Learning Sparsifying Filter Banks

Luke Pfister and Yoram Bresler

Dept. of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

ABSTRACT

Recent years have numerous algorithms to learn a sparse synthesis or analysis model from data. Recently, a generalized analysis model called the ’transform model’ has been proposed. Data following the transform model is approximately sparsified when acted on by a linear operator called a sparsifying transform. While existing transform learning algorithms can learn a transform for any vectorized data, they are most often used to learn a model for overlapping image patches. However, these approaches do not exploit the redundant nature of this data and scale poorly with the dimensionality of the data and size of patches.

We propose a new sparsifying transform learning framework where the transform acts on entire images rather than on patches. We illustrate the connection between existing patch-based transform learning approaches and the theory of block transforms, then develop a new transform learning framework where the transforms have the structure of an undecimated filter bank with short filters. Unlike previous work on transform learning, the filter length can be chosen independently of the number of filter bank channels.

We apply our framework to accelerating magnetic resonance imaging. We simultaneously learn a sparsifying filter bank while reconstructing an image from undersampled Fourier measurements. Numerical experiments show our new model yields higher quality images than previous patch based sparsifying transform approaches.

Keywords: Sparisfying transform learning, Sparse representations, Filter banks, MR reconstruction

1. INTRODUCTION

Problems in fields ranging from statistical inference to medical imaging can be posed as the recovery of high-quality data from incomplete or corrupted linear measurements. Formally, the goal is to recover the unknown signal $x \in \mathbb{R}^N$ from linear measurements $y \in \mathbb{R}^M$, given by

$$y = Ax + e, \tag{1}$$

where $e \in \mathbb{R}^p$ represents measurement error and the matrix $A \in \mathbb{R}^{M \times N}$ models the data acquisition process.

Often, the data acquisition process leads to A that are either poorly conditioned or underdetermined, as is the case when we have access to fewer measurements than the ambient signal dimension ($M < N$). In this case, the inverse problem can be solved only by utilizing additional information about the signal of interest to regularize the inverse problem. This can be accomplished by solving the variational problem

$$\arg\min_x \frac{1}{2}\|y - Ax\|_2^2 + \lambda\psi(x), \tag{2}$$

where the regularization functional $\psi : \mathbb{R}^N \times \mathbb{R}$ penalizes solutions that do not satisfy the prescribed signal model.

Regularization based on sparse representations has proven to be especially effective for a wide variety of inverse problems. An n-dimensional signal is said to follows a sparse signal model if it can be represented using far fewer than n nonzero entries. Traditionally, sparse representations have been carefully designed to provide optimal properties on a particular mathematical class of signals, *e.g.,* wavelets enjoy optimal coefficient decay properties for piecewise constant signals. However, it is difficult to develop an analytic sparse representation that is fine-tuned for a practical class of data, such as medical images or speech. These difficulties are further exacerbated for high dimensional data such as social network data, 4D imaging, or gene expression data.

These considerations have led to a variety of algorithms to *learn* a sparse signal model directly from data. These algorithms attempt to train a sparse model by minimizing a metric of sparsity over a set of representative training data. For many years, attention was focused primarily on adaptive synthesis sparsity models, wherein the data is synthesized as the linear combination of a few generating signals.

More recently, attention has shifted to learning signal models wherein the data becomes sparse after being acted or *analyzed* on a by a linear operator. Many of these approaches follow the *co-sparse analysis model*, which is understood to require exact sparsity of the analyzed signal.

However, other work has focused on a generalization of the analysis model using a particular notion of compressibility.[1–4] A signal $x \in \mathbb{R}^N$ is said to obey the *transform sparsity model* if there is a matrix $W \in \mathbb{R}^{M \times N}$ such that $Wx = z + \eta$, where z is sparse and $\|\eta\|_2$ is small. The matrix W is called a *sparsifying transform*, and the vector z is referred to as an *transform sparse code*. Unlike the usual analysis model, the transform sparsity model allows for the sparse code z to lie outside of the range space of W.

The task of finding z for a particular W and x can be posed as either a constrained or penalized optimization problem. The constrained problem is stated as[1]

$$\min_x \frac{1}{2}\|Wx - z\|_2^2 \,\text{s. t.}\|z\|_0 \leq s, \tag{3}$$

where $\|z\|_0$ simply counts the number of nonzero entries in the vector z. The solution to (3) is given in closed form by retaining the s-largest entries of Wx and setting the rest to zero. The penalized variation is written[5]

$$\min_x \frac{1}{2}\|Wx - z\|_2^2 + \nu\|z\|_0, \tag{4}$$

and the solution can be found by taking $z_j = [Wx]_j$ whenever $|[Wx]_j| \geq \sqrt{\nu}$ and setting $z_j = 0$ otherwise. This operation is called *hard thresholding* and will be written as $z = \mathcal{T}_\nu(Wx)$. We note that the ℓ_0 norm can be replaced with a variety of other functionals, such as the ℓ_1 norm or the Huber function, and retain the closed form and efficient solutions to the penalized sparse coding problem.

Several methods have been developed to learn a sparsifying transform from data. These include methods to learn square transforms ($K = N$),[1] tall transforms ($K > N$),[3] and structured transforms.[4,6] These methods seek to find a W such that Wx is close in the ℓ_2 sense to some sparse vector z. Of course, with no additional constraints, this problem can easily be solved by taking W to be a matrix of all zeros! As a consequence, further constraints must be imposed on the learning problem.

Existing transform learning algorithms prohibit trivial solutions by requiring W to be left-invertible. This has the further benefit of providing easy approximation of x from its transform sparse codes through $W^\dagger \mathcal{T}_\nu(Wx)$. This property, reminiscent of transform coding with orthonormal matrices, motivates the name "transform sparsity".

The effectiveness of transform learning for inverse problems has been demonstrated on a variety of inverse problems, including image denoising, magnetic resonance imaging, and computed tomography. As is common with adaptive synthesis or co-sparse analysis models for spatiotemporal data, sparsifying transforms are often used to model small, possibly overlapping sub-blocks of data called *patches*. The resulting model is called a *patch-based* signal model and stands in contrast to an *image-based*

While patch-based models can be used for many types of data, such as images, audio, video, or high dimensional data, we restrict our attention to two-dimensional images, although the approach generalizes to any type of data that can be modeled by overlapping patches.

Patch-based models enjoy several desirable features. For instance, a single image can be decomposed of a large number of patches, providing enough training data to learn a model for that particular image. Further, patch-based models can have fewer parameters than a model for an entire image, leading to computational efficiency and lower risk of overfitting.

Using a patch-based model to regularize an inverse problem requires a method to link the individual patches to the desired image. A simple approach is to simply average the overlapping regions of each patch together to form a final estimate.[7–9] However, this technique ignores any correlation between neighboring patches.

A different technique is to learn a patch-based model assuming independence of the patches, but use a least squares approach to form the final image estimate.[5, 7, 10–13] Still, if we desire to reconstruct a complete image, it is natural to expect better results by using an image-based signal model.

Existing work in this direction is based on the *Field of Experts* model.[14, 15] These approaches use a Markov Random Field (MRF) with overlapping cliques of pixels to motivate image patches. A probablilistic model for an image is generated by taking the product of separate patch-level priors and normalizing. Each patch level prior is expressed as a potential function applied to the product of the vectorized image patch with a matrix, the coefficients of which are determined using maximum likelihood estimation. Thus these approaches are similar in structure, if not spirit, to patch-based cosparse or transform models.

In this work, we propose a framework to learn an image-based transform sparsity model. Unlike existing probabilistic MRF approaches, we propose a deterministic framework based on the connection between patch-based sparsifying transforms and multichannel filter banks.

In Section 2 we examine this connection and interpret existing transform learning algorithms using the language of polyphase filter banks. In Section 3, we use this connection to motivate a new method to learn image-based sparsifying transforms that are structured as non-subsampled filter banks. We refer to these as filter bank sparsifying transforms. In Section 5, we apply our new transforms to magnetic resonance image reconstruction from undersampled data. We demonstrate that our new filter bank sparsifying transforms outperform existing patch-based transform learning approaches.

Similar to what we propose, Cai *et. al*[16] developed an algorithm to learn an analysis operator that is implemented using a filter bank structure. However, they restricted their attention to learning critically sampled filter banks. We show in Section 2 that this is equivalent to learning a patch-based analysis operator.

2. PATCH-BASED SPARSIFYING TRANSFORMS AS FILTER BANKS

We focus on the case where our training data $X \in \mathbb{R}^{K^2 \times (N/s)^2}$ consists of vectorized $K \times K$ patches extracted from a single $N \times N$ image x, although it is straightforward to generalize to other forms of spatiotemporal data or multiple images. The integer $s \geq 1$ represents the distance between the same pixel location in consecutive patches, which we call the *patch stride.* We take the sparsifying transform to be a matrix of size $N_c \times K^2$. In this section, we show that the transformed patches, WX, can be viewed as an N_c channel filter bank applied to the image x, and that the properties of the filter bank are controlled by W and s. We consider two cases: nonoverlapping patches and maximally overlapping patches.

The extracted patches do not overlap whenever $s = K$. Multiplying non-overlapping blocks of an image by a fixed transform matrix can be interpreted as a *block transformation*,[17] and thus WX can implemented as a uniformly downsampled FIR filter bank. Here, the filter for the i-th channel w_i is related to the i-th row of W as $W_{i,:} = \text{vec}\,(w_i)$. The number of rows of W determine the number of channels in this filter bank and the downsampling factor is s. The shape of the filter corresponds to the shape of extracted patches. We will restrict our attention to square, $K \times K$ patches and thus square, $K \times K$ filters.

While the relationship between block transforms and multirate filter banks is well established, it lends a fresh perspective on patch-based analysis operators. Often, the product WX is thought of as applying W to a set of completely independent data vectors $\{x_j\}$, even when the patches are extracted from a single image. The filter bank perspective encourages us to consider WX as a single linear operator, which we denote $\mathcal{H}_W$, applied to the image x.

Recall that a filter bank $\mathcal{H}_W$ is said to be orthonormal if $\mathcal{H}_W^T \mathcal{H}_W = I$, and $\mathcal{H}_W$ is said to be a *perfect reconstruction* (PR) filter bank if $\mathcal{H}_W$ is left invertible. Whenever $s = K$ and the filters w_i are square $(K \times K)$, the matrix W corresponds to the polyphase matrix of the filter bank $\mathcal{H}_W$, and thus the orthonormality and perfect reconstruction properties are determined entirely by W. Indeed, when $W^T W = I$, $\mathcal{H}_W$ is an orthonormal filter

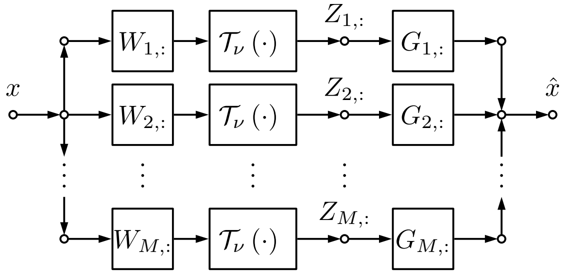


Figure 1. Block diagram of a sparsifying transform as a filter bank with an elementwise nonlinearity. The channels of the analysis filter bank $\mathcal{H}_W$ are given by the rows of W. The matrix G represents the channel coefficients for a left inverse of $\mathcal{H}_W$.

bank, and the downsampled filter bank $\mathcal{H}_W$ is perfect reconstruction if and only if its polyphase matrix W is left invertible.[18]

Recall that existing transform learning algorithms require W to be left invertible. The relationship between left invertibility of W and $\mathcal{H}_W$ show that existing transform learning algorithms are, in fact, learning perfect reconstruction downsampled filter banks!

As an aside, note that if an additive ℓ_0 sparsity penalty is used, as in (4), Z is found from WX by means the elementwise nonlinearity $\mathcal{T}_\nu(\cdot)$. Using the filter bank interpretation, Z is found by applying a pointwise nonlinearity to each channel of the filter bank as illustrated in Figure 1. Note that this is true for any penalty that is separable ($\psi(Z) = \sum_i \psi(Z_i)$), such as an ℓ_1 penalty.

In practice, we usually do not operate on nonoverlapping patches of the image, instead choosing $1 \leq s < K$. We can still view WX as passing x through a filter bank with uniform downsampling by a factor of s, however the matrix W no longer corresponds to the polyphase matrix of this filter bank. In this case, requiring W to be left invertible is stronger condition than requiring $\mathcal{H}_W$ to be perfect reconstruction.

This prompts two questions: do we benefit by placing the burden of invertibility on $\mathcal{H}_W$ rather than W, and if so, can we devise an efficient algorithm to learn such a filter bank?

We restrict our attention to maximally overlapping patches ($s = 1$), implying that $\mathcal{H}_W$ is an undecimated filter bank as illustrated in Figure 1. Our first task is to draw a connection between the matrix of filter coefficients, W, and the left invertibility of the filter bank $\mathcal{H}_W$. In general, it is difficult to determine when a multidimensional undecimated filter bank satisfies the perfect reconstruction condition. We simplify matters by considering filter banks that implement circular, rather than linear, convolution, corresponding to the common strategy of extracting patches that are allowed to wrap around the image boundary. Using circular convolution provides easy access to the eigenvalues of $\mathcal{H}_W^T \mathcal{H}_W$ and thus grants a simple way to characterize perfect reconstruction filter banks.

We will make use of a matrix representation of the filter bank $\mathcal{H}_W$. Two dimensional circular convolution can be implemented as a matrix-vector multiplication, where the matrix is block-circulant with circulant blocks. In an abuse of notation, we will refer to these matrices as simply "circulant".

Let C_{w_j} be the circulant matrix such that $C_{w_j} x$ computes the circular convolution of the signals w_j and x. We stack these matrices to form $C_W = [C_{w_1}^T, C_{w_2}^T, \ldots, C_{W_{N_C}}^T]^T$, so that $\mathcal{H}_W x = C_W x$. Let $\Phi \in \mathbb{C}^{N^2 \times N^2}$ be the orthonormal 2D discrete Fourier transform (DFT) matrix, constructed as the Kronecker product of a normalized 1D DFT matrix with itself. For $K < N$, we represent an $N \times N$ 2D-DFT of a $K \times K$ signal using the matrix $\bar{\Phi} \in \mathbb{C}^{N^2 \times K^2}$. This is equivalent to padding the $K \times K$ matrix with $(N - K)$ rows and columns of zeros before performing the DFT.

We denote a length k vector of all ones as $\mathbf{1}_k$. Given a vector (resp. matrix) x, the operation $|x|^2$ is performed independently to each element of the vector (resp. matrix). Finally, given a vector $x \in \mathbb{R}^n$, the matrix $\mathrm{ddiag}(x) \in \mathbb{R}^{n \times n}$ is diagonal with i-th diagonal entry given by the i-th element of x.

With these definitions in place, we relate the perfect reconstruction property of $\mathcal{H}_W$ to the matrix W though the following two simple results. Our characterization of perfect reconstruction circulant filter banks is based on a simple diagonalization of the matrix $C_W^T C_W$.

Lemma 2.1. *Given $W \in \mathbb{R}^{N_c \times K^2}$, the matrix $\mathcal{H}_W^T \mathcal{H}_W = C_W^T C_W$ can be diagonalized as $\Phi^H \operatorname{ddiag}\left(\left|\bar{\Phi} W^T\right|^2 \mathbf{1}_{N_c}\right) \Phi$.*

Proof. The circulant matrix C_{w_j} can be diagonalized as $\Phi D_j \Phi^*$, where $D_j = \operatorname{ddiag}\left(\bar{\Phi} W_{j,:}\right)$ is a diagonal matrix constructed from the vector $\bar{\Phi} W_{j,:}$. Then, we have $C_W^T C_W = \sum_{j=1}^{N_c} C_{w_j}^T C_{w_j} = \Phi(\sum_{j=1}^{N_c} D_j^* D_j)\Phi^* = \Phi D \Phi^*$, where $D \triangleq \operatorname{ddiag}\left(\sum_{j=1}^{N_c} \left|\bar{\Phi} W_{j,:}\right|^2\right) = \operatorname{ddiag}\left(\left|\bar{\Phi} W^T\right|^2 \mathbf{1}_{N_c}\right)$ has the eigenvalues of $C_W^T C_W$ along its diagonal. □

Corollary 1. *The undecimated circulant filter bank $\mathcal{H}_W$ is PR if and only if each entry of $\left|\bar{\Phi} W^T\right|^2 \mathbf{1}_{N_c}$ is strictly positive.*

Proof. This follows immediately from Lemma 2.1 and properties of symmetric positive semidefinite matrices. □

Lemma 1 provides straightforward way to express the PR property of a $\mathcal{H}_W$ in terms of its coefficient matrix W. Importantly, this result shows that W need not be left invertible for $\mathcal{H}_W$ to be a perfect reconstruction filter bank. This represents a significant relaxation of the constraints in existing transform learning algorithms. Because we no longer require W to be left invertible, we are no longer restricted to choosing square or tall W. The number of channels can be chosen independently of the filter size, and allows for fat W for the first time. This property is especially desirable for high dimensional data- even in three dimensional data, a square W operating on modest $8 \times 8 \times 8$ patches must have 512 rows!

It should be noted that the relationship between patch-based analysis operators and convolution has been previously explored,[14–16] primarily as a computational tool. However, these works did not relate perfect reconstruction filter banks with left invertible analysis operators.

3. FILTER BANK TRANSFORM LEARNING

In this section, we formulate the problem of *learning* a filter bank sparsifying transform from data.

We have three equivalent ways to express the action of our filter bank on an image: $\mathcal{H}_W x, C_W x$, and WX, where X is formed by extracting patches from x with unit stride, including those that wrap around the image boundary. The final representation, WX, is attractive as it provides a concise parameterization of the learning problem. This will be our primary representation of the action of the filter bank.

3.1 Problem Formulation

Our goal is to learn a perfect reconstruction filter bank that sparsifies our data, with the ultimate goal of using this filter bank as a regularizer for inverse problems. We want our filter bank to be well conditioned as to prevent undue noise amplification and provide a stable representation of the data. Further, we do not want our filter bank to contain filters that are identically zero or copies of other filters, as these do not provide additional insight to our data.

Similar to previous work on transform learning, we encourage sparsifying filters by using the penalty $F(W, Z, x) = \frac{1}{2}\|WX - Z\|_F^2 + \nu \|Z\|_0$. These terms ensure that the filtered data, WX, is close to a sparse matrix Z. We promote well conditioned filter banks, containing no uniformly zero filters, by adding the penalty

$$J_1(W) = \left\|\left|\bar{\Phi} W^T\right|^2 \mathbf{1}_{N_c} - \mathbf{1}_{N^2}\right\|_2^2 - \beta \sum_{j=1}^{N_c} \log\left(\|W_{j,:}\|_2^2\right). \tag{5}$$

The first term encourages the eigenvalues of $\mathcal{H}_W^T \mathcal{H}_W$ to be close to unity. The second term is a log barrier to ensure no filters are identically zero.

Finally, we wish to encourage diversity in our learned filters. This is accomplished by penalizing the coherence between the filters through a form of a penalty used in an existing algorithm for analysis operator learning.[10] In particular, we use

$$J_2(W) = \sum_{1 \leq i < j \leq N_c} -\log\left(1 - \left(\frac{\langle W_{i,:}, W_{j,:}\rangle}{\|W_{i,:}\|_2 \|W_{j,:}\|_2}\right)^2\right). \tag{6}$$

Our complete learning problem is written as

$$\min_{W,Z} \frac{1}{2}\|WX - Z\|_F^2 + \nu\|Z\|_0 + \alpha J_1(W) + \gamma J_2(W). \tag{7}$$

3.2 Optimization Strategy

We utilize an alternating minimization algorithm to minimize (7). In the first stage, we hold W fixed and minimize (7) over Z. This is the so-called *sparse coding* step, and the solution $Z^{(k+1)}$ can be obtained directly by hard thresholding as $Z^{(k+1)} = \mathcal{T}_\nu\left(W^{(k)}X\right)$. In the second stage, we hold Z fixed and update W by solving

$$\min_{W} \frac{1}{2}\|WX - Z\|_F^2 + \alpha J_1(W) + \gamma J_2(W). \tag{8}$$

This is referred to as the *transform update* step. Unlike the square and patch-based case, we do not have a closed form solution to the transform update step, and must utilize an iterative approach. Fortunately, the objective function (7) is differentiable with respect to W and there are many iterative algorithms to choose from. We have found the limited-memory BFGS (L-BFGS) algorithm to work well in practice. The required gradients are given by

$$\nabla_W \|WX - Z\|_F^2 = 2WXX^T - 2ZX^T, \tag{9}$$

$$\nabla_W \left\| \left|\bar{\Phi} W^T\right|^2 \mathbf{1}_{N_c} - \mathbf{1}_{N^2}\right\|_2^2 = 4W\bar{\Phi}^* \operatorname{ddiag}\left(\left|\bar{\Phi}W^T\right|^2 \mathbf{1}_{N_c} - \mathbf{1}_{N^2}\right)\bar{\Phi}, \tag{10}$$

$$\frac{\partial}{\partial W_{r,s}} \sum_{k=1}^{N_c} \log\left(\|W_{k,:}\|_2^2\right) = \frac{2W_{r,s}}{\|W_{r,:}\|_2^2}, \tag{11}$$

$$\frac{\partial}{\partial W_{r,s}} J_2(W) = 2\sum_{i \neq s} \frac{W_{i,r}\|W_{s,:}\|_2^2([WW^T]_{s,i}) - W_{r,s}([WW^T]_{s,i})^2}{\|W_{i,:}\|_2^2\|W_{s,:}\|_2^4 - \|W_{s,:}\|_2^2[WW^T]_{s,i}}. \tag{12}$$

Note that as the W update step is a subproblem of a larger iterative algorithm, we do not require L-BFGS to fully converge. We found that terminating after a fixed number of L-BFGS iterations worked well.

Our filter bank transform learning algorithm is summarized as Algorithm 1. A stopping criterion can be developed by looking at the relative change in the iterates $W^{(k)}$, but we found that a fixed number of iterations worked well in practice.

Care must be taken to initialize the algorithm with a W that contains no duplicated or uniformly zero rows as to ensure that the log barrier terms are finite. We typically use a random initialization. We have found that the algorithm occasionally gets stuck and yields transforms that are very poorly conditioned. In each of these cases, the eigenvalue of $\mathcal{H}_W^T\mathcal{H}_W$ corresponding to zero frequency (DC) was nearly zero. A possible fix is to constrain one filter to consist of all ones to ensure that low frequencies are passed. However, we found that initializing the algorithm with one of the filters set to all ones to be sufficient in preventing this behavior. Although the filter was allowed to vary during the training procedure, in each case it retained its low-pass behavior.

4. APPLICATION TO MAGNETIC RESONANCE IMAGING

In this section we demonstrate the use of filter bank sparsifying transforms as a regularizer for magnetic resonance imaging (MRI).

Algorithm 1 Filter bank sparsifying transform learning

```
INPUT: Image $x$, Initial transform $W^{(0)}$
$X \leftarrow$ unit stride patches of $x$
$Z^{(0)} \leftarrow \mathcal{T}_\nu \left( W^0 X \right)$
$k \leftarrow 0$
repeat
    $Z^{(k+1)} \leftarrow \mathcal{T}_\nu \left( W^{(k)} X \right)$
    $W^{(k+1)} \leftarrow \arg\min_W 0.5\|WX - Z^{(k+1)}\|_F^2 + \alpha J_1(W) + \gamma J_2(W)$
    $k \leftarrow k+1$
until Halting Condition
```

While MRI provides a means to noninvasively determine both anatomical structure and physiological function, it is a relatively slow imaging modality. As such, a great deal of effort has been spent in reducing the amount of data necessary for MR image reconstruction.

We model the relationship between the MR data $y \in \mathbb{C}^M$ and the image $x \in \mathbb{R}^{N^2}$ as

$$y = \Gamma \Phi x + e, \tag{13}$$

where e represents zero mean Gaussian noise with variance σ^2. The matrix $\Phi \in \mathbb{C}^{N^2 \times N^2}$ represents the 2D orthonormal Fourier matrix. The row selection matrix $\Gamma \in \mathbb{R}^{M \times N^2}$ is formed by selecting $M < N^2$ rows from the $N^2 \times N^2$ identity matrix I_{N^2}. Thus y is formed by subsampling the Fourier transform of x. We aim to reconstruct the image from the measurements while jointly learning a sparsifying filter bank for this data. To that end, we solve

$$\min_{x, \mathcal{H}_W, z} \frac{1}{2}\|y - \Gamma\Phi x\|_2^2 + \lambda \left(\frac{1}{2}\|\mathcal{H}_W x - z\|_2^2 + \nu\|z\|_0 + \alpha J_1(\mathcal{H}_W) + \gamma J_2(\mathcal{H}_W) \right), \tag{14}$$

where the parameter $\lambda > 0$ controls the strength given to the sparsifying transform regularizer. The first term of the objective function enforces data fidelity in Fourier space. The second term ensures that the reconstructed image is well sparsified by the transform $\mathcal{H}_W$, and the remaining terms ensure we learn a "good" sparsifying transform.

We solve (14) by using alternating minimization. We begin by fixing x and updating $\mathcal{H}_W$ and z as described in Section 3. Then, with these quantities fixed, we update x by minimizing (14) over x. This problem simplifies to

$$\min_x \frac{1}{2}\|y - \Gamma\Phi x\|_2^2 + \frac{\lambda}{2}\|\mathcal{H}_W x - z\|_F^2 \tag{15}$$

which is a least squares problem in x. The solution is given in closed form by

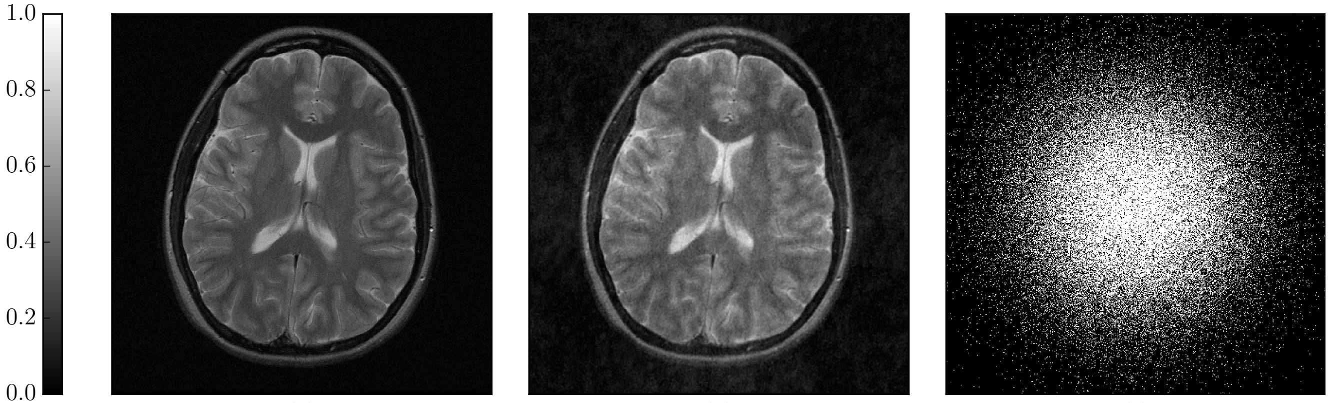
$$x^* = \left(\lambda \mathcal{H}_W^T \mathcal{H}_W + \Phi^H \Gamma^T \Gamma \Phi \right)^{-1} \left(\lambda \mathcal{H}_W^T z + \Phi^* \Gamma^T y \right). \tag{16}$$

Fortunately, when $\mathcal{H}_W$ implements circular convolution, the necessary matrix inversion can be solved cheaply. Observe that $\Gamma^T\Gamma$ is an $n \times n$ diagonal matrix with only ones or zeros along its diagonal. Further, we have $\mathcal{H}_W^T \mathcal{H}_W = \Phi^H D \Phi$, where the diagonal matrix D is found by way of Lemma 2.1. Thus we can rewrite (17) solution as

$$x^* = \Phi^H \left(\lambda D + \Gamma^T \Gamma \right)^{-1} \Phi \left(\lambda \mathcal{H}_W^T z + \Phi^H \Gamma^T y \right), \tag{17}$$

which requires only a single FFT and inverse FFT pair and multiplication by a diagonal matrix. We can identify $\Phi^H \Gamma^T y$ as a zero-filled reconstruction from the undersampled Fourier measurements. Thus the solution (17) can be thought of as a passing a weighted sum of $\mathcal{H}_W^T z$ and a zero-filled reconstruction through the filter $\lambda \mathcal{H}_W^T \mathcal{H}_W + \Phi^H \Gamma^T \Gamma \Phi$.

As MR images are typically constrained to a finite region surrounded by empty space, we do not expect any distortion due to the use of circular convolution. Indeed, patch-based methods for MRI often extract patches that wrap around the image boundary.



(a) (b) (c)

Figure 2. (a) Input magnitude image. (b) Zero filled reconstruction using 23% of Fourier measurements. (c) Random sampling mask.

The proposed algorithm is summarized as Algorithm 2. This algorithm differs from earlier work on sparsifying transforms for MRI[7] in two key ways. First, the previous work utilized a patch-based transform learning formulation. Second, rather than the additive ℓ_0 penalty we use here, the authors constrained the sparsity of each transformed patch to lie beneath a given threshold. This threshold is varied on a patch-by-patch basis. To facilitate this, the authors used a variable splitting approach to separate the transform sparse coding step from the image reconstruction step. This consists of two steps. In the first step, each image patch is denoised independently. In the second step, the denoised patches are averaged with the zero-filled reconstruction to provide an updated image estimate. If the variable patch sparsity constraint is replaced by a single sparsity level or an additive penalty, this two-step procedure can be replaced with the closed form update of Algorithm 2.

Algorithm 2 MR image reconstruction with filter bank transform

```
INPUT: Image $x$, Initial transform $W^{(0)}$
repeat
    $Z^{(k+1)}, W^{(k+1)} \leftarrow$ Output of Algorithm 1
    $D^{(k+1)} \leftarrow \text{ddiag}\left(\left|\bar{\Phi}(W^{(k+1)})^T\right|^2 \mathbf{1}_{N_c}\right)$
    $x^{(k+1)} \leftarrow \Phi^H(\Gamma^T\Gamma + \lambda D^{(k+1)})^{-1}\Phi((\lambda\mathcal{H}_W^{(k+1)})^T z^{(k+1)} + \Phi^H\Gamma y)$
    $k \leftarrow k + 1$
until Halting Condition
```

5. EXPERIMENTS

We evaluated our algorithm using 512×512 fully sampled MRI image of a human brain (provided by Prof. Micheal Lustig). We synthesized MR measurements satisfying (13) by taking a 2D DFT of the image and retaining only 23% of the Fourier coefficients. The reference image, sampling mask, and a zero-filled reconstruction are shown in Figure 2. The reference image was normalized to have pixel values ranging from 0 to 1. We tested out algorithm with no noise (undersampling only) and additive noise with $\sigma = 10/255$ and $\sigma = 20/255$.

We compare our filter bank sparsifying transform (FBST) algorithm against a square patch-based sparsifying transform (PBST) algorithm. Unlike previous work on MR reconstruction using patch-based transforms we utilize an additive ℓ_0 penalty, so the two approaches considered here differ only in the choice of regularization for the transform update step. We make use of the closed-form solution for square transform learning.[2]

We evaluate the FBST approach using a variety of filter sizes and number of channels. The PBST uses 8×8 patches, so we have $W \in \mathbb{R}^{64\times 64}$. Our primary metric is PSNR (in dB), defined as $PSNR = 20\log_{10}(255/\sum_{i=1}^{N^2}(x_i - x_i^*))$ where x^* represents the reference image. The best choice of parameters for Algorithm 2 remains an open question. We set $\alpha = 0.2, \beta = 1 \times 10^{-4}$, and $\gamma = 5 \times 10^{-5}$. The remaining parameters λ and ν were optimized for each noise level and filter bank configuration. Both FBST and PBST were initialized with a discrete cosine transform matrix.

Examples of learned filters are shown in Figure 3. The magnitude response of each channel are also shown, with zero frequency located at the center of each image. While some of the filters resemble the DCT filters, others filters exhibit a highly directional bandpass frequency response. Both filter banks are well conditioned and have a condition number of around 1.5.

Table 1 lists the recovery PSNR for each noise level and filter bank configuration. In each case, the FBST algorithm outperforms the PBST algorithm. For $N_c = 64$ and $K = 8$, the only difference in the two algorithms is the type of underlying model- PBST is a patch-based model that neglects correlations in neighboring patches, while FBST makes use of this redundant information. This result illustrates the advantage in chosing an image-based signal model. We note that our FBST learning algorithm is signficantly slower than the square PBST learning algorithm, owing to a lack of a closed-form solution to the transform update step.

Table 1 shows that there is not much benefit in choosing a 2× overcomplete filter bank or larger filters for this experiment. We believe that the benefit in choosing overcomplete transforms, or larger filter sizes, will be more pronounced when learning a transform that must sparsify a large number of images or when learning from high-quality data.

Table 1. Image Reconstruction PSNR

σ / PSNR	FBST			PBST
	$N_c = 64$, $K = 8$	$N_c = 128$, $K = 8$	$N_c = 64$, $K = 12$	64×64
0 / 29.6	35.15	**35.22**	35.13	34.63
$\frac{10}{255}$ / 28.8	32.62	**32.72**	32.60	32.53
$\frac{20}{255}$ / 26.9	**31.68**	31.61	31.21	31.30

6. CONCLUSION

We have developed a new framework for learning filter bank sparsifying transforms. Unlike previous work on transform learning, these transforms operate at the image level, and allow for the filter length to be chosen independently of the number of channels. We anticipate this flexibility will be beneficial for high dimensional data. Numerical results illustrate that filter bank sparsifying transforms outperform square patch-based sparsifying transforms for MR image reconstruction.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (NSF) under grant CCF-1320953 and the Andrew T. Yang Fellowship.

REFERENCES

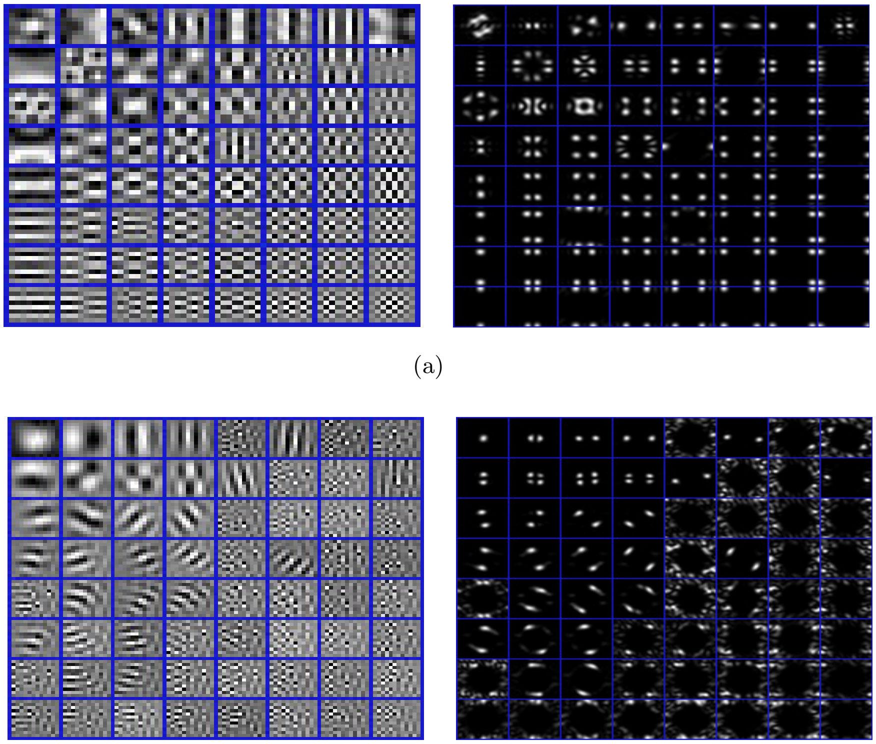
[1] Ravishankar, S. and Bresler, Y., "Learning sparsifying transforms," *IEEE Trans. Signal Process.* **61**(5), 1072–1086 (2013).

[2] Ravishankar, S. and Bresler, Y., "Closed-form solutions within sparsifying transform learning," in [*Acoustics Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*], (2013).

[3] Ravishankar, S. and Bresler, Y., "Learning overcomplete sparsifying transforms for signal processing," in [*Acoustics, Speech and Sig. Proc (ICASSP)*], 3088–3092 (2013).

[4] Wen, B., Ravishankar, S., and Bresler, Y., "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *International Journal of Computer Vision* (Oct 2014).

[5] Pfister, L. and Bresler, Y., "Tomographic reconstruction with adaptive sparsifying transforms," in [*Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*], 6914–6918 (May 2014).



(a)

(b)

Figure 3. Examples of learned filters and their magnitude respones. Filters learned while jointly reconstructing MR image with $\sigma = 0$. Both filter banks have 64 channels. (a) 8×8 filters. (b) 12×12 filters.

[6] Ravishankar, S. and Bresler, Y., "Learning doubly sparse transforms for images," *IEEE Transactions on Image Processing* **22**(12), 4598–4612 (2013).

[7] Ravishankar, S. and Bresler, Y., "Sparsifying transform learning for compressed sensing MRI," in [*International Symposium on Biomedical Imaging*], (2013).

[8] Rubinstein, R., Peleg, T., and Elad, M., "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Trans. Signal Process.* **61**, 661–677 (Feb. 2013).

[9] Yaghoobi, M., Nam, S., Gribonval, R., and Davies, M. E., "Constrained overcomplete analysis operator learning for cosparse signal modelling," *IEEE Trans. Signal Process.* **61**, 2341–2355 (May 2013).

[10] Hawe, S., Kleinsteuber, M., and Diepold, K., "Analysis operator learning and its application to image reconstruction," *IEEE Trans. Image Process.* **22**, 2138–2150 (June 2013).

[11] Pfister, L. and Bresler, Y., "Model-based iterative tomographic reconstruction with adaptive sparsifying transforms," *Proc. SPIE Computational Imaging XII* , 90200H–90200H–11, SPIE (Mar 2014).

[12] Pfister, L. and Bresler, Y., "Adaptive sparsifying transforms for iterative tomographic reconstruction," in [*International Conference on Image Formation in X-Ray Computed Tomography*], (2014).

[13] Zoran, D. and Weiss, Y., "From learning models of natural image patches to whole image restoration," in [*Computer Vision (ICCV), 2011 IEEE International Conference on*], 479–486 (2011).

[14] Roth, S. and Black, M. J., "Fields of experts," *International Journal of Computer Vision* **82**(2), 205–229 (2009).

[15] Chen, Y., Ranftl, R., and Pock, T., "Insights into analysis operator learning: From patch-based sparse models to higher order MRFs," *IEEE Transactions on Image Processing* **23**, 1060?1072 (Mar 2014).

[16] Cai, J.-F., Ji, H., Shen, Z., and Ye, G.-B., "Data-driven tight frame construction and image denoising," *Applied and Computational Harmonic Analysis* **37**(1), 89–105 (2014).

[17] Malvar, H. S., [*Signal Processing with Lapped Transforms*], Artech Print on Demand (1992).

[18] Strang, G. and Nguyen, T., [*Wavelets and Filter Banks*], Wellesley College (1996).