)$$

■

APPENDIX D OPTIMAL GIVENS ROTATION

In this appendix, we find the solution to the optimization problem of (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 (41) if and only if it simultaneously diagonalizes both R_y and R_H . To do this, let $T^* = B \Lambda A$, and let

$$R_y = \begin{bmatrix} \sigma_{y11}^2 & \sigma_{y12} \\ \sigma_{y21} & \sigma_{y22}^2 \end{bmatrix} \text{ and } R_H = \begin{bmatrix} \sigma_{h11}^2 & \sigma_{h12} \\ \sigma_{h21} & \sigma_{h22}^2 \end{bmatrix}. \quad (66)$$

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

$$A R_y A^t = \begin{bmatrix} \sigma_{y11}^{\prime 2} & 0 \\ 0 & \sigma_{y22}^{\prime 2} \end{bmatrix} \quad (67)$$

where

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

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

$$\sigma_{y11}^{\prime 2} = 1 + \sigma_{y12} \quad (68)$$

$$\sigma_{y22}^{\prime 2} = 1 - \sigma_{y12}. \quad (69)$$

If we define $\Lambda = \begin{bmatrix} 1/\sigma_{y11}^{\prime} & 0 \\ 0 & 1/\sigma_{y22}^{\prime} \end{bmatrix}$, then this leads to

$$R_{y'} = \Lambda A R_y A^t \Lambda = I \quad (70)$$

$$R_{H'} = \Lambda^{-1} A R_H A^t \Lambda^{-1} = \begin{bmatrix} \sigma_{h11}^{\prime 2} & \sigma_{h12}^{\prime} \\ \sigma_{h21}^{\prime} & \sigma_{h22}^{\prime 2} \end{bmatrix} \quad (71)$$

where

$$\begin{aligned} \sigma_{h11}^{\prime 2} &= \sigma_{y11}^{\prime 2} \left[\frac{1 + \cos 2\theta}{2} \sigma_{h11}^2 + \sigma_{h12} \sin 2\theta \right. \\ & \quad \left. + \frac{1 - \cos 2\theta}{2} \sigma_{h22}^2 \right] \\ \sigma_{h22}^{\prime 2} &= \sigma_{y22}^{\prime 2} \left[\frac{1 - \cos 2\theta}{2} \sigma_{h11}^2 - \sigma_{h12} \sin 2\theta \right. \\ & \quad \left. + \frac{1 + \cos 2\theta}{2} \sigma_{h22}^2 \right] \\ \sigma_{h12}^{\prime} &= \sigma_{y11}^{\prime} \sigma_{y22}^{\prime} \left[\frac{1}{2} (\sigma_{h22}^2 - \sigma_{h11}^2) \sin 2\theta \right. \\ & \quad \left. + \sigma_{h12} \cos 2\theta \right] \end{aligned}$$

and $\sigma_{h21}^{\prime} = \sigma_{h12}^{\prime}$. Then, a Givens rotation $B = I + \Theta(1, 2, \phi)$ with $\phi = (1/2)\text{atan}(2\sigma_{h12}^{\prime}, \sigma_{h11}^{\prime 2} - \sigma_{h22}^{\prime 2})$ makes $B R_{H'} B^t$ diagonal while maintaining $B R_{y'} B^t = I$. For the special case that $\sigma_{y11}^2 = \sigma_{y22}^2 = 1$, the simplified expression for B is given in (48).

So we see that both $T^* R_y T^{*t}$ and $(T^{*-1})^t R_H T^{*-1}$ are diagonal with $T^* = B \Lambda A$ given above. Hence, T^* must minimize the cost function of (41). Based on (65), we know that the ratio of the cost function before and after the transform of T^* is given as

$$\begin{aligned} \frac{C(T^*, R_H, R_y)}{C(I, R_H, R_y)} &= \frac{|R_H| \cdot |R_y|}{|\text{diag}(R_H)| \cdot |\text{diag}(R_y)|} \\ &= \left(1 - \frac{\sigma_{y12}^2}{\sigma_{y11}^2 \sigma_{y22}^2}\right) \\ &\quad \times \left(1 - \frac{\sigma_{h12}^2}{\sigma_{h11}^2 \sigma_{h22}^2}\right). \end{aligned} \quad (72)$$

APPENDIX E

Here, we give the details of the run-length encoder [30] used in numerical experiments. After quantization of \tilde{H} , the run-length encoder codes the first voxel position of each nonzero run along every column, the run length and the nonzero 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 whether it belongs to the same column as the previous run. An 8-bit or 16-bit integer is used to store every quantized nonzero 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 nonzero 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 nonzero off-diagonal entries.

REFERENCES

- [1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1991.
- [2] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
- [3] E. E. Fenimore and T. M. Cannon, "Coded aperture imaging with uniformly redundant arrays," *Appl. Opt.*, 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," presented at the IEEE Information Theory Workshop on Detection, Estimation, Classification, and Imaging, Feb. 1999.
- [5] E. J. Candes, "Compressive sampling," presented at the Int. Congr. Mathematicians, Madrid, Spain, Aug. 2006.
- [6] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
- [7] C. F. Van Loan, *Computational Frameworks for the Fast Fourier Transform*. Philadelphia, PA: SIAM, 1992.
- [8] P. A. Regalia and S. K. Mitra, "Kronecker products, unitary matrices and signal processing applications," *SIAM Rev.*, vol. 31, no. 4, pp. 586–613, Dec. 1989.
- [9] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA: Academic, 1998.
- [10] T. Hebert and R. Leahy, "A generalized EM algorithm for 3-D Bayesian reconstruction from Poisson data using Gibbs priors," *IEEE Trans. Med. Imag.*, vol. 8, no. 2, pp. 194–202, Jun. 1989.
- [11] K. Sauer and C. A. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 534–548, Feb. 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," *Med. Phys.*, vol. 34, no. 11, pp. 4526–4544, 2007.
- [13] S. Geman and D. E. McClure, "Statistical methods for tomographic image reconstruction," *Bull. Int. Statist. Inst.*, 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. Med. Imag.*, vol. 6, no. 3, pp. 185–192, Mar. 1987.
- [15] P. J. Green, "Bayesian reconstructions from emission tomography data using a modified EM algorithm," *IEEE Trans. Med. Imag.*, vol. 9, no. 1, pp. 84–93, Jan. 1990.
- [16] G. Cao, C. A. Bouman, and K. J. Webb, "Fast and efficient stored matrix techniques for optical tomography," in *Proc. 40th Asilomar Conf. Signals, Systems, and Computers*, Oct. 2006.
- [17] G. Cao, C. A. Bouman, and K. J. Webb, "Fast reconstruction algorithms for optical tomography using sparse matrix representations," presented at the IEEE Int. Symp. Biomedical Imaging, Apr. 2007.
- [18] W. Givens, "Computation of plane unitary rotations transforming a general matrix to triangular form," *J. SIAM*, vol. 6, no. 1, pp. 26–50, Mar. 1958.
- [19] H. Rutishauser, "The jacobi method for real symmetric matrices," *Numer. Math.*, vol. 9, pp. 1–10, 1966.
- [20] P. A. Regalia, "An adaptive unit norm filter with applications to signal analysis and Karhunen–Loeve transformations," *IEEE Trans. Circuits Syst.*, vol. 37, no. 5, pp. 646–649, May 1990.
- [21] P. Comon, "Independent component analysis: A new concept?," *Signal Process.*, 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 of nonlinear systems," *IEEE Signal Process. Lett.*, vol. 4, no. 6, pp. 182–183, Jun. 1997.
- [24] A. Hyvarinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Netw.*, vol. 13, pp. 411–430, May-Jun. 2000.
- [25] A. Soman and P. Vaidyanathan, "Paraunitary filter banks and wavelet packets," in *Proc. IEEE ICASSP*, Mar. 1992, vol. 4, pp. 397–400.
- [26] A. B. Lee, B. Nadler, and L. Wasserman, "Treelets—an adaptive multiscale basis for sparse unordered data," *Ann. Appl. Statist.*, 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*. Cambridge, MA: MIT Press, 2008, pp. 225–232.
- [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, Feb. 2005.
- [29] A. Tikhonov and V. Arsenin, *Solutions of Ill-Posed Problems*. New York: Winston & Sons, 1977.
- [30] A. N. Netravali and B. G. Haskell, *Digital Pictures: Representation, Compression, and Standards*. New York: Plenum, 1995.
- [31] G. Golub and C. V. Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1996.
- [32] D. Tretter and C. A. Bouman, "Optimal transforms for multispectral and multilayer image coding," *IEEE Trans. Image Process.*, vol. 4, no. 3, pp. 296–308, Mar. 1995.
- [33] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, Apr. 1992.
- [34] P. Dragotti, G. Poggi, and A. Ragozini, "Compression of multispectral images by three-dimensional SPIHT algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 1, pp. 416–428, Jan. 2000.
- [35] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.
- [36] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, Apr. 1965.
- [37] D. K. Hoffman, R. C. Raffanetti, and K. Ruedenberg, "Generalization of euler angles to n-dimensional orthogonal matrices," *J. Math. Phys.*, vol. 13, no. 4, pp. 528–533, 1972.
- [38] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [39] J.-F. Cardoso, "High-order contrasts for independent component analysis," *Neural Comput.*, vol. 11, no. 1, pp. 157–192, 1999.
- [40] J. Eriksson and V. Koivunen, "Characteristic-function-based independent component analysis," *Signal Process.*, 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 Process. Mag.*, vol. 18, no. 6, pp. 57–75, Nov. 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," *Appl. Opt.*, vol. 42, no. 16, pp. 3081–3094, Jun. 2003.
- [43] M. S. Patterson, J. D. Moulton, B. C. Wilson, K. W. Berndtand, and J. R. Lakowicz, "Frequency-domain reflectance for the determination of the scattering and absorption properties of tissue," *Appl. Opt.*, 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," *Phys. Rev. Lett.*, 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, no. 2, pp. 024033, 1–9, Mar./Apr. 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," *Appl. Opt.*, 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. Opt. Soc. Amer. A*, vol. 16, no. 10, pp. 2400–2412, Oct. 1999.
- [48] C. A. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Trans. Image Process.*, vol. 2, no. 3, pp. 296–310, Jul. 1993.
- [49] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, 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. Med. Imag.*, vol. 24, no. 2, pp. 137–154, Feb. 2005.
- [51] J. Wei, G. Cao, C. Bouman, and J. Allebach, "Fast space-varying convolution and its application in stray light reduction," presented at the SPIE/IS&T Conf. Computational Imaging VII, Jan. 2009.



Guangzhi Cao (S'07) received the B.S. and M.S. degrees in electrical engineering from Zhejiang University, China, in 2002 and 2004, respectively. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

His current research interests include statistical signal and image processing, inverse problems, machine learning, and computer vision.



Charles A. Bouman (S'86–M'89–SM'97–F'01) received the B.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1981, the M.S. degree from the University of California, Berkeley, in 1982, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1989.

From 1982 to 1985, he was a full staff member at the Massachusetts Institute of Technology Lincoln Laboratory. In 1989, he joined the faculty of Purdue University, West Lafayette, IN, where he is a Professor with a primary appointment in the School of Electrical and Computer Engineering and a secondary appointment in the School of Biomedical Engineering. Currently, he is Co-Director of Purdue's Magnetic Resonance Imaging Facility located in Purdue's Research Park. His research focuses on the use of statistical image models, multiscale techniques, and fast algorithms in applications including tomographic reconstruction, medical imaging, and document rendering and acquisition.

Prof. Bouman is a Fellow of the IEEE, a Fellow of the American Institute for Medical and Biological Engineering (AIMBE), a Fellow of the society for Imaging Science and Technology (IS&T), a Fellow of the SPIE professional society, a recipient of IS&T's Raymond C. Bowman Award for outstanding contributions to digital imaging education and research, and a University Faculty Scholar of Purdue University. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING and a member of the IEEE Biomedical Image and Signal Processing Technical Committee. He has been a member of the Steering Committee for the IEEE TRANSACTIONS ON MEDICAL IMAGING and an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has also been Co-Chair of the 2006 SPIE/IS&T Symposium on Electronic Imaging, Co-Chair of the SPIE/IS&T Conferences on Visual Communications and Image Processing 2000 (VCIP), a Vice President of Publications and a member of the Board of Directors for the IS&T Society, and he is the founder and Co-Chair of the SPIE/IS&T Conference on Computational Imaging.



Kevin J. Webb (S'81–M'84–SM'98–F'04) received B.Eng. and M.Eng. degrees from the Royal Melbourne Institute of Technology, Australia, in 1978 and 1983, respectively, the M.S.E.E. from the University of California, Santa Barbara, in 1981, and the Ph.D. degree from the University of Illinois, Urbana, in 1984.

He is a Professor in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. Prior to joining Purdue in 1990, he was on the faculty of the University of Maryland, College Park. During the calendar year 2003, he was a Visiting Professor at the Massachusetts Institute of Technology, Cambridge. During 2006/2007, he was a Visiting Professor at the Royal Institute of Technology, Stockholm, Sweden, and at the University of Canterbury, Christchurch, U.K.