LEARNING OVERCOMPLETE SPARSIFYING TRANSFORMS FOR SIGNAL PROCESSING

Saiprasad Ravishankar and Yoram Bresler

Department of Electrical and Computer Engineering and the Coordinated Science Laboratory,
University of Illinois, Urbana-Champaign, IL 61801, USA

ABSTRACT

Adaptive sparse representations have been very popular in numerous applications in recent years. The learning of synthesis sparsifying dictionaries has particularly received much attention, and such adaptive dictionaries have been shown to be useful in applications such as image denoising, and magnetic resonance image reconstruction. In this work, we focus on the alternative sparsifying transform model, for which sparse coding is cheap and exact, and study the learning of tall or overcomplete sparsifying transforms from data. We propose various penalties that control the sparsifying ability, condition number, and incoherence of the learnt transforms. Our alternating algorithm for transform learning converges empirically, and significantly improves the quality of the learnt transform over the iterations. We present examples demonstrating the promising performance of adaptive overcomplete transforms over adaptive overcomplete synthesis dictionaries learnt using K-SVD, in the application of image denoising.

Index Terms— Sparsifying transform learning, Sparse representations, dictionary learning, Overcomplete representations

1. INTRODUCTION

1.1. Synthesis and Analysis Models

Sparse representation of signals and images has become widely popular in recent years. Two well-known models for sparse representation are the synthesis model and the analysis model [1]. The synthesis model suggests that a signal $y \in \mathbb{R}^n$ may be represented as a linear combination of a small number of atoms from a synthesis dictionary $D \in \mathbb{R}^{n\times K}$. Hence, $y = Dx$ with $x \in \mathbb{R}^K$ being sparse, i.e., $\|x\|_0 \ll K$, and the l_0 quasi norm counts the number of non-zero entries in x. "Real world" signals are more generally assumed to satisfy $y = Dx + e$, where e is an approximation/noise term in the signal domain [2]. When $K = n$ and D is full rank, we have a basis representation. When $K > n$, the dictionary is said to be overcomplete.

Given the signal y and synthesis dictionary D, the problem of finding the sparse representation x is known as the synthesis sparse coding problem [3]. The problem is to find x that minimizes $\|y - Dx\|_2^2$ subject to $\|x\|_0 \leq s$, where s is the required sparsity level. This problem is NP-hard (Non-deterministic Polynomial-time hard). However, under certain conditions it can be solved exactly using polynomial-time algorithms [4, 5, 6]. These algorithms are typically computationally expensive, particularly for large-scale problems [7].

On the other hand, the analysis model [1, 8] suggests that a signal y is sparse in an "analysis" domain, i.e., given an analysis dictionary $\Omega \in \mathbb{R}^{m\times n}$, we have $\Omega y \in \mathbb{R}^m$ to be sparse ($\|\Omega y\|_0 \ll m$). When the signal y is contaminated with noise, it is more generally assumed to satisfy a noisy signal analysis model, which states that $y = q + e$ with Ωq being sparse, and e is the noise in the signal domain.

Given the noisy signal y and analysis dictionary Ω, the problem of finding the noiseless q is known as analysis sparse coding [8], with Ωq representing the sparse code. This problem is to find q by minimizing $\|y - q\|_2^2$ subject to $\|\Omega q\|_0 \leq m - l$, where l is referred to as the co-sparsity level (minimum number of zeros allowed in Ωq) [8]. This problem too is NP-hard just like sparse coding in the synthesis model. Moreover, when Ω is square and non-singular, then $q = \Omega^{-1} z$ for some sparse z, and the problem of finding q is identical to a synthesis sparse coding problem (of finding z), with Ω^{-1} being the synthesis dictionary. Similarly to sparse coding in the synthesis model, approximate algorithms exist for analysis sparse coding [8, 9, 10], which however, tend to be computationally expensive.

1.2. Transform Model - A Generalized Analysis Model

Recently, we considered a generalization of the analysis model, which we call the transform model [11]. It suggests that a signal y is approximately sparsifiable using a transform $W \in \mathbb{R}^{m\times n}$. Here, the assumption is that $Wy = x + \eta$, where $x \in \mathbb{R}^m$ is sparse, i.e., $\|x\|_0 \ll m$, and η is the residual in the transform domain. Natural signals and images are well-known to be approximately sparse in analytical transform domains such as Wavelets [12], discrete cosine transform (DCT), Ridgelets [13], Contourlets [14], and Curvelets [15]. The transform model is a generalization of the analysis model with Ωy exactly sparse. The generalization allows the transform model to include a wider class of signals within its ambit than the analysis model. Moreover, while the analysis model enforces the sparse code (Ωy) to lie in the range space of Ω, the sparse representation x in the transform model is not forced to lie in the range space of W. This makes the transform model more general than even the noisy signal analysis model (cf. [11]). The reason we have chosen the name "transform model" is because the assumption $Wy \approx x$ has been traditionally used in transform coding (with orthonormal transforms), and the concept of transform coding is older [16] and pre-dates the terms analysis and synthesis [17].

When a suitable sparsifying transform W is known for the signal y, the process of obtaining a sparse code x of given sparsity s involves minimizing $\|Wy - x\|_2^2$ subject to $\|x\|_0 \leq s$. We call this transform sparse coding for simplicity. This problem is easy and its solution is obtained exactly by hard-thresholding the product Wy (i.e., retaining only the s largest coefficients). Given the W and sparse code x, we can also recover a least squares estimate of the true signal y by minimizing $\|Wy - x\|_2^2$ over all $y \in \mathbb{R}^n$. The recovered signal is $W^{\dagger} x$, with $W^{\dagger}$ denoting the pseudo-inverse of W. Thus, unlike the previous models, the transform model allows

This work was supported in part by the National Science Foundation (NSF) under grant CCF 10-18660.

for exact and fast computations, a property that has been exploited heavily in the context of analytical sparsifying transforms.

Recent research has focused on adapting the sparse models to data. The learning of synthesis dictionaries from training signals has been studied by many authors [18, 19, 20]. The learnt dictionaries have been shown to be useful in numerous applications [21, 22, 23, 7, 24]. However, the synthesis dictionary learning problems are typically NP-hard and non-convex, with popular algorithms such as K-SVD [19] likely to get caught in local minima. Another very recent development has been the study of adaptive analysis models. Numerous authors have attempted to learn analysis dictionaries [25, 26, 10, 8]. However, analysis dictionary learning is also typically non-convex and NP-hard, and no theoretical or empirical global/local convergence properties have been demonstrated for the various learning algorithms.

We have very recently developed formulations and algorithms for square transform learning [11]. The algorithms therein have a much lower computational cost compared to synthesis and analysis dictionary learning, and moreover, also provide convergence of the cost and iterates regardless of initial conditions. In this paper, we however focus on the learning of *overcomplete* or tall sparsifying transforms, i.e., $W \in \mathbb{R}^{m \times n}$, with $m > n$. We illustrate the convergence of our learning algorithm, and demonstrate its usefulness in image denoising.

2. TRANSFORM LEARNING

2.1. Square Transform Learning

Given a matrix $Y \in \mathbb{R}^{n \times N}$ whose columns represent training signals, a formulation for learning a square transform $W \in \mathbb{R}^{n \times n}$ has been proposed by us [11] as follows.

$$\text{(P1)} \quad \min_{W,X} \|WY - X\|_F^2 - \lambda \log\det W + \mu \|W\|_F^2$$
$$s.t. \quad \|X_i\|_0 \leq s \;\; \forall \; i$$

Here, $X \in \mathbb{R}^{n \times N}$ is a matrix with columns X_i, that are the sparse codes of the training signals, or columns in Y. The first term in the cost of (P1) is called *sparsification error* [11]. It represents the deviation of the data in the transform domain from perfect sparsity at sparsity level s. The $-\log\det W$ penalty enforces non-singularity of the transform W, and helps eliminate degenerate solutions such as those with zero rows, or repeated rows [11]. The $\|W\|_F^2$ penalty helps remove a 'scale ambiguity' [11] in the solution, which occurs when the data admits an exactly sparse representation.

The $-\log\det W$ and $\|W\|_F^2$ penalties are functions of the singular values of W (for $\det W > 0$), and together additionally help control the condition number of the learnt transform [11]. Badly conditioned transforms typically convey little information and may degrade performance in applications. Well conditioned adaptive transforms have been shown to be useful in applications [11, 27]. As the parameter $\lambda \to \infty$ with fixed μ/λ, the condition number of the optimal transforms tends to 1 [11]. Note that the restriction $\det W > 0$, can be made without loss of generality in (P1) [11] (one can switch from a W with $\det W < 0$ to one with $\det W > 0$, trivially by pre-multiplying W with a diagonal sign matrix Γ, with $\det \Gamma < 0$). Furthermore, the $\det W > 0$ constraint need not be enforced explicitly in (P1). This is because the cost function has log-barriers in the space of matrices at W with $\det W \leq 0$. These log-barriers prevent an iterative minimization algorithm initialized with W satisfying $\det W > 0$ from getting into the infeasible regions, where $\det W \leq 0$.

2.2. Overcomplete Transform Learning

We now extend (P1) to the overcomplete transform ($W \in \mathbb{R}^{m \times n}$, $m > n$) case. For the overcomplete or tall case, we replace $\log\det W$ in (P1) with $\log\det\left(W^T W\right)$, which would enable full column rank of W. Note that in this case, $\det\left(W^T W\right)$ is always non-negative. The $\log\det\left(W^T W\right)$ and $\|W\|_F^2$ penalties together help control the conditioning of the columns of W. However, good conditioning of $W^T W$ alone is not sufficient to ensure meaningful tall transforms. For instance, consider a tall W of the form

$$W = \begin{bmatrix} W_1 \\ 0_{m-n \times n} \end{bmatrix}$$

where W_1 is a well-conditioned square transform learnt using (P1) and $0_{m-n \times n}$ is a matrix of zeros. In this case, $W^T W$ is well-conditioned, since $W^T W = W_1^T W_1$. Moreover, W is a candidate sparsifying 'tall' transform. However, such a tall W has the ambiguity of repeated zero rows and the penalty $\log\det\left(W^T W\right)$ is unable to preclude such a W.

Hence, we introduce an additional penalty $\sum_{j \neq k} |\langle w_j, w_k \rangle|^p$, that enforces incoherence between the rows of W, denoted as w_j ($1 \leq j \leq m$). The notation $\langle \cdot, \cdot \rangle$ stands for the standard inner product between vectors. Note that larger values of p emphasize the peak coherence. When $p = 2$, we can consider for example, a $W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$ that is a concatenation of two orthonormal transforms W_1 and W_2 (e.g., DCT and identity). The penalty $\sum_{j \neq k} \langle w_j, w_k \rangle^2$ is, however, a fixed constant when W consists of such orthonormal blocks, irrespective of the choice of those blocks. For this reason, we consider $p \gg 2$ (e.g., a large even natural number), to enforce better incoherence.

We also additionally constrain the rows of W to unit norm. Under this constraint, the penalty $|\langle w_j, w_k \rangle|$ truly measures the incoherence (or angle) between the rows w_j and w_k. Thus, our problem formulation for overcomplete transform learning is as follows.

$$\text{(P2)} \quad \min_{W,X} \|WY - X\|_F^2 - \lambda \log\det\left(W^T W\right)$$
$$+ \eta \sum_{j \neq k} |\langle w_j, w_k \rangle|^p$$
$$s.t. \quad \|X_i\|_0 \leq s \;\; \forall \; i, \;\; \|w_k\|_2 = 1 \;\; \forall \; k$$

where $\eta > 0$ weights the incoherence penalty. Note that the $\|W\|_F^2$ penalty is a constant under the unit row norm assumption. Problem (P2) is however, non-convex.

2.3. Algorithm and Properties

Our algorithm for solving (P2) alternates between updating X and W. In one step called the **Sparse Coding Step**, we solve (P2) with fixed W as follows.

$$\min_{X} \|WY - X\|_F^2 \quad s.t. \quad \|X_i\|_0 \leq s \;\; \forall \; i \tag{1}$$

The solution $\hat{X}$ is computed exactly by thresholding WY, and retaining the s largest coefficients (in magnitude) in each column. Note that if the l_0 quasi norm for sparsity is relaxed to an l_1 norm and added as a penalty in the cost (1), we can still obtain an exact solution for X by soft thresholding [11].

In the second step of our algorithm called **Transform Update Step**, we solve Problem (P2) with fixed X as follows.

$$\min_{W} \; \|WY - X\|_F^2 - \lambda \log\det\left(W^TW\right) + \eta \sum_{j \neq k} |\langle w_j, w_k \rangle|^p$$
$$s.t. \;\; \|w_k\|_2 = 1 \;\; \forall \; k \qquad (2)$$

This problem does not have an analytical solution, and is moreover non-convex. We could solve for W using iterative algorithms such as the projected conjugate gradient method. However, we observed that the alternative strategy of employing the standard conjugate gradient (CG) algorithm, followed by post-normalization of the rows of W led to better empirical performance in applications. Hence, we choose the alternative strategy. When employing the standard CG, we also retain the $\|W\|_F^2$ penalty in the cost for CG, to prevent the scaling ambiguity [11].

The gradient expressions for the various terms in the cost (2) are as follows (cf. [28]). We choose p to be an even natural number (for simplicity), and assume $\det\left(W^TW\right) > 0$ on some neighborhood of W, otherwise $\log()$ would be discontinuous.

$$\nabla_W \log\det\left(W^TW\right) = 2W\left(W^TW\right)^{-1} \qquad (3)$$

$$\nabla_W \|WY - X\|_F^2 = 2WYY^T - 2XY^T \qquad (4)$$

$$\nabla_W \sum_{j \neq k} |\langle w_j, w_k \rangle|^p = 2p\,(ZW - B) \qquad (5)$$

The matrices $Z \in \mathbb{R}^{m \times m}$ and $B \in \mathbb{R}^{m \times n}$ above have entries $z_{ij} = \langle w_i, w_j \rangle^{p-1}$ and $b_{ij} = z_{ii} w_{ij}$.

The computational cost per iteration (of sparse update and transform update) of the proposed algorithm scales as $O(mnN)$ for learning an $m \times n$ transform from N training vectors. Note that this cost is typically much lower than the per-iteration cost of learning an $n \times K$ synthesis dictionary D using K-SVD [19], which scales as $O(Kn^2N)$ [7] (assuming that the synthesis sparsity $s \propto n$).

3. CONVERGENCE AND LEARNING

In this section, we illustrate the convergence of our learning algorithm for (P2), and its ability to learn meaningful transforms. We learn a 128×64 transform from the 8×8 non-overlapping patches of the Barbara image [19]. The means of the patches are removed, and we only sparsify the mean-subtracted patches. We fix our alorithm parameters as $s = 11$, $p = 20$, $\lambda = \eta = 4 \times 10^5$. The CG algorithm in the transform update step is executed for 128 iterations with a fixed step size of 10^{-9}. Note that we use a weighting $\mu = \lambda$ for the frobenius-norm penalty within CG. The algorithm is initialized with the (vertical) concatenation of the 2D DCT (obtained as the Kronecker product of two 8×8 1D DCT matrices) and identity matrices.

Figure 1 plots the objective function and sparsification error for our algorithm over iterations. Both the objective and sparsification error converge quickly. Moreover, the sparisifcation error improves significantly (by more than 8 dB) over the iterations compared to that of the initial transform (DCT concatenated with identity). We define the normalized sparsification error [11] as $\|WY - X\|_F^2 / \|WY\|_F^2$. It measures the fraction of energy lost in sparse fitting in the transform domain, which is an interesting property to observe for the adapted transforms. The normalized sparsification error for the final learnt transform is 0.08, while that for the initialization is 0.42, indicating the significantly enhanced sparsification ability of the adapted transform.

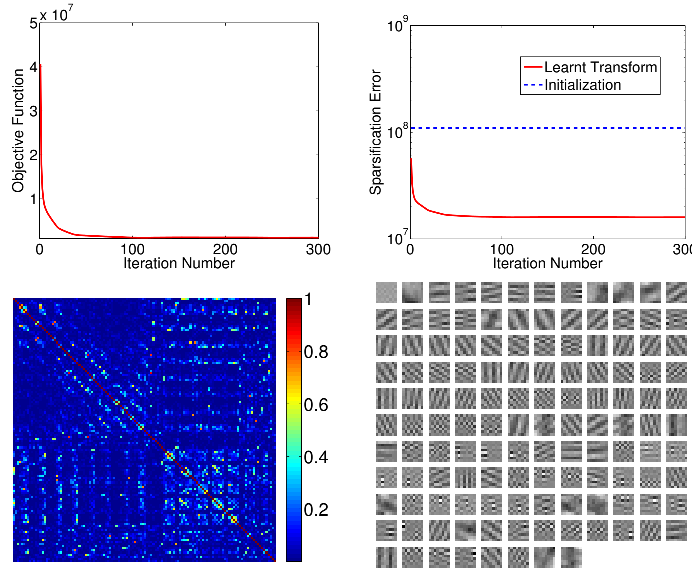

Fig. 1. Top: Objective function vs. iterations (left), Sparsification error vs. iterations along with the sparsification error of the initial transform (right). Bottom: Magnitude of WW^T(left), Rows of learnt transform as patches (right).

The learnt transform is shown in Figure 1, with each of its row displayed as an 8×8 patch, called the 'transform atom'. The atoms appear different from each other, and exhibit a lot of geometric and frequency like structures, which are reflective of the fact that the Barbara image has a lot of structure, textures. The learnt transform is also well-conditioned with a condition number of 2.4. The magnitude of WW^T, shown in Figure 1, indicates mostly small values for the off-diagonal elements. The mutual coherence of W [29] (maximum off-diagonal magnitude in WW^T) is 0.88. The results here indicate the fast convergence of our algorithm, and its ability to learn meaningful, non-trivial overcomplete transforms.

4. IMAGE DENOISING

Image denoising is a well-known problem of recovering an image $x \in \mathbb{R}^P$ (2D image represented as vector) from its measurement $y = x + g$ corrupted by noise g. We recently proposed a simple image denoising technique [27] based on adaptive sparsifying transforms. In this work, we propose the following formulation, which is an extension of our previous technique.

$$(P3) \;\; \min_{W, x_i, \alpha_i} \sum_{i=1}^{M} \|Wx_i - \alpha_i\|_2^2 + \lambda Q(W) + \tau \sum_{i=1}^{M} \|R_i\, y - x_i\|_2^2$$
$$s.t. \;\; \|\alpha_i\|_0 \leq s_i \;\; \forall \; i \;, \; \|w_k\|_2 = 1 \;\; \forall \; k$$

where $Q(W) = -\log\det\left(W^TW\right) + \frac{\eta}{\lambda} \sum_{j \neq k} |\langle w_j, w_k \rangle|^p$ represents the portion of the objective depending on only W. Operator $R_i \in \mathbb{R}^{n \times P}$ extracts a $\sqrt{n} \times \sqrt{n}$ patch from the noisy image y as $R_i y$ (we assume M overlapping patches). We model the noisy patch $R_i\, y$ as being approximated by a noiseless patch x_i, that is approximately sparsifiable in a adaptive transform W (i.e., the *noisy signal transform model* [11]). Vector $\alpha_i \in \mathbb{R}^n$ denotes the sparse code of x_i with s_i non-zeros. The weighting τ in (P3) is typically chosen as inversely proportional to the noise level σ [21].

Fig. 2. Noisy Images (left), Denoised Images (right).

Problem (P3) is non-convex. We propose a two step iterative procedure to solve (P3). In the *transform learning* **Step 1**, we fix $x_i = R_i\, y$ and $s_i = s$ (fixed input s initially) in (P3), and solve for W and $\alpha_i\ \forall i$, using the proposed overcomplete transform learning algorithm. In the *variable sparsity update* **Step 2**, we update the sparsity levels s_i for all i. For fixed W and α_i, (P3) is a least squares problem, which can be solved independently for each x_i. However, we don't fix α_i, but rather only let it be a thresholded version of $WR_i\, y$ (since learning was done on $R_i\, y$), and adaptively find the sparsity level s_i.

The sparsity level s_i for the i^{th} patch needs to be chosen such that the denoising error term $\|R_i\, y - x_i\|_2^2$ computed after updating x_i by least squares (with α_i held at $H_{s_i}(WR_i y)$, where $H_{s_i}(\cdot)$ is the operator that retains the s_i components of largest magnitude in a vector, and sets the remaining components to zero) is below $nC^2\sigma^2$ [21], with C being a fixed parameter. Note that the denoising error term (with the updated x_i) decreases to zero, as $s_i \nearrow n$. Thus, finding s_i requires in general, repeating the least squares update of x_i for each i at various sparsity levels incrementally, to determine the level at which the error term falls below the required threshold. However, this process can be done very efficiently (cf. [27] for details).

Once the variable sparsity levels s_i are chosen for all i, we use the new s_i's back in the transform learning Step 1, and iterate over the learning and variable sparsity update steps, which leads to a better denoising performance compared to one iteration. In the final iteration, the x_i's that are computed (satisfying the $\|R_i\, y - x_i\|_2^2 \leq nC^2\sigma^2$ condition) represent the denoised patches.

Once the denoised patches x_i have been estimated, the denoised image x is obtained by averaging the x_i's at their respective locations in the image. The x is then restricted to its range (e.g., 0-255), if known. Note that we work with mean subtracted patches in our algorithm and typically learn on a subset of all patches (cf. [27]). The means are added back to the denoised patch estimates.

We now present some preliminary results for our denoising framework, using our proposed overcomplete transform learning. The goal here is to illustrate the potential for adaptive overcomplete transforms in this classical and prototypical application. We add i.i.d. gaussian noise at noise level $\sigma = 10$ to the peppers image [21]. The denoising algorithm is executed for 3 iterations with parameters $n = 64$, $m = 100$, $\eta = \lambda = 8 \times 10^6$, $p = 20$, initial sparsity $s = 0.15 \times n$ (rounded to nearest integer), $C = 1.08$, and $\tau = 0.01/\sigma$. Transform learning is executed for 80 iterations (the weighting for the frobenius-norm term within CG is λ).

The noisy image (PSNR = 28.1 dB) is shown along with its denoised version (PSNR = 34.49 dB) in Figure 2. The learnt transform in this case has a condition number of 2.1 (well-conditioned), and also has incoherent rows (mutual coherence of 0.785). We compared our denoising performance to that obtained with the 64×256 K-SVD overcomplete synthesis dictionary [21, 30], which provided a lower denoising PSNR of 34.21 dB. Our denoising algorithm also takes less time (2.95 mins) compared to K-SVD (9.5 mins), due to the lower computational cost of sparse coding in the transform model. Note that we used a smaller training set for learning compared to K-SVD, since the 100×64 transform has fewer free parameters. The adapted overcomplete sparsifying transform also denoises better than the adapted square transform [11] learnt using Problem (P1) (PSNR for the latter is 34.38 dB), indicating the usefulness of overcompleteness.

We also repeat the denoising experiment with overcomplete transforms for the cameraman image using a high noise of $\sigma = 20$. The noisy image (PSNR = 22.1 dB), and its denoised version (PSNR = 29.95 dB) are shown in Figure 2. The denoising PSNR obtained using adaptive overcomplete transforms is better than that obtained using the 64×256 K-SVD synthesis dictionary (PSNR = 29.84 dB) [21, 30]. Our denoising algorithm is also 7x faster than K-SVD [21]. (Note that we used a smaller number of 20 learning iterations here, to prevent overfitting to high noise.) We expect the run times for our algorithm to decrease substantially with code optimization.

Transform-based denoising has also been shown to perform better than analysis-dictionary-based denoising [31]. We expect the denoising performance of our algorithms to improve/become comparable to the state of the art (for example [32]) with better choice of parameters, and with further extensions of transform learning (e.g., multiscale transforms).

5. CONCLUSIONS

In this work, we introduced a novel problem formulation for learning overcomplete sparsifying transforms. The proposed alternating algorithm for transform learning involves a sparse coding step and a transform update step. The solution of the sparse coding step is cheap and exact, and we use iterative methods (CG) for the transform update step. The learnt transforms have better properties compared to the initialization. Moreover, the computational cost of overcomplete transform learning is lower than that of overcomplete dictionary learning. We also applied the adaptive overcomplete sparsifying transforms to image denoising, where they provide better performance over the synthesis K-SVD, while being faster. The overcomplete transforms also denoise better than square transforms. The promise of transform learning in other signal and image processing applications [33] merits further study.

6. REFERENCES

[1] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, 2007.

[2] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[3] D. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[4] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[5] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[6] E. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[7] S. Ravishankar and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE Trans. Med. Imag.*, vol. 30, no. 5, pp. 1028–1041, 2011.

[8] R. Rubinstein, T. Faktor, and M. Elad, "K-SVD dictionary-learning for the analysis sparse model," in *Proc. IEEE Int. Conf. Acoust. Speech, Sig. Proc.*, 2012, pp. 5405–5408.

[9] R. Rubinstein and M. Elad, "K-SVD dictionary-learning for analysis sparse models," in *Proc. SPARS11*, June 2011.

[10] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Noise aware analysis operator learning for approximately cosparse signals," in *Proc. IEEE Int. Conf. Acoust. Speech, Sig. Proc.*, 2012, pp. 5409–5412.

[11] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Trans. Signal Process.*, vol. 61, no. 5, pp. 1072–1086, 2013.

[12] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.

[13] E. J. Candès and D. L. Donoho, "Ridgelets: A key to higher-dimensional intermittency?," *Phil. Trans. R. Soc. Lond. A*, vol. 357, no. 1760, pp. 2495–2509, 1999.

[14] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, 2005.

[15] E. J. Candès and D. L. Donoho, "Curvelets - a surprisingly effective nonadaptive representation for objects with edges," in *Curves and Surfaces*, pp. 105–120. Vanderbilt University Press, 1999.

[16] W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard transform image coding," *Proc. IEEE*, vol. 57, no. 1, pp. 58–68, 1969.

[17] J. B. Allen and L. R. Rabiner, "A unified approach to short-time fourier analysis and synthesis," *Proc. IEEE*, vol. 65, no. 11, pp. 1558–1564, 1977.

[18] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[19] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[20] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2178–2191, 2009.

[21] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[22] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[23] H. Y. Liao and G. Sapiro, "Sparse representations for limited data tomography," in *Proc. IEEE International Symposium on Biomedical Imaging (ISBI)*, 2008, pp. 1375–1378.

[24] S. Ravishankar and Y. Bresler, "Multiscale dictionary learning for MRI," in *Proc. ISMRM*, 2011, p. 2830.

[25] G. Peyré and J. Fadili, "Learning analysis sparsity priors," in *Proc. of Sampta'11*, 2011.

[26] M. Yaghoobi, S. Nam, R. Gribonval, and M. Davies, "Analysis operator learning for overcomplete cosparse representations," in *European Signal Processing Conference (EUSIPCO)*, 2011.

[27] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms for image processing," in *IEEE Int. Conf. Image Process.*, 2012, pp. 681–684.

[28] J. Dattorro, *Convex Optimization & Euclidean Distance Geometry*, Meboo Publishing USA, 2005.

[29] M. Elad, "Optimized projections for compressed-sensing," *IEEE Trans. on Signal Processing*, vol. 55, no. 12, pp. 5695–5702, 2007.

[30] M. Elad, "Michael Elad personal page," `http://www.cs.technion.ac.il/~elad/Various/KSVD_Matlab_ToolBox.zip`, 2009.

[31] S. Ravishankar and Y. Bresler, "Learning doubly sparse transforms for images," *IEEE Trans. Image Process.*, 2012, submitted.

[32] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[33] S. Ravishankar and Y. Bresler, "Sparsifying transform learning for compressed sensing MRI," in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2013, to appear.