ℓ_0 Sparsifying Transform Learning with Efficient Optimal Updates and Convergence Guarantees

Saiprasad Ravishankar, *Student Member, IEEE,* and Yoram Bresler, *Fellow, IEEE*

***Abstract*—Many applications in signal processing benefit from the sparsity of signals in a certain transform domain or dictionary. Synthesis sparsifying dictionaries that are directly adapted to data have been popular in applications such as image denoising, inpainting, and medical image reconstruction. In this work, we focus instead on the sparsifying transform model, and study the learning of well-conditioned square sparsifying transforms. The proposed algorithms alternate between a ℓ_0 "norm"-based sparse coding step, and a non-convex transform update step. We derive the exact analytical solution for each of these steps. The proposed solution for the transform update step achieves the global minimum in that step, and also provides speedups over iterative solutions involving conjugate gradients. We establish that our alternating algorithms are globally convergent to the set of local minimizers of the non-convex transform learning problems. In practice, the algorithms are insensitive to initialization. We present results illustrating the promising performance and significant speed-ups of transform learning over synthesis K-SVD in image denoising.**

***Index Terms*—Transform model, Fast algorithms, Image representation, Sparse representation, Denoising, Dictionary learning, Non-convex.**

I. Introduction

The sparsity of signals and images in a certain transform domain or dictionary has been widely exploited in numerous applications in recent years. While transforms are a classical tool in signal processing, alternative models have also been studied for sparse representation of data, most notably the popular *synthesis model* [1], [2], the *analysis model* [1] and its more realistic extension, the *noisy signal analysis model* [3]. In this paper, we focus specifically on the sparsifying *transform model* [4], [5], which is a generalized analysis model, and suggests that a signal $y \in \mathbb{R}^n$ is *approximately sparsifiable* using a transform $W \in \mathbb{R}^{m \times n}$, that is $Wy = x + e$ where $x \in \mathbb{R}^m$ is sparse in some sense, and e is a small residual. A distinguishing feature is that, unlike the synthesis or noisy signal analysis models, where the residual is measured in the signal domain, in the transform model the residual is in the transform domain.

The transform model is not only more general in its modeling capabilities than the analysis models, it is also much more efficient and scalable than both the synthesis and noisy signal analysis models. We briefly review the main distinctions between these sparse models (cf. [5] for a more detailed review, and for the relevant references) in this and the following paragraphs. One key difference is in the process of finding a sparse representation for data given the model, or dictionary. For the transform model, given the signal y and transform W, the *transform sparse coding* problem [5] minimizes $\|Wy - x\|_2^2$ subject to $\|x\|_0 \leq s$, where s is a given sparsity level. The solution $\hat{x}$ is obtained exactly and cheaply by zeroing out all but the s coefficients of largest magnitude in Wy [1]. In contrast, for the synthesis or noisy analysis models, the process of *sparse coding* is NP-hard (Non-deterministic Polynomial-time hard) in general. While some of the approximate algorithms that have been proposed for synthesis or analysis sparse coding are guaranteed to provide the correct solution under certain conditions, in applications, especially those involving learning the models from data, these conditions are often violated. Moreover, the various synthesis and analysis sparse coding algorithms tend to be computationally expensive for large-scale problems.

Recently, the data-driven adaptation of sparse models has received much interest. The adaptation of synthesis dictionaries based on training signals [6]–[12] has been shown to be useful in various applications [13]–[15]. The learning of analysis dictionaries, employing either the analysis model or its noisy signal extension, has also received some recent attention [3], [16]–[18].

Focusing instead on the transform model, we recently developed the following formulation [5] for the learning of well-conditioned square sparsifying transforms. Given a matrix $Y \in \mathbb{R}^{n \times N}$, whose columns represent training signals, our formulation for learning a square sparsifying transform $W \in \mathbb{R}^{n \times n}$ for Y is [5]

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

where $\lambda > 0$, $\xi > 0$ are parameters, and $X \in \mathbb{R}^{n \times N}$ is a matrix, whose columns X_i are the sparse codes of the corresponding training signals Y_i. The term $\|WY - X\|_F^2$ in (P1) is called *sparsification error*, and denotes the deviation of the data in the transform domain from its sparse approximation (i.e., the deviation of WY from the sparse matrix X). Problem (P1) also has $v(W) \triangleq -\log |\det W| + \xi \|W\|_F^2$ as a regularizer in the objective to prevent trivial solutions.

This work was supported in part by the National Science Foundation (NSF) under grants CCF-1018660 and CCF-1320953.

S. Ravishankar and Y. Bresler are with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, IL, 61801 USA e-mail: (ravisha3, ybresler)@illinois.edu.

[1] Moreover, given W and sparse code x, we can also recover a least squares estimate of the underlying signal as $\hat{y} = W^\dagger x$, where $W^\dagger$ is the pseudo-inverse of W.

We have proposed an alternating minimization algorithm for solving (P1) [5], that alternates between updating X (sparse coding), and W (transform update), with the other variable kept fixed.

Because of the simplicity of sparse coding in the transform model, the alternating algorithm for transform learning [5] has a low computational cost. On the other hand, because, in the case of the synthesis or noisy signal analysis models, the learning formulations involve the NP-hard sparse coding, such learning formulations are also NP hard. Moreover, even when the ℓ_0 sparse coding in these problems is approximated by a convex relaxation, the learning problems remain highly non-convex. Because the approximate algorithms for these problems usually solve the sparse coding problem repeatedly in the iterative process of adapting the sparse model, the cost of sparse coding is multiplied manyfold. Hence, the synthesis or analysis dictionary learning algorithms tend to be computationally expensive in practice for large scale problems. Finally, popular algorithms for synthesis dictionary learning such as K-SVD [9], or algorithms for analysis dictionary learning do not have convergence guarantees.

In this follow-on work on transform learning, keeping the focus on the square transform learning formulation (P1), we make the following contributions.

- We derive highly *efficient closed-form solutions* for the update steps in the alternating minimization procedure for (P1), that further enhance the computational properties of transform learning.
- We also consider the minimization of an alternative version of (P1) in this paper, which is obtained by replacing the sparsity constraints with sparsity penalties.
- Importantly, we establish for the first time that our iterative algorithms for transform learning are globally convergent to the set of local minimizers of the non-convex transform learning problems.

We organize the rest of this paper as follows. Section II briefly describes our transform learning formulations. In Section III, we derive efficient algorithms for transform learning, and discuss the algorithms' computational cost. In Section IV, we present convergence guarantees for our algorithms. The proof of convergence is provided in the Appendix. Section V presents experimental results demonstrating the convergence behavior, and the computational efficiency of the proposed scheme. We also show brief results for the image denoising application. In Section VI, we conclude.

II. Learning Formulations and Properties

The transform learning Problem (P1) was introduced in Section I. Here, we discuss some of its important properties. The regularizer $v(W) \triangleq -\log|\det W| + \xi \|W\|_F^2$ helps prevent trivial solutions in (P1). The $\log|\det W|$ penalty eliminates degenerate solutions such as those with zero, or repeated rows. While it is sufficient to consider the $\det W > 0$ case [5], to simplify the algorithmic derivation we replace the positivity constraint by the absolute value in the formulation in this paper. The $\|W\|_F^2$ penalty in (P1) helps remove a 'scale ambiguity' in the solution, which occurs when the data admit an exactly sparse representation [5]. The $-\log|\det W|$ and $\|W\|_F^2$ penalties together additionally help control the condition number $\kappa(W)$ of the learnt transform. (Recall that the condition number of a matrix $A \in \mathbb{R}^{n\times n}$ is defined as $\kappa(A) = \beta_1/\beta_n$, where β_1 and β_n denote the largest and smallest singular values of A, respectively.) In particular, badly conditioned transforms typically convey little information and may degrade performance in applications such as signal/image representation, and denoising [5]. Well-conditioned transforms, on the other hand, have been shown to perform well in (sparse) image representation, and denoising [5], [19].

The condition number $\kappa(W)$ can be upper bounded by a monotonically increasing function of $v(W)$ (see Proposition 1 of [5]). Hence, minimizing $v(W)$ encourages reduction of the condition number. The regularizer $v(W)$ also penalizes bad scalings. Given a transform W and a scalar $\alpha \in \mathbb{R}$, $v(\alpha W) \to \infty$ as the scaling $\alpha \to 0$ or $\alpha \to \infty$. For a fixed ξ, as λ is increased in (P1), the optimal transform(s) become well-conditioned. In the limit $\lambda \to \infty$, their condition number tends to 1, and their spectral norm (or, scaling) tends to $1/\sqrt{2\xi}$. Specifically, for $\xi = 0.5$, as $\lambda \to \infty$, the optimal transform tends to an *orthonormal transform*. In practice, the transforms learnt via (P1) have condition numbers close to 1 even for finite λ [5]. The specific choice of λ depends on the application and desired condition number.

In this paper, to achieve invariance of the learned transform to trivial scaling of the training data Y, we set $\lambda = \lambda_0 \|Y\|_F^2$ in (P1), where $\lambda_0 > 0$ is a constant. Indeed, when the data Y are replaced with αY ($\alpha \in \mathbb{R}$, $\alpha \neq 0$) in (P1), we can set $X = \alpha X'$. Then, the objective function becomes $\alpha^2 \left(\|WY - X'\|_F^2 + \lambda_0 \|Y\|_F^2 \, v(W) \right)$, which is just a scaled version of the objective in (P1) (for un-scaled Y). Hence, its minimization over (W, X') (with X' constrained to have columns of sparsity $\leq s$) yields the same solution(s) as (P1). Thus, the learnt transform for data αY is the same as for Y, while the learnt sparse code for αY is α times that for Y.

We have shown [5] that the cost function in (P1) is lower bounded by λv_0, where $v_0 = \frac{n}{2} + \frac{n}{2}\log(2\xi)$. The minimum objective value in Problem (P1) equals this lower bound if and only if there exists a pair $(\hat{W}, \hat{X})$ such that $\hat{W}Y = \hat{X}$, with $\hat{X} \in \mathbb{R}^{n\times N}$ whose columns have sparsity $\leq s$, and $\hat{W} \in \mathbb{R}^{n\times n}$ whose singular values are all equal to $1/\sqrt{2\xi}$ (hence, the condition number $\kappa(\hat{W}) = 1$). Thus, when an "error-free" transform model exists for the data, and the underlying transform is unit conditioned, such a transform model is guaranteed to be a global minimizer of Problem (P1) (i.e., such a model is *identifiable* by solving (P1)). Therefore, it makes sense to solve (P1) to find such good models.

Another interesting property of Problem (P1) is that it admits an equivalence class of solutions/minimizers. Because the objective in (P1) is unitarily invariant, then given a minimizer $(\tilde{W}, \tilde{X})$, the pair $(\Theta\tilde{W}, \Theta\tilde{X})$ is another equivalent minimizer for all *sparsity-preserving orthonormal matrices* Θ, i.e., Θ such that $\|\Theta\tilde{X}_i\|_0 \leq s \ \forall \ i$. For example, Θ can be a row permutation matrix, or a diagonal ± 1 sign matrix.

We note that a cost function similar to that in (P1), but lacking the $\|W\|_F^2$ penalty has been derived under certain

assumptions in the very different setting of blind source separation [20]. However, the transforms learnt via Problem (P1) perform poorly [5] in signal processing applications, when the learning is done excluding the crucial $\|W\|_F^2$ penalty, which as discussed, helps overcome the scale ambiguity and control the condition number.

In this work, we also consider an alternative version of Problem (P1) by replacing the ℓ_0 sparsity constraints with ℓ_0 penalties in the objective (this version of the transform learning problem has been recently used for example in adaptive tomographic reconstruction [21], [22]). In this case, we obtain the following unconstrained (or, sparsity penalized) transform learning problem

$$\text{(P2)} \quad \min_{W,X} \|WY - X\|_F^2 + \lambda\, v(W) + \sum_{i=1}^{N} \eta_i^2 \|X_i\|_0$$

where η_i^2, with $\eta_i > 0 \; \forall\, i$, denote the weights (e.g., $\eta_i = \eta \; \forall\, i$ for some η) for the sparsity penalties. The various aforementioned properties for Problem (P1) can be easily extended to the case of the alternative Problem (P2).

III. Transform Learning Algorithm

A. Algorithm

We have previously proposed [5] an alternating algorithm for solving (P1) that alternates between solving for X (*sparse coding step*) and W (*transform update step*), with the other variable kept fixed. While the sparse coding step has an exact solution, the transform update step was performed using iterative nonlinear conjugate gradients (NLCG). This alternating algorithm for transform learning has a low computational cost compared to synthesis or analysis dictionary learning. In the following, we provide a further improvement: we show that both steps of transform learning (for either (P1) or (P2)) can in fact, be performed exactly and cheaply.

1) Sparse Coding Step: The sparse coding step in the alternating algorithm for (P1) is as follows [5]

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

The above problem is to project WY onto the (non-convex) set of matrices whose columns have sparsity $\leq s$. Due to the additivity of the objective, this corresponds to projecting each column of WY onto the set of sparse vectors $\{x \in \mathbb{R}^n : \|x\|_0 \leq s\}$, which we call the s-ℓ_0 ball. Now, for a vector $z \in \mathbb{R}^n$, the optimal projection $\hat{z}$ onto the s-ℓ_0 ball is computed by zeroing out all but the s coefficients of largest magnitude in z. If there is more than one choice for the s coefficients of largest magnitude in z (can occur when multiple entries in z have identical magnitude), then the optimal $\hat{z}$ is not unique. We then choose $\hat{z} = H_s(z)$, where $H_s(z)$ is the projection, for which the indices of the s largest magnitude elements (in z) are the lowest possible. Hence, an optimal sparse code in (1) is computed as $\hat{X}_i = H_s(WY_i) \; \forall\, i$.

In the case of Problem (P2), we solve the following sparse coding problem

$$\min_{X} \|WY - X\|_F^2 + \sum_{i=1}^{N} \eta_i^2 \|X_i\|_0 \tag{2}$$

A solution $\hat{X}$ of (2) in this case is obtained as $\hat{X}_i = \hat{H}^1_{\eta_i}(WY_i)$ $\forall\, i$, where the (hard-thresholding) operator $\hat{H}^1_\eta(\cdot)$ is defined as follows.

$$\left(\hat{H}^1_\eta(b)\right)_j = \begin{cases} 0 & , \text{if } \; |b_j| < \eta \\ b_j & , \text{if } \; |b_j| \geq \eta \end{cases} \tag{3}$$

Here, $b \in \mathbb{R}^n$, and the subscript j indexes vector entries. This form of the solution to (2) has been mentioned in prior work [21]. For completeness, we include a brief proof in Appendix A. When the condition $|(WY)_{ji}| = \eta_i$ occurs for some i and j (where $(WY)_{ji}$ is the element of WY on the j^{th} row and i^{th} column), the corresponding optimal $\hat{X}_{ji}$ in (2) can be either $(WY)_{ji}$ or 0 (both of which correspond to the minimum value of the cost in (2)). The definition in (3) breaks the tie between these equally valid solutions by selecting the first. Thus, similar to Problem (1), the solution to (2) can be computed exactly.

2) Transform Update Step: The transform update step of (P1) or (P2) involves the following unconstrained non-convex [5] minimization.

$$\min_{W} \; \|WY - X\|_F^2 + \lambda\xi \|W\|_F^2 - \lambda \log |\det W| \tag{4}$$

Note that although NLCG works well for the transform update step [5], convergence to the global minimum of the non-convex transform update step has not been proved with NLCG. Instead, replacing NLCG, the following proposition provides the closed-form solution for Problem (4). The solution is written in terms of an appropriate singular value decomposition (SVD). We use $(\cdot)^T$ to denote the matrix transpose operation, and $M^{\frac{1}{2}}$ to denote the positive definite square root of a positive definite matrix M. We let I denote the $n \times n$ identity matrix.

Proposition 1: Given the training data $Y \in \mathbb{R}^{n \times N}$, sparse code matrix $X \in \mathbb{R}^{n \times N}$, and $\lambda > 0$, $\xi > 0$, consider the factorization $YY^T + \lambda\xi I = LL^T$, with $L \in \mathbb{R}^{n \times n}$, and full SVD $L^{-1}YX^T = Q\Sigma R^T$. Then, a global minimizer for the transform update step (4) can be written as

$$\hat{W} = 0.5R\left(\Sigma + \left(\Sigma^2 + 2\lambda I\right)^{\frac{1}{2}}\right) Q^T L^{-1} \tag{5}$$

The solution is unique if and only if YX^T is non-singular. Furthermore, the solution is invariant to the choice of factor L.

Proof: The objective function in (4) can be re-written as $tr\left\{W\left(YY^T + \lambda\xi I\right)W^T\right\} - 2\,tr(WYX^T) + tr(XX^T)$ $-\lambda \log |\det W|$. We then decompose the positive-definite matrix $YY^T + \lambda\xi I$ as LL^T (e.g., L can be the positive-definite square root, or the cholesky factor of $YY^T + \lambda\xi I$). The objective function then simplifies as follows

$$tr\left(WLL^TW^T - 2WYX^T + XX^T\right) - \lambda \log |\det W|$$

Using a change of variables $B = WL$, the multiplicativity of the determinant implies $\log |\det B| = \log |\det W| + \log |\det L|$. Problem (4) is then equivalent to

$$\min_{B} \; tr\left(BB^T\right) - 2tr\left(BL^{-1}YX^T\right) - \lambda \log |\det B| \tag{6}$$

Next, let $B = U\Gamma V^T$, and $L^{-1}YX^T = Q\Sigma R^T$ be full SVDs ($U, \Gamma, V, Q, \Sigma, R$ are all $n \times n$ matrices), with γ_i and

σ_i denoting the diagonal entries of Γ and Σ, respectively. The unconstrained minimization (6) then becomes

$$\min_{\Gamma} \left[tr\left(\Gamma^2\right) - 2 \max_{U,V} \left\{ tr\left(U\Gamma V^T Q \Sigma R^T\right)\right\} - \lambda \sum_{i=1}^{n} \log\, \gamma_i \right]$$

For the inner maximization, we use the result $\max_{U,V}\, tr\left(U\Gamma V^T Q\Sigma R^T\right) = tr\left(\Gamma\Sigma\right)$ [23], where the maximum is attained by setting $U = R$ and $V = Q$. The remaining minimization with respect to Γ is then

$$\min_{\{\gamma_i\}} \sum_{i=1}^{n} \gamma_i^2 - 2\sum_{i=1}^{n} \gamma_i \sigma_i - \lambda \sum_{i=1}^{n} \log\, \gamma_i \tag{7}$$

This problem is convex in the non-negative singular values γ_i, and the solution is obtained by setting the derivative of the cost in (7) with respect to the γ_i's to 0. This gives $\gamma_i = 0.5\left(\sigma_i \pm \sqrt{\sigma_i^2 + 2\lambda}\right)\ \forall\, i$. Since all the $\gamma_i \geq 0$, the only feasible solution is

$$\gamma_i = \frac{\sigma_i + \sqrt{\sigma_i^2 + 2\lambda}}{2}\ \forall\, i \tag{8}$$

Thus, a closed-form solution or global minimizer for the transform update step (4) is given as in (5).

The solution (5) is invariant to the specific choice of the matrix L. To show this, we use the easy result that if $L \in \mathbb{R}^{n\times n}$ and $\tilde{L} \in \mathbb{R}^{n\times n}$ are two (different) invertible matrices satisfying $YY^T + \lambda\xi I = LL^T = \tilde{L}\tilde{L}^T$, then $\tilde{L} = LG$, where G is an orthonormal matrix satisfying $GG^T = I$ [2]. Consider L and $\tilde{L}$ as defined above. Now, if Q is the left singular matrix corresponding to $L^{-1}YX^T$, then $\tilde{Q} = G^TQ$ is a corresponding left singular matrix for $\tilde{L}^{-1}YX^T = G^TL^{-1}YX^T$. Therefore, replacing L by $\tilde{L}$ in (5), making the substitutions $\tilde{L}^{-1} = G^TL^{-1}$, $\tilde{Q}^T = Q^TG$, and using the orthonormality of G, it is obvious that the closed-form solution (5) involving $\tilde{L}$ is identical to that involving L.

Finally, we show that the solution (5) is unique if and only if YX^T is non-singular (or, equivalently $L^{-1}YX^T$ is non-singular). Owing to the invariance of the solution to the choice of factor L, *it suffices to show that* for any specific choice of factor L, the solution (5) is unique or invariant to the different choices for the full SVD of $L^{-1}YX^T$ if and only if $L^{-1}YX^T$ is non-singular. Fixing the choice for the factor L, note that the solution (5) can be written using the notations introduced above as $\hat{W} = \left(\sum_{i=1}^{n} \gamma_i R_i Q_i^T\right) L^{-1}$, where R_i and Q_i are the i$^{\text{th}}$ columns of R and Q, respectively.

Suppose that $L^{-1}YX^T$ has rank $< n$. Then a singular vector pair $\left(Q_k, R_k\right)$ of $L^{-1}YX^T$ corresponding to a zero singular value $\sigma_k = 0$ can also be modified as $\left(Q_k, -R_k\right)$ or $\left(-Q_k, R_k\right)$, yielding equally valid alternative SVDs of $L^{-1}YX^T$. However, because by (8), zero singular values in the matrix Σ are mapped to non-zero singular values in the matrix Γ, we have that the following two matrices are equally valid solutions to (4) in this case.

$$\hat{W}^a = \left(\sum_{i\neq k} \gamma_i R_i Q_i^T + \gamma_k R_k Q_k^T\right) L^{-1} \tag{9}$$

$$\hat{W}^b = \left(\sum_{i\neq k} \gamma_i R_i Q_i^T - \gamma_k R_k Q_k^T\right) L^{-1} \tag{10}$$

where $\gamma_k > 0$. It is obvious that $\hat{W}^a \neq \hat{W}^b$, i.e., the optimal transform is not unique in this case. Therefore, $L^{-1}YX^T$ being non-singular is necessary for the uniqueness of (5).

Next, suppose $L^{-1}YX^T$ is nonsingular. We show that this implies the aforementioned solution invariance. First, if the singular values of $L^{-1}YX^T$ are non-degenerate (distinct and non-zero), then the SVD of $L^{-1}YX^T$ is unique up to joint scaling of any pair $\left(Q_i, R_i\right)$ by ± 1. This immediately implies that the solution $\hat{W} = \left(\sum_{i=1}^{n} \gamma_i R_i Q_i^T\right) L^{-1}$ is invariant to the different choices of R and Q in this case. On the other hand, if $L^{-1}YX^T$ has some repeated but still non-zero singular values, then, by (8), they are mapped to repeated (and non-zero) singular values in Γ. Let us assume that Σ has only one singular value that repeats (the proof easily extends to the case of multiple repeated singular values) say r times, and that these repeated values are arranged in the bottom half of the matrix Σ (i.e., $\sigma_{n-r+1} = \sigma_{n-r+2} = ... = \sigma_n = \hat{\sigma} > 0$). Then, we have

$$L^{-1}YX^T = \sum_{i=1}^{n-r} \sigma_i Q_i R_i^T + \hat{\sigma}\left(\sum_{i=n-r+1}^{n} Q_i R_i^T\right) \tag{11}$$

Because the matrix defined by the first sum on the right (in (11)) corresponding to distinct singular values is unique [3], and $\hat{\sigma} > 0$, so too is the second matrix defined by the second sum. (This is also a simple consequence of the fact that although the singular vectors associated with repeated singular values are not unique, the subspaces spanned by them are [24].) The transform update solution (5) in this case is given as

$$\hat{W} = \left\{\sum_{i=1}^{n-r} \gamma_i R_i Q_i^T + \gamma_{n-r+1}\left(\sum_{i=n-r+1}^{n} R_i Q_i^T\right)\right\} L^{-1} \tag{12}$$

Based on the preceding arguments, it is clear that the right hand side of (12) is invariant to the particular choice of (non-unique) Q and R. Thus, the transform update solution (5) is invariant to different alternative choices for R and Q even when $L^{-1}YX^T$ has possibly repeated, but non-zero singular values. Therefore, the non-singularity of $L^{-1}YX^T$ is also a sufficient condition for the invariance of (5) to different alternative choices for the full SVD of $L^{-1}YX^T$. ■

The transform update solution (5) is expressed in terms of the full SVD of $L^{-1}YX^T$, where L is for example, the Cholesky factor, or alternatively, the Eigenvalue Decomposition (EVD) square root of $YY^T + \lambda\xi I$. Although in practice the SVD, or even the square root of non-negative scalars, are computed using iterative methods, we will assume in the theoretical analysis in this paper, that the solution (5) is computed exactly. In practice, standard numerical methods are guaranteed to quickly provide machine precision accuracy for the SVD and other computations. Therefore, the solution (5) is computed to within machine precision accuracy in practice.

[2] Since L and $\tilde{L}$ are both non-singular matrices (being square roots of the positive definite matrix $YY^T + \lambda\xi I$), we have $\tilde{L} = L\left(L^{-1}\tilde{L}\right) = LG$, with $G \triangleq L^{-1}\tilde{L}$ a non-singular matrix. Moreover, since $LL^T - \tilde{L}\tilde{L}^T = 0$, we have $L\left(I - GG^T\right)L^T = 0$. Because L is invertible, we therefore have that $GG^T = I$ in the preceding equation. Therefore, G is an orthonormal matrix satisfying $GG^T = I$.

[3] That matrix is invariant to joint scaling of any pair $\left(Q_i, R_i\right)$, for $1 \leq i \leq n - r$, by ± 1.

Transform Learning Algorithms A1 and A2

Input : $Y \in \mathbb{R}^{n \times N}$ - training data, s - sparsity, λ - constant, ξ - constant, η_i for $1 \leq i \leq N$ - constants, J_0 - number of iterations.

Output : $\hat{W}$ - learned transform, $\hat{X}$ - learned sparse code matrix.

Initial Estimates: $(\hat{W}^0, \hat{X}^0)$.

Pre-Compute: $L^{-1} = \left(YY^T + \lambda\xi I\right)^{-1/2}$.

For k = 1: J_0 Repeat

1) Compute full SVD of $L^{-1}Y(\hat{X}^{k-1})^T$ as $Q\Sigma R^T$.
2) $\hat{W}^k = 0.5R\left(\Sigma + \left(\Sigma^2 + 2\lambda I\right)^{\frac{1}{2}}\right)Q^T L^{-1}$.
3) $\hat{X}_i^k = H_s(\hat{W}^k Y_i)\ \forall\, i$ for Algorithm A1, or $\hat{X}_i^k = \hat{H}^1_{\eta_i}(\hat{W}^k Y_i)\ \forall\, i$ for Algorithm A2.

End

Fig. 1. Algorithms A1 and A2 for solving Problems (P1) and (P2), respectively. Superscript of k denotes the iterates in the algorithms. Although we begin with the transform update step in each iteration above, one could alternatively start with the sparse coding step as well.

Algorithms A1 and A2 for (P1) and (P2) respectively, for transform learning are shown in Fig. 1. The algorithms assume that an initial estimate $(\hat{W}^0, \hat{X}^0)$ for the variables is provided. The initial $\hat{W}^0$ is only used by the algorithms in a degenerate scenario mentioned later (see footnote 13).

While Proposition 1 provides the closed-form solution to (4) for real-valued matrices, the solution can be extended to the complex-valued case (useful in applications such as magnetic resonance imaging (MRI) [15]) by replacing the $(\cdot)^T$ operation in Proposition 1 and its proof by $(\cdot)^H$, the Hermitian transpose operation. The same proof applies, with the trace maximization result for the real case replaced by $\max_{U,V}\ Re\left\{tr\left(U\Gamma V^H Q\Sigma R^H\right)\right\} = tr\left(\Gamma\Sigma\right)$ for the complex case, where $Re(A)$ denotes the real part of scalar A.

B. The Orthonormal Transform Limit

We have seen that for $\xi = 0.5$, as $\lambda \to \infty$, the W minimizing (P1) tends to an orthonormal matrix. Here, we study the behavior of the actual sparse coding and transform update steps of our algorithm as the parameter λ (or, equivalently λ_0, since $\lambda = \lambda_0 \left\|Y\right\|_F^2$) tends to infinity. The following Proposition 2 establishes that as $\lambda \to \infty$ with ξ held at 0.5, the sparse coding and transform update solutions for (P1) approach the corresponding solutions for an orthonormal transform learning problem. Although we consider Problem (P1) here, a similar result also holds with respect to (P2).

Proposition 2: For $\xi = 0.5$, as $\lambda \to \infty$, the sparse coding and transform update solutions in (P1) coincide with the corresponding solutions obtained by employing alternating minimization on the following orthonormal transform learning problem.

$$\min_{W,X} \left\|WY - X\right\|_F^2 \ \ s.t.\ W^T W = I,\ \left\|X_i\right\|_0 \leq s\ \forall\ i \quad (13)$$

Specifically, the sparse coding step for Problem (13) involves

$$\min_{X} \left\|WY - X\right\|_F^2 \ \ s.t.\ \ \left\|X_i\right\|_0 \leq s\ \forall\ i$$

and the solution is $\hat{X}_i = H_s(WY_i)\ \forall\, i$. Moreover, the transform update step for Problem (13) involves

$$\max_{W} tr\left(WYX^T\right) \ \ s.t.\ W^T W = I \quad (14)$$

Denoting the full SVD of YX^T by $U\Sigma V^T$, where $U \in \mathbb{R}^{n\times n}$, $\Sigma \in \mathbb{R}^{n\times n}$, $V \in \mathbb{R}^{n\times n}$, the optimal solution in Problem (14) is $\hat{W} = VU^T$. This solution is unique if and only if YX^T is non-singular.

Proof: See Appendix B. ■

For $\xi \neq 0.5$, Proposition 2 holds with the constraint $W^T W = I$ in Problem (13) replaced by the constraint $W^T W = (1/2\xi)I$. The transform update solution for Problem (13) with the modified constraint $W^T W = (1/2\xi)I$ is the same as mentioned in Proposition 2, except for an additional scaling of $1/\sqrt{2\xi}$.

The orthonormal transform case is special, in that Problem (13) is also an orthonormal synthesis dictionary learning problem, with W^T denoting the synthesis dictionary. This follows immediately, using the identity $\left\|WY - X\right\|_F = \left\|Y - W^T X\right\|_F$, for orthonormal W. Hence, Proposition 2 provides an alternating algorithm with optimal updates not only for the orthonormal transform learning problem, but at the same time for the orthonormal dictionary learning problem.

C. Computational Cost

The proposed transform learning algorithms A1 and A2 alternate between the sparse coding and transform update steps. Each of these steps has a closed-form solution. We now discuss their computational costs. We assume that the matrices $YY^T + \lambda\xi I$ and L^{-1} (used in (5)) are pre-computed (at the beginning of the algorithm) at total costs of $O(Nn^2)$ and $O(n^3)$, respectively, for the entire algorithm.

The computational cost of the sparse coding step in both Algorithms A1 and A2 is dominated by the computation of the product WY, and therefore scales as $O(Nn^2)$. In contrast, the projection onto the s-ℓ_0 ball in Algorithm A1 requires only $O(nN\log n)$ operations, when employing sorting [5], and the hard thresholding (as in equation (3)) in Algorithm A2 requires only $O(nN)$ comparisons.

For the transform update step, the computation of the product YX^T requires αNn^2 multiply-add operations for an X with s-sparse columns, and $s = \alpha n$. Then, the computation of $L^{-1}YX^T$, its SVD, and the closed-form transform update (5) require $O(n^3)$ operations. On the other hand, when NLCG is employed for transform update, the cost (excluding the YX^T pre-computation) scales as $O(Jn^3)$, where J is the number of NLCG iterations [5]. Thus, compared to NLCG, the proposed update formula (5) allows for both an exact and, depending on J, potentially cheaper solution to the transform update step.

Under the assumption that $n \ll N$, the total cost per iteration (of sparse coding and transform update) of the proposed algorithms scales as $O(Nn^2)$. This is much lower than the per-iteration cost of learning an $n \times K$ overcomplete ($K > n$) synthesis dictionary D using K-SVD [9], which scales (assuming that the synthesis sparsity level $s \propto n$) as $O(KNn^2)$. Our transform learning schemes also hold a

similar computational advantage [5] over analysis dictionary learning schemes such as analysis K-SVD.

As illustrated in Section V-B, our algorithms converge in few iterations in practice. Therefore, the per-iteration computational advantages (e.g., over K-SVD) also typically translate to a net computational advantage in practice (e.g., in denoising).

IV. Main Convergence Results

A. Result for Problem (P1)

Problem (P1) has the constraint $\|X_i\|_0 \leq s \, \forall \, i$, which can instead be added as a penalty in the objective by using a barrier function $\psi(X)$ (which takes the value $+\infty$ when the constraint is violated, and is zero otherwise). In this form, Problem (P1) is unconstrained, and we denote its objective as $g(W,X) = \|WY - X\|_F^2 + \lambda \xi \|W\|_F^2 - \lambda \log |\det W| + \psi(X)$. The unconstrained minimization problem involving the objective $g(W,X)$ is exactly equivalent to the constrained formulation (P1) in the sense that the minimum objective values as well as the set of minimizers of the two formulations are identical. To see this, note that whenever the constraint $\|X_i\|_0 \leq s \, \forall \, i$ is satisfied, the two objectives coincide. Otherwise, the objective in the unconstrained formulation takes the value $+\infty$ and therefore, its minimum value is achieved where the constraint $\|X_i\|_0 \leq s \, \forall \, i$ holds. This minimum value (and the corresponding set of minimizers) is therefore the same as that for the constrained formulation (P1). The proposed Algorithm A1 is an exact alternating algorithm for both the constrained and unconstrained formulations above.

Problem (P1) is to find the best possible transform model for the given training data Y by minimizing the sparsification error, and controlling the condition number (avoiding triviality). We are interested to know whether the proposed alternating algorithm converges to a minimizer of (P1), or whether it could get stuck in saddle points, or some non-stationary points. Problem (P1) is highly non-convex, and therefore well-known results on convergence of alternating minimization (e.g., [25]) do not apply here. The following Theorem 1 provides the convergence of our Algorithm A1 for (P1). We say that a sequence $\left\{a^k\right\}$ has an accumulation point a, if there is a subsequence that converges to a. For a vector h, we let $\phi_j(h)$ denote the magnitude of the j^{th} largest element (magnitude-wise) of h. For some matrix B, $\|B\|_\infty \triangleq \max_{i,j} |B_{ij}|$.

Theorem 1: Let $\left\{W^k, X^k\right\}$ denote the iterate sequence generated by Algorithm A1 with training data Y and initial (W^0, X^0). Then, the objective sequence $\left\{g(W^k, X^k)\right\}$ is monotone decreasing, and converges to a finite value, say $g^* = g^*\left(W^0, X^0\right)$. Moreover, the iterate sequence is bounded, and each accumulation point (W,X) of the iterate sequence is a fixed point of the algorithm, and a local minimizer of the objective g in the following sense. For each accumulation point (W,X), there exists $\epsilon' = \epsilon'(W) > 0$ such that

$$g(W + dW, X + \Delta X) \geq g(W,X) = g^* \qquad (15)$$

holds for all $dW \in \mathbb{R}^{n \times n}$ satisfying $\|dW\|_F \leq \epsilon'$, and all $\Delta X \in \mathbb{R}^{n \times N}$ in the union of the following regions.

R1. The half-space $tr\left\{(WY - X)\Delta X^T\right\} \leq 0$.

R2. The local region defined by $\|\Delta X\|_\infty < \min_i \{\phi_s(WY_i) : \|WY_i\|_0 > s\}$.

Furthermore, if we have $\|WY_i\|_0 \leq s \, \forall \, i$, then ΔX can be arbitrary.

The notation $g^*\left(W^0, X^0\right)$ in Theorem 1 represents the value to which the objective sequence $\left\{g(W^k, X^k)\right\}$ converges, starting from an initial (estimate) (W^0, X^0). Local region R2 in Theorem 1 is defined in terms of the scalar $\min_i \{\phi_s(WY_i) : \|WY_i\|_0 > s\}$, which is computed by taking the columns of WY with sparsity greater than s, and finding the s-largest magnitude element in each of these columns, and choosing the smallest of those magnitudes. The intuition for this particular construction of the local region is provided in the proof of Lemma 9 in Appendix D.

Theorem 1 indicates local convergence of our alternating Algorithm A1. Assuming a particular initial (W^0, X^0), we have that every accumulation point (W,X) of the iterate sequence is a local optimum by equation (15), and satisfies $g(W,X) = g^*\left(W^0, X^0\right)$. Thus, all accumulation points of the iterates (for a particular initial (W^0, X^0)) are equivalent (in terms of their cost), or are equally good local minima. We thus have the following corollary.

Corollary 1: For Algorithm A1, assuming a particular initial (W^0, X^0), the objective converges to a local minimum, and the iterates converge to an equivalence class of local minimizers.

The local optimality condition (15) holds for the algorithm irrespective of initialization. However, the local minimum $g^*\left(W^0, X^0\right)$ that the objective converges to may possibly depend on (i.e., vary with) initialization. Nonetheless, empirical evidence presented in Section V suggests that the proposed transform learning scheme is insensitive to initialization. This leads us to conjecture that our algorithm could potentially converge to the global minimizer(s) of the learning problem in some (practical) scenarios. Fig. 2 provides a simple illustration of the convergence behavior of our algorithm. We also have the following corollary of Theorem 1, where 'globally convergent' refers to convergence from any initialization.

Corollary 2: Algorithm A1 is globally convergent to the set of local minimizers of the non-convex transform learning objective $g(W,X)$.

Note that our convergence result for the proposed non-convex learning algorithm, is free of any extra conditions or requirements. This is in clear distinction to algorithms such as IHT [26], [27] that solve non-convex problems, but require extra stringent conditions (e.g., tight conditions on restricted isometry constants of certain matrices) for their convergence results to hold. Theorem 1 also holds for any choice of the parameter λ_0 (or, equivalently λ) in (P1), that controls the condition number.

The optimality condition (15) in Theorem 1 holds true not only for local (small) perturbations in X, but also for arbitrarily large perturbations of X in a half space. For a particular accumulation point (W,X), the condition $tr\left\{(WY - X)\Delta X^T\right\} \leq 0$ in Theorem 1 defines a half-space of permissible perturbations in $\mathbb{R}^{n \times N}$. Now, even among the perturbations outside this half-space, i.e., ΔX satisfying

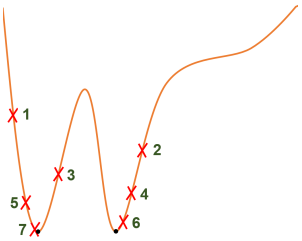


Fig. 2. Possible behavior of the algorithm near two hypothetical local minima (marked with black dots) of the objective. The numbered iterate sequence here has two subsequences (one even numbered, and one odd numbered) that converge to the two equally good (i.e., corresponding to the same value of the objective) local minima.

$tr\left\{(WY-X)\Delta X^T\right\} > 0$ (and also outside the local region R2 in Theorem 1), we only need to be concerned about the perturbations that maintain the sparsity level, i.e., ΔX such that $X+\Delta X$ has sparsity $\leq s$ per column. For any other ΔX, $g(W+dW, X+\Delta X) = +\infty > g(W,X)$ trivially. Now, since $X+\Delta X$ needs to have sparsity $\leq s$ per column, ΔX itself can be at most $2s$ sparse per column. Therefore, the condition $tr\left\{(WY-X)\Delta X^T\right\} > 0$ (corresponding to perturbations that could violate (15)) essentially corresponds to a union of low dimensional half-spaces (each corresponding to a different possible choice of support of ΔX). In other words, the set of "bad" perturbations is vanishingly small in $\mathbb{R}^{n\times N}$.

Note that Problem (P1) can be directly used for adaptive sparse representation (compression) of images [5], [28], in which case the convergence results here are directly applicable. (P1) can also be used in applications such as blind denoising [19], and blind compressed sensing [29]. The overall problem formulations [19], [29] in these applications are highly non-convex (see Section V-D). However, the problems are solved using alternating optimization [19], [29], and the transform learning Problem (P1) arises as a sub-problem. Therefore, by using the proposed learning scheme, the transform learning step of the alternating algorithms for denoising/compressed sensing can be guaranteed (by Theorem 1) to converge [4].

B. Result for Penalized Problem (P2)

When the sparsity constraints in (P1) are replaced with ℓ_0 penalties in the objective with weights η_i^2, we obtain the unconstrained transform learning problem (P2) with objective $u(W,X) = \|WY-X\|_F^2 + \lambda\xi\|W\|_F^2 - \lambda\log|\det W| + \sum_{i=1}^{N}\eta_i^2\|X_i\|_0$. In this case too, we have a convergence guarantee (similar to Theorem 1) for Algorithm A2 that minimizes $u(W,X)$.

Theorem 2: Let $\left\{W^k, X^k\right\}$ denote the iterate sequence generated by Algorithm A2 with training data Y and initial (W^0, X^0). Then, the objective sequence $\left\{u(W^k,X^k)\right\}$ is monotone decreasing, and converges to a finite value, say $u^* = u^*\left(W^0,X^0\right)$. Moreover, the iterate sequence is bounded, and each accumulation point (W,X) of the iterate sequence is a fixed point of the algorithm, and a local minimizer of the objective u in the following sense. For each accumulation point (W,X), there exists $\epsilon' = \epsilon'(W) > 0$ such that

$$u(W+dW, X+\Delta X) \geq u(W,X) = u^* \qquad (16)$$

holds for all $dW \in \mathbb{R}^{n\times n}$ satisfying $\|dW\|_F \leq \epsilon'$, and all $\Delta X \in \mathbb{R}^{n\times N}$ satisfying $\|\Delta X\|_\infty < \min_i\{\eta_i/2\}$.

The proofs of Theorems 1 and 2 are provided in Appendices D and F. Owing to Theorem 2, results analogous to Corollaries 1 and 2 apply.

Corollary 3: Corollaries 1 and 2 apply to Algorithm A2 and the corresponding objective $u(W,X)$ as well.

V. Experiments

A. Framework

In this section, we present results demonstrating the properties of our proposed transform learning Algorithm A1 for (P1), and its usefulness in applications. (Applications of Algorithm A2 for (P2) are demonstrated elsewhere [21], [22].) First, we illustrate the convergence behavior of our alternating learning algorithm for (P1). We consider various initializations for transform learning and investigate whether the proposed algorithm is sensitive to initializations. This study will provide some (limited) empirical understanding of local/global convergence behavior of the algorithm. Then, we compare our proposed algorithm to the NLCG-based transform learning algorithm [5] at various patch sizes, in terms of image representation quality and computational cost of learning. Finally, we briefly discuss the usefulness of the proposed scheme in image denoising.

All our implementations were coded in Matlab version R2013a. All computations were performed with an Intel Core i5 CPU at 2.5GHz and 4GB memory, employing a 64-bit Windows 7 operating system.

The data in our experiments are generated as the 2D patches of natural images. We use our transform learning Problem (P1) to learn adaptive sparse representations of such image patches. The means of the patches are removed and we only sparsify the mean-subtracted patches which are stacked as columns of the training matrix Y (patches reshaped as vectors) in (P1). The means are added back for image display. Mean removal is typically adopted in image processing applications such as compression and denoising [19], [30]. Similar to prior work [5], [19], the weight $\xi = 1$ in all our experiments.

We have previously introduced several metrics to evaluate the quality of learnt transforms [5], [19]. The *normalized sparsification error* (NSE) for a transform W is defined as $\|WY-X\|_F^2 / \|WY\|_F^2$, where Y is the data matrix, and the columns $X_i = H_s(WY_i)$ of the matrix X denote the sparse codes [5]. The NSE measures the fraction of energy lost in sparse fitting in the transform domain, and is an interesting property to observe for the learnt transforms. A useful performance metric for learnt transforms in image representation is the recovery peak signal to noise ratio (or *recovery PSNR*), which was previously defined as $255\sqrt{P}/\left\|Y-W^{-1}X\right\|_F$ in decibels (dB), where P is the number of image pixels and X is again the transform sparse code of data Y [5]. The recovery PSNR measures the error in

[4] Even when different columns of X are required to have different sparsity levels in (P1), our learning algorithm and Theorem 1 can be trivially modified to guarantee convergence.

recovering the patches Y (or equivalently the image, in the case of non-overlapping patches) as $W^{-1}X$ from their sparse codes X. The recovery PSNR serves as a simple surrogate for the performance of the learnt transform in compression. Note that if the proposed approach were to be used for compression, then W too would have to be transmitted as side information.

B. Convergence Behavior

Here, we study the convergence behavior of the proposed transform learning Algorithm A1. We extract the 8×8 ($n = 64$) non-overlapping (mean-subtracted) patches of the 512×512 image Barbara [9]. Problem (P1) is solved to learn a square transform W that is adapted to this data. The data matrix Y in this case has $N = 4096$ training signals (patches represented as vectors). The parameters are set as $s = 11$, $\lambda_0 = 3.1 \times 10^{-3}$. The choice of λ_0 here ensures well-conditioning of the learnt transform. Badly conditioned transforms degrade performance in applications [5], [19]. Hence, we focus our investigation here only on the well-conditioned scenario.

We study the convergence behavior of Algorithm A1 for various initializations of W. Once W is initialized, the algorithm iterates over the sparse coding and transform update steps (this corresponds to a different ordering of the steps in Fig. 1). We consider four different initializations (initial transforms) for W. The first is the 64×64 2D DCT matrix (obtained as the Kronecker product of two 8×8 1D DCT matrices). The second initialization is the Karhunen-Loève Transform (KLT) (i.e., the inverse of PCA), obtained here by inverting/transposing the left singular matrix of Y [5]. The third and fourth initializations are the identity matrix, and a random matrix with i.i.d. Gaussian entries (zero mean and standard deviation 0.2), respectively.

Figure 3 shows the progress of the algorithm over iterations for the various initializations of W. The objective function (Fig. 3(a)), sparsification error (Fig. 3(b)), and condition number (Fig. 3(c)), all converge quickly for our algorithm. The sparsification error decreases over the iterations, as required. Importantly, the final values of the objective (similarly, the sparsification error, and condition number) are nearly identical for all the initializations. This indicates that our learning algorithm is reasonably robust, or insensitive to initialization. Good initializations for W such as the DCT and KLT lead to faster convergence of learning. The learnt transforms also have identical Frobenius norms (5.14) for all the initializations.

Figure 3(d) shows the (well-conditioned) transform learnt with the DCT initialization. Each row of the learnt W is displayed as an 8×8 patch, called the transform atom. The atoms here exhibit frequency and texture-like structures that sparsify the patches of Barbara. Similar to our prior work [5], we observed that the transforms learnt with different initializations, although essentially equivalent in the sense that they produce similar sparsification errors and are similarly scaled and conditioned, appear somewhat different (i.e., they are not related by only row permutations and sign changes).

[5]We did not remove the means of the rows of Y here. However, we obtain almost identical plots in Fig. 3, when the learning algorithm is instead initialized with the KLT computed on (row) mean-centered data Y.

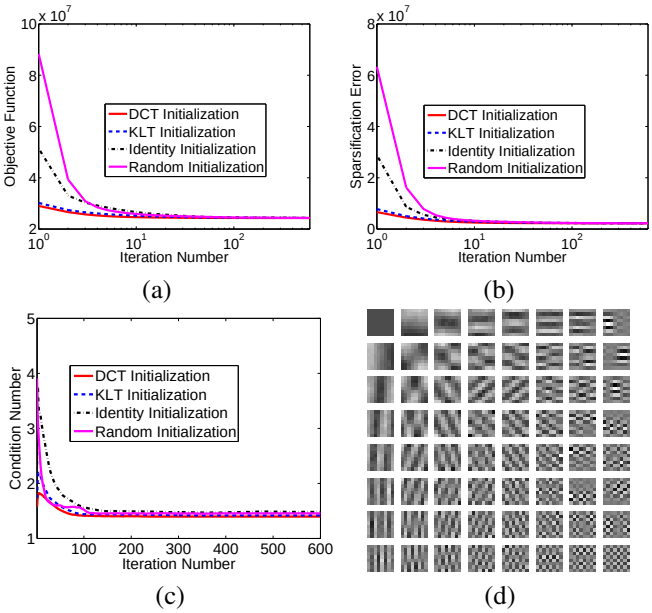


Fig. 3. Effect of different Initializations: (a) Objective function, (b) Sparsification error, (c) Condition number, (d) Rows of the learnt transform shown as patches for the case of DCT initialization.

The transforms learnt with different initializations in Fig. 3 also provide similar recovery PSNRs (that differ by hundredths of a dB) for the Barbara image.

C. Image Representation

For the second experiment, we learn sparsifying transforms from the $\sqrt{n} \times \sqrt{n}$ (zero mean) non-overlapping patches of the image Barbara at various patch sizes n. We study the image representation performance of the proposed algorithm involving closed-form solutions for Problem (P1). We compare the performance of our algorithm to the NLCG-based algorithm [5] that solves a version (without the absolute value within the log-determinant) of (P1), and the fixed 2D DCT. The DCT is a popular analytical transform that has been extensively used in compression standards such as JPEG. We employ sparsity levels $s = 0.17 \times n$ (rounded to nearest integer) throughout this subsection (for all methods), and λ_0 is fixed to the same value as in Section V-B for simplicity. The NLCG-based algorithm is executed with 128 NLCG iterations for each transform update step, and a fixed step size of 10^{-8} [5].

Figure 4 plots the normalized sparsification error (Fig. 4(a)) and recovery PSNR (Fig. 4(b)) metrics for the learnt transforms, and for the patch-based 2D DCT, as a function of patch size. The runtimes of the various transform learning schemes (Fig. 4(c)) are also plotted.

The learnt transforms provide better sparsification and recovery than the analytical DCT at all patch sizes. The gap in performance between the adapted transforms and the fixed DCT also increases with patch size (cf. [19] for a similar result and the reasoning). The learnt transforms in our experiments are all well-conditioned (condition numbers $\approx 1.2 - 1.6$). Note that the performance gap between the adapted transforms and the DCT can be amplified further at each patch size, by optimal

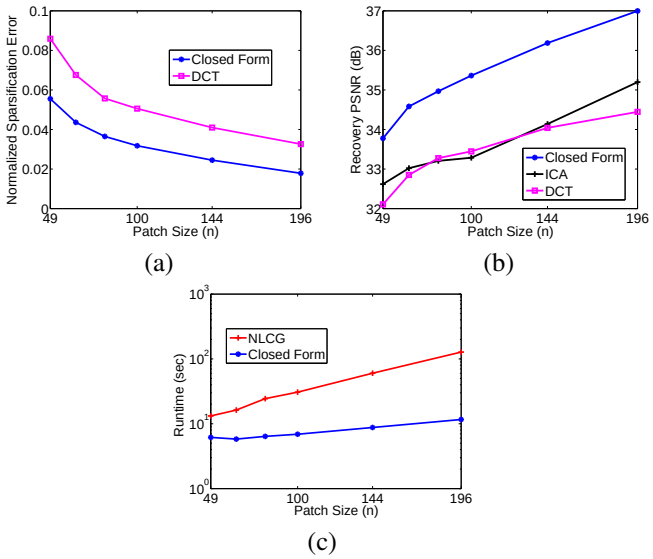


Fig. 4. Comparison of NLCG-based transform learning [5], Closed Form transform learning via (P1), DCT, and ICA [31] for different patch sizes: (a) Normalized sparsification error, (b) Recovery PSNR, (c) Runtime of transform learning. The plots for the NLCG and Closed Form methods overlap in (a) and (b). Therefore, we only show the plots for the Closed Form method there.

choice of λ_0 (or, optimal choice of condition number [6]).

The performance (normalized sparsification error and recovery PSNR) of the NLCG-based algorithm [5] is identical to that of the proposed Algorithm A1 for (P1) involving closed-form solutions. However, the latter is much faster (by 2-11 times) than the NLCG-based algorithm. The actual speedups depend in general, on how J (the number of NLCG iterations) scales with respect to N/n.

In yet another comparison, we show in Fig. 4(b), the recovery PSNRs obtained by employing Independent Component Analysis (ICA – a method for blind source separation) [31]–[35]. Similar to prior work on ICA-based image representation [36], we learn an ICA model A (a basis here) using the FastICA algorithm [31], [37], to represent the training signals as $Y = AZ$, where the rows of Z correspond to independent sources. Note that the ICA model enforces different properties (e.g., independence) than the transform model. Once the ICA model is learnt (using default settings in the author's MATLAB implementation [37]), the training signals are sparse coded in the learnt ICA model A [36] using the orthogonal matching pursuit algorithm [38], and the recovery PSNR (defined as in Section V-A, but with $W^{-1}X$ replaced by $A\hat{Z}$, where $\hat{Z}$ is the sparse code in the ICA basis) is computed. We found that the $A^{\dagger}$ obtained using the FastICA algorithm provides poor normalized sparsification errors (i.e., it is a bad transform model). Therefore, we only show the recovery PSNRs for ICA. As seen in Fig. 4(b), the proposed transform learning algorithm provides better recovery PSNRs than the ICA approach. This illustrates the superiority of the transform model for sparse representation (compression) of images compared to ICA. While we used the FastICA algorithm in Fig. 4(b), we have also observed similar performance for alternative (but slower) ICA methods [35], [39].

Finally, in comparison to synthesis dictionary learning, we have observed that algorithms such as K-SVD [9] perform slightly better than the transform learning Algorithm A1 for the task of image representation. However, the learning and application of synthesis dictionaries also imposes a heavy computational burden (cf. [5] for a comparison of the runtimes of synthesis K-SVD and NLCG-based transform learning). Indeed, an important advantage of our transform-based scheme for a compression application (similar to classical approaches involving the DCT or Wavelets), is that the transform can be applied as well as learnt very cheaply.

While we adapted the transform to a specific image (i.e., image-specific transform) in Fig. 4, a transform adapted to a variety of images (global transform) also performs well in test images [28]. Both global and image-specific transforms may hold promise for compression.

D. Image Denoising

The goal of denoising is to recover an estimate of an image $x \in \mathbb{R}^P$ (2D image represented as a vector) from its corrupted measurement $y = x + h$, where h is the noise. We work with h whose entries are i.i.d. Gaussian with zero mean and variance σ^2. We have previously presented a formulation [19] for patch-based image denoising using adaptive transforms as follows.

$$\min_{W,\{x_i\},\{\alpha_i\}} \sum_{i=1}^{N} \left\{ \|Wx_i - \alpha_i\|_2^2 + \lambda_i v(W) + \tau \|R_i\, y - x_i\|_2^2 \right\} \quad s.t. \quad \|\alpha_i\|_0 \leq s_i \ \forall\ i \qquad \text{(P3)}$$

Here, $R_i \in \mathbb{R}^{n \times P}$ extracts the i^{th} patch (N overlapping patches assumed) of the image y as a vector $R_i y$. Vector $x_i \in \mathbb{R}^n$ denotes a denoised version of $R_i y$, and $\alpha_i \in \mathbb{R}^n$ is a sparse representation of x_i in a transform W, with an a priori unknown sparsity s_i. The weight $\tau \propto 1/\sigma$ [13], [19], and λ_i is set based on the given noisy data $R_i y$ as $\lambda_0 \|R_i y\|_2^2$. The net weighting on $v(W)$ in (P3) is then $\lambda = \sum_i \lambda_i$.

We have previously proposed a simple *two-step* iterative algorithm to solve (P3) [19], that also estimates the unknown s_i. The algorithm iterates over a transform learning step and a variable sparsity update step (cf. [19] for a full description of these steps). We use the proposed alternating transform learning Algorithm A1 (involving closed-form updates) in the transform learning step. Once the denoised patches x_i are found, the denoised image x is obtained by averaging the x_i's at their respective locations in the image [19].

We now present brief results for our denoising framework employing the proposed efficient closed-form solutions in transform learning. We work with the images Barbara, Cameraman, Couple [7], and Brain (same as the one in Fig. 1 of [15]), and simulate i.i.d. Gaussian noise at 5 different noise levels (σ = 5, 10, 15, 20, 100) for each of the images. We compare the denoising results and runtimes obtained by

[6]The recovery PSNR depends on the trade-off between the sparsification error and condition number [5], [19]. For natural images, the recovery PSNR using the learnt transform is typically better at λ values corresponding to intermediate conditioning or well-conditioning, rather than unit conditioning, since unit conditioning is too restrictive [5].

[7]These three well-known images have been used in our previous work [19].

Parameter	Value	Parameter	Value
n	121	N'	32000
λ_0	0.031	τ	$0.01/\sigma$
C	1.04	s	12
M'	11	M	12

TABLE I

PARAMETER SETTINGS FOR OUR DENOISING ALGORITHM: n - NUMBER OF PIXELS IN A PATCH, λ_0 - WEIGHT IN (P3), C - SETS THRESHOLD THAT DETERMINES SPARSITY LEVELS IN THE VARIABLE SPARSITY UPDATE STEP [19], M' - NUMBER OF ITERATIONS OF THE TWO-STEP DENOISING ALGORITHM [19], N' - TRAINING SIZE FOR THE TRANSFORM LEARNING STEP (THE TRAINING PATCHES ARE CHOSEN UNIFORMLY AT RANDOM FROM ALL PATCHES IN EACH DENOISING ITERATION) [19], M - NUMBER OF ITERATIONS IN TRANSFORM LEARNING STEP, τ - WEIGHT IN (P3), s - INITIAL SPARSITY LEVEL FOR PATCHES [19].

our proposed algorithm with those obtained by the adaptive overcomplete synthesis K-SVD denoising scheme [13]. The Matlab implementation of K-SVD denoising [13] available from Michael Elad's website [30] was used in our comparisons, and we used the built-in parameter settings of that implementation.

We use 11×11 maximally overlapping image patches for our transform-based scheme. The resulting 121×121 square transform [8] has about the same number of free parameters as the 64×256 overcomplete K-SVD dictionary [13], [30]. The settings for the various parameters (not optimized) in our transform-based denoising scheme are listed in Table I. At $\sigma = 100$, we set the number of iterations of the two-step denoising algorithm [19] to $M' = 5$ (lower than the value in Table I), which also works well, and provides slightly smaller runtimes in denoising.

Table II lists the denoising PSNRs obtained by our transform-based scheme, along with the PSNRs obtained by K-SVD. The transform-based scheme provides better PSNRs than K-SVD for all the images and noise levels considered. The average PSNR improvement (averaged over all rows of Table II) provided by the transform-based scheme over K-SVD is 0.18 dB. When the NLCG-based transform learning [5] is used in our denoising algorithm, the denoising PSNRs obtained are very similar to the ones shown in Table II for the algorithm involving closed-form updates. However, the latter scheme is faster.

We also show the average speedups provided by our transform-based denoising scheme [9] over K-SVD denoising in Table III. For each image and noise level, the ratio of the runtimes of K-SVD denoising and transform denoising (involving closed-form updates) is first computed, and these speedups are averaged over the four images at each noise level. The transform-based scheme is about 10x faster than K-SVD denoising at lower noise levels. Even at very high noise ($\sigma = 100$), the transform-based scheme is still computationally cheaper than the K-SVD method.

[8] We have previously shown reasonable denoising performance for adapted (using NLCG-based transform learning [5]) 64×64 transforms [19]. The denoising performance usually improves when the transform size is increased, but with some degradation in runtime.

[9] Our MATLAB implementation is not currently optimized for efficiency. Therefore, the speedups here are computed by comparing our unoptimized MATLAB implementation (for transform-based denoising) to the corresponding MATLAB implementation [30] of K-SVD denoising.

Image	σ	Noisy PSNR	K-SVD	Transform
Barbara	5	34.15	38.09	38.28
	10	28.14	34.42	34.55
	15	24.59	32.34	32.39
	20	22.13	30.82	30.90
	100	8.11	21.86	22.42
Cameraman	5	34.12	37.82	37.98
	10	28.14	33.72	33.87
	15	24.60	31.50	31.65
	20	22.10	29.83	29.96
	100	8.14	21.75	22.01
Brain	5	34.14	42.14	42.74
	10	28.12	38.54	38.78
	15	24.62	36.27	36.43
	20	22.09	34.70	34.71
	100	8.13	24.73	24.83
Couple	5	34.16	37.29	37.35
	10	28.11	33.48	33.67
	15	24.59	31.44	31.60
	20	22.11	30.01	30.17
	100	8.13	22.58	22.60

TABLE II

PSNR VALUES IN DECIBELS FOR DENOISING WITH ADAPTIVE TRANSFORMS, ALONG WITH THE CORRESPONDING VALUES FOR 64×256 OVERCOMPLETE K-SVD [13]. THE PSNR VALUES OF THE NOISY IMAGES (DENOTED AS NOISY PSNR) ARE ALSO SHOWN.

σ	5	10	15	20	100
Average Speedup	9.82	8.26	4.94	3.45	2.16

TABLE III

THE DENOISING SPEEDUPS PROVIDED BY OUR TRANSFORM-BASED SCHEME (INVOLVING CLOSED-FORM SOLUTIONS) OVER K-SVD [13]. THE SPEEDUPS ARE AVERAGED OVER THE FOUR IMAGES AT EACH NOISE LEVEL.

We observe that the speedup of the transform-based scheme over K-SVD denoising decreases as σ increases in Table III. This is mainly because the computational cost of the transform-based scheme is dominated by matrix-vector multiplications (see [19] and Section III-C), and is invariant to the sparsity level s. On the other hand, the cost of the K-SVD denoising method is dominated by synthesis sparse coding, which becomes cheaper as the sparsity level decreases. Since sparsity levels in K-SVD denoising are set according to an error threshold criterion (and the error threshold $\propto \sigma^2$) [13], [30], they decrease with increasing noise in the K-SVD scheme. For these reasons, the speedup of the transform method over K-SVD is lower at higher noise levels in Table III.

We would like to point out that the actual value of the speedup over K-SVD also depends on the patch size used (by each method). For example, for larger images, a larger patch size would be used to capture image information better. The sparsity level in the synthesis model typically scales as a fraction of the patch size (i.e., $s \propto n$). Therefore, the actual speedup of transform-based denoising over K-SVD at a particular noise level would increase with increasing patch size – an effect that is not fully explored here due to limitations of space.

Thus, here, we have shown the promise of the transform-based denoising scheme (involving closed-form updates in learning) over overcomplete K-SVD denoising. Adaptive transforms provide better denoising, and are faster. The denoising PSNRs shown for adaptive transforms in Table II become even better at larger transform sizes, or by optimal choice of parameters [10]. We plan to combine transform learning with the state-of-the-art denoising scheme BM3D [40] in the near future. Since the BM3D algorithm involves some sparsifying transformations, we conjecture that adapting such transforms could improve the performance of the algorithm.

VI. Conclusions

In this work, we studied the problem formulations for learning well-conditioned square sparsifying transforms. The proposed alternating algorithms for transform learning involve efficient updates. In the limit of $\lambda \to \infty$, the proposed algorithms become orthonormal transform (or orthonormal synthesis dictionary) learning algorithms. Importantly, we provided convergence guarantees for the proposed transform learning schemes. We established that our alternating algorithms are globally convergent to the set of local minimizers of the non-convex transform learning problems. Our convergence guarantee does not rely on any restrictive assumptions. The learnt transforms obtained using our schemes provide better representations than analytical ones such as the DCT for images. In the application of image denoising, our algorithm provides comparable or better performance compared to synthesis K-SVD, while being much faster. Importantly, our learning algorithms, while performing comparably (in sparse image representation or denoising) to our previously proposed learning methods [5] involving iterative NLCG in the transform update step, are faster. We discuss the extension of our transform learning framework to the case of overcomplete (or, tall) transforms elsewhere [41], [42].

Appendix A
Solution of the Sparse Coding Problem (2)

First, it is easy to see that Problem (2) can be rewritten as follows

$$\sum_{i=1}^{N} \sum_{j=1}^{n} \min_{X_{ji}} \left\{ \left| (WY)_{ji} - X_{ji} \right|^2 + \eta_i^2 \, \theta \left(X_{ji} \right) \right\} \tag{17}$$

where the subscript ji denotes the element on the j^{th} row and i^{th} column of a matrix, and

$$\theta(a) = \begin{cases} 0 & , \text{ if } a = 0 \\ 1 & , \text{ if } a \neq 0 \end{cases} \tag{18}$$

We now solve the inner minimization problem in (17) with respect to X_{ji}. This corresponds to the problem

$$\min_{X_{ji}} \left\{ \left| (WY)_{ji} - X_{ji} \right|^2 + \eta_i^2 \, \theta \left(X_{ji} \right) \right\} \tag{19}$$

It is obvious that the optimal $\hat{X}_{ji} = 0$ whenever $(WY)_{ji} = 0$. In general, we consider two cases in (19). First, if the optimal $\hat{X}_{ji} = 0$ in (19), then the corresponding optimal objective value is $(WY)_{ji}^2$. If on the other hand, the optimal $\hat{X}_{ji} \neq 0$, then we must have $\hat{X}_{ji} = (WY)_{ji}$, in order to minimize the quadratic term in (19). In this (second) case, the optimal objective value in (19) is η_i^2. Comparing the optimal objective values in the two cases above, we conclude that

$$\hat{X}_{ji} = \begin{cases} 0 & , \text{ if } (WY)_{ji}^2 < \eta_i^2 \\ (WY)_{ji} & , \text{ if } (WY)_{ji}^2 > \eta_i^2 \end{cases} \tag{20}$$

If $|(WY)_{ji}| = \eta_i$, then the optimal $\hat{X}_{ji}$ in (19) can be either $(WY)_{ji}$ or 0, since both values correspond to the minimum value (i.e., η_i^2) of the cost in (19).

The preceding arguments establish that a (particular) solution $\hat{X}$ of (2) can be obtained as $\hat{X}_i = \hat{H}_{\eta_i}^1 (WY_i) \; \forall \, i$, where the (hard-thresholding) operator $\hat{H}_{\eta}^1(\cdot)$ was defined in Section III-A1.

Appendix B
Proof of Proposition 2

First, in the sparse coding step, we solve (1) for $\hat{X}$ with a fixed W. Then, the $\hat{X}$ discussed in Section III-A1 does not depend on the weight λ, and its form remains unaffected as $\lambda \to \infty$.

Next, in the transform update step, we solve for $\hat{W}$ in (4) with a fixed sparse code X. The transform update solution (5) does depend on the weight λ. For a particular λ, let us choose the matrix L_λ (indexed by λ) as the positive-definite square root $\left(YY^T + 0.5\lambda I \right)^{1/2}$. By Proposition 1, the closed-form formula (5) is invariant to the specific choice of this matrix. Let us define matrix M_λ as

$$M_\lambda \triangleq \sqrt{0.5\lambda} \, L_\lambda^{-1} Y X^T = \left[(2/\lambda) Y Y^T + I \right]^{-\frac{1}{2}} Y X^T \tag{21}$$

and its full SVD as $Q_\lambda \tilde{\Sigma}_\lambda R_\lambda^T$. As $\lambda \to \infty$, by (21), $M_\lambda = Q_\lambda \tilde{\Sigma}_\lambda R_\lambda^T$ converges to $M = Y X^T$, and it can be shown (see Appendix C) that the accumulation points of $\{Q_\lambda\}$ and $\{R_\lambda\}$ (considering the sequences indexed by λ, and letting $\lambda \to \infty$) belong to the set of left and right singular matrices of YX^T, respectively. Moreover, as $\lambda \to \infty$, the matrix $\tilde{\Sigma}_\lambda$ converges to a non-negative $n \times n$ diagonal matrix, which is the matrix of singular values of YX^T.

On the other hand, using (21) and the SVD of M_λ, (5) can be rewritten as follows

$$\hat{W}_\lambda = R_\lambda \left[\frac{\tilde{\Sigma}_\lambda}{\lambda} + \left(\frac{\tilde{\Sigma}_\lambda^2}{\lambda^2} + I \right)^{\frac{1}{2}} \right] Q_\lambda^T \left(\frac{YY^T}{0.5\lambda} + I \right)^{-\frac{1}{2}}$$

In the limit of $\lambda \to \infty$, using the aforementioned arguments on the limiting behavior of $\{Q_\lambda\}$, $\{\tilde{\Sigma}_\lambda\}$, and $\{R_\lambda\}$, the above update formula becomes (or, when YX^T has some degenerate singular values, the accumulation point(s) of the above formula assume the following form)

$$\hat{W} = \hat{R} \hat{Q}^T \tag{22}$$

where $\hat{Q}$ and $\hat{R}$ above are the full left and right singular matrices of YX^T, respectively. It is clear that the updated transform in (22) is orthonormal.

[10] The parameter settings in Table I (used in all our experiments for simplicity) can be optimized for each noise level, similar to [19].

Importantly, as $\lambda \to \infty$ (with $\xi = 0.5$), the sparse coding and transform update solutions in (P1) coincide with the corresponding solutions obtained by employing alternating minimization on the orthonormal transform learning Problem (13). Specifically, the sparse coding step for Problem (13) involves the same aforementioned Problem (1). Furthermore, using the condition $W^T W = I$, it is easy to show that the minimization problem in the transform update step of Problem (13) simplifies to the form in (14). Problem (14) is of the form of the well-known orthogonal Procrustes problem [43]. Therefore, denoting the full SVD of YX^T by $U\Sigma V^T$, the optimal solution in Problem (14) is given exactly as $\hat{W} = VU^T$. It is now clear that the solution for W in the orthonormal transform update Problem (14) is identical to the limit shown in (22).

Lastly, the solution to Problem (14) is unique if and only if YX^T is non-singular. The reasoning for the latter statement is similar to that provided in the proof of Proposition 1 (in Section III-A) for the uniqueness of the transform update solution for Problem (P1). ■

APPENDIX C
LIMIT OF A SEQUENCE OF SINGULAR VALUE DECOMPOSITIONS

Lemma 1: Consider a sequence $\{M_k\}$ with $M_k \in \mathbb{R}^{n \times n}$, that converges to M. For each k, let $Q_k \Sigma_k R_k^T$ denote a full SVD of M_k. Then, every accumulation point [11] (Q, Σ, R) of the sequence $\{Q_k, \Sigma_k, R_k\}$ is such that $Q\Sigma R^T$ is a full SVD of M. In particular, $\{\Sigma_k\}$ converges to Σ, the $n \times n$ singular value matrix of M.

Proof: Consider a convergent subsequence $\{Q_{q_k}, \Sigma_{q_k}, R_{q_k}\}$ of the sequence $\{Q_k, \Sigma_k, R_k\}$, that converges to the accumulation point (Q, Σ, R). It follows that

$$\lim_{k \to \infty} M_{q_k} = \lim_{k \to \infty} Q_{q_k} \Sigma_{q_k} R_{q_k}^T = Q \Sigma R^T \tag{23}$$

Obviously, the subsequence $\{M_{q_k}\}$ converges to the same limit M as the (original) sequence $\{M_k\}$. Therefore, we have

$$M = Q \Sigma R^T \tag{24}$$

By the continuity of inner products, the limit of a sequence of orthonormal matrices is orthonormal. Therefore the limits Q and R of the orthonormal subsequences $\{Q_{q_k}\}$ and $\{R_{q_k}\}$ are themselves orthonormal. Moreover, Σ, being the limit of a sequence $\{\Sigma_{q_k}\}$ of non-negative diagonal matrices (each with decreasing diagonal entries), is also a non-negative diagonal (the limit maintains the decreasing ordering of the diagonal elements) matrix. By these properties and (24), it is clear that $Q\Sigma R^T$ is a full SVD of M. The preceding arguments also indicate that the accumulation point of $\{\Sigma_k\}$ is unique, i.e., Σ. In other words, $\{\Sigma_k\}$ converges to Σ, the singular value matrix of M. ■

[11] Non-uniqueness of the accumulation point may arise due to the fact that the left and right singular vectors in the singular value decomposition (of M_k, M) are non-unique.

APPENDIX D
MAIN CONVERGENCE PROOF

Here, we present the proof of convergence for our alternating algorithm for (P1), i.e., proof of Theorem 1. The proof for Theorem 2 is very similar to that for Theorem 1. The only difference is that the non-negative barrier function $\psi(X)$ and the operator $H_s(\cdot)$ (in the proof of Theorem 1) are replaced by the non-negative penalty $\sum_{i=1}^{N} \eta_i^2 \left\|X_i\right\|_0$ and the operator $\hat{H}_\eta^1(\cdot)$, respectively. Hence, for brevity, we only provide a sketch of the proof of Theorem 2.

We will use the operation $\tilde{H}_s(b)$ here to denote the *set* of all optimal projections of $b \in \mathbb{R}^n$ onto the s-ℓ_0 ball, i.e., $\tilde{H}_s(b)$ is the set of all minimizers in the following problem.

$$\tilde{H}_s(b) = \underset{x \,:\, \|x\|_0 \leq s}{\arg\min} \ \|x - b\|_2^2 \tag{25}$$

Similarly, in the case of Theorem 2, the operation $\hat{H}_\eta(b)$ is defined as a mapping of a vector b to a set as

$$\left(\hat{H}_\eta(b)\right)_j = \begin{cases} 0 & , \text{if } \ |b_j| < \eta \\ \{b_j,\, 0\} & , \text{if } \ |b_j| = \eta \\ b_j & , \text{if } \ |b_j| > \eta \end{cases} \tag{26}$$

The set $\hat{H}_\eta(b)$ is in fact, the set of all optimal solutions to (2), when Y is replaced by the vector b, and $\eta_1 = \eta$.

Theorem 1 is now proved by proving the following properties one-by-one.

(i) Convergence of the objective in Algorithm A1.
(ii) Existence of an accumulation point for the iterate sequence generated by Algorithm A1.
(iii) All the accumulation points of the iterate sequence are equivalent in terms of their objective value.
(iv) Every accumulation point of the iterate sequence is a fixed point of the algorithm.
(v) Every fixed point of the algorithm is a local minimizer of $g(W, X)$ in the sense of (15).

The following shows the convergence of the objective.

Lemma 2: Let $\left\{W^k, X^k\right\}$ denote the iterate sequence generated by Algorithm A1 with data Y and initial (W^0, X^0). Then, the sequence of objective function values $\left\{g(W^k, X^k)\right\}$ is monotone decreasing, and converges to a finite value $g^* = g^*\left(W^0, X^0\right)$.

Proof: In the transform update step, we obtain a global minimizer with respect to W in the form of the closed-form analytical solution (5). Thus, the objective can only decrease in this step, i.e., $g(W^{k+1}, X^k) \leq g(W^k, X^k)$. In the sparse coding step too, we obtain an exact solution for X with fixed W as $\hat{X}_i = H_s(WY_i) \ \forall\, i$. Thus, $g(W^{k+1}, X^{k+1}) \leq g(W^{k+1}, X^k)$. Combining the results for the two steps, we have $g(W^{k+1}, X^{k+1}) \leq g(W^k, X^k)$ for any k.

Now, in Section II, we stated an explicit lower bound for the function $g(W, X) - \psi(X)$. Since $\psi(X) \geq 0$, we have that the function $g(W, X)$ is also lower bounded. Since the sequence of objective function values $\left\{g(W^k, X^k)\right\}$ is monotone decreasing and lower bounded, it must converge. ■

Lemma 3: The iterate sequence $\left\{W^k, X^k\right\}$ generated by Algorithm A1 is bounded, and it has at least one accumulation point.

Proof: The existence of a convergent subsequence for a bounded sequence is a standard result. Therefore, a bounded sequence has at least one accumulation point. We now prove the boundedness of the iterates. Let us denote $g(W^k, X^k)$ as g^k for simplicity. We then have the boundedness of $\{W^k\}$ as follows. First, since g^k is the sum of $v(W^k)$, and the non-negative sparsification error and $\psi(X^k)$ terms, we have that

$$v(W^k) \leq g^k \leq g^0 \tag{27}$$

where the second inequality above follows from Lemma 2. Denoting the singular values of W^k by β_i ($1 \leq i \leq n$), we have that $v(W^k) = \sum_{i=1}^{n}(\xi\beta_i^2 - \log\,\beta_i)$. The function $\sum_{i=1}^{n}(\xi\beta_i^2 - \log\,\beta_i)$, as a function of the singular values $\{\beta_i\}_{i=1}^{n}$ (all positive) is strictly convex, and it has bounded lower level sets. (Note that the level sets of a function $f : A \subset \mathbb{R}^n \mapsto \mathbb{R}$ (where A is unbounded) are bounded if $\lim_{k\to\infty} f(x^k) = +\infty$ whenever $\{x^k\} \subset A$ and $\lim_{k\to\infty} \|x^k\| = \infty$.) This fact, together with (27) implies that $\|W^k\|_F = \sqrt{\sum_{i=1}^{n}\beta_i^2} \leq c_0$ for a constant c_0, that depends on g^0. The same bound (c_0) works for any k.

We also have the following inequalities for sequence $\{X^k\}$.

$$\|X^k\|_F - \|W^kY\|_F \leq \|W^kY - X^k\|_F \leq \sqrt{g^k - v_0}$$

The first inequality follows from the triangle inequality and the second inequality follows from the fact that g^k is the sum of the sparsification error and $v(W^k)$ terms (since $\psi(X^k) = 0$), and $v(W^k) \geq v_0$ (v_0 defined in Section II). By Lemma 2, $\sqrt{g^k - v_0} \leq \sqrt{g^0 - v_0}$. Denoting $\sqrt{g^0 - v_0}$ by c_1, we have

$$\|X^k\|_F \leq c_1 + \|W^kY\|_F \leq c_1 + \sigma_1 \|W^k\|_F \tag{28}$$

where σ_1 is the largest singular value of the matrix Y. The boundedness of $\{X^k\}$ then follows from the previously established fact that $\|W^k\|_F \leq c_0$. ■

We now prove some important properties (Lemmas 4, 5, and 6) satisfied by any accumulation point of the iterate sequence $\{W^k, X^k\}$ in our algorithm.

Lemma 4: Any accumulation point (W^*, X^*) of the iterate sequence $\{W^k, X^k\}$ generated by Algorithm A1 satisfies

$$X_i^* \in \tilde{H}_s(W^*Y_i) \ \forall\ i \tag{29}$$

Proof: Let $\{W^{q_k}, X^{q_k}\}$ be a subsequence of the iterate sequence converging to the accumulation point (W^*, X^*). It is obvious that W^* is non-singular. Otherwise, the objective cannot be monotone decreasing over $\{W^{q_k}, X^{q_k}\}$.

We now have that for each (column) i ($1 \leq i \leq N$),

$$X_i^* = \lim_{k\to\infty} X_i^{q_k} = \lim_{k\to\infty} H_s(W^{q_k}Y_i) \in \tilde{H}_s(W^*Y_i) \tag{30}$$

where we have used the fact that when a vector sequence $\{\alpha^k\}$ converges to α^*, then the accumulation point of the sequence $\{H_s(\alpha^k)\}$ lies in $\tilde{H}_s(\alpha^*)$ [12] (see proof in Appendix E). ■

Lemma 5: All the accumulation points of the iterate sequence $\{W^k, X^k\}$ generated by Algorithm A1 with initial (W^0, X^0) correspond to the same objective value. Thus, they are equivalent in that sense.

[12]Since the mapping $H_s(\cdot)$ is discontinuous, the sequence $\{H_s(\alpha^k)\}$ need not converge to $H_s(\alpha^*)$, even though $\{\alpha^k\}$ converges to α^*.

Proof: Let $\{W^{q_k}, X^{q_k}\}$ be a subsequence of the iterate sequence converging to an accumulation point (W^*, X^*). Define a function $g'(W, X) = g(W, X) - \psi(X)$. Then, for any non-singular W, $g'(W, X)$ is continuous in its arguments. Moreover, for the subsequence $\{W^{q_k}, X^{q_k}\}$ and its accumulation point ($X_i^* \in \tilde{H}_s(W^*Y_i)\ \forall\ i$ by Lemma 4), the barrier function $\psi(X) = 0$. Therefore,

$$\begin{aligned}\lim_{k\to\infty} g(W^{q_k}, X^{q_k}) &= \lim_{k\to\infty} g'(W^{q_k}, X^{q_k}) + \lim_{k\to\infty} \psi(X^{q_k}) \\ &= g'(W^*, X^*) + 0 = g(W^*, X^*)\end{aligned} \tag{31}$$

where we have used the continuity of g' at (W^*, X^*) (since W^* is non-singular). Now, since, by Lemma 2, the objective converges for Algorithm A1, we have that $\lim_{k\to\infty} g(W^{q_k}, X^{q_k}) = \lim_{k\to\infty} g(W^k, X^k) = g^*$. Combining with (31), we have

$$g^* = g(W^*, X^*) \tag{32}$$

Equation (32) indicates that any accumulation point (W^*, X^*) of the iterate sequence $\{W^k, X^k\}$ satisfies $g(W^*, X^*) = g^*$, with g^* being the limit of $\{g(W^k, X^k)\}$. ■

Lemma 6: Any accumulation point (W^*, X^*) of the iterate sequence $\{W^k, X^k\}$ generated by Algorithm A1 satisfies

$$W^* \in \arg\min_W \|WY - X^*\|_F^2 + \lambda\xi\,\|W\|_F^2 - \lambda \log\,|\det W| \tag{33}$$

Proof: Let $\{W^{q_k}, X^{q_k}\}$ be a subsequence of the iterate sequence converging to the accumulation point (W^*, X^*). We then have (due to linearity) that

$$\lim_{k\to\infty} L^{-1}Y\,(X^{q_k})^T = L^{-1}Y\,(X^*)^T \tag{34}$$

Let $Q^{q_k}\Sigma^{q_k}\,(R^{q_k})^T$ denote the full singular value decomposition of $L^{-1}Y\,(X^{q_k})^T$. Then, by Lemma 1 of Appendix C, we have that every accumulation point (Q^*, Σ^*, R^*) of the sequence $\{Q^{q_k}, \Sigma^{q_k}, R^{q_k}\}$ is such that $Q^*\Sigma^*\,(R^*)^T$ is a full SVD of $L^{-1}Y\,(X^*)^T$, i.e.,

$$Q^*\Sigma^*\,(R^*)^T = L^{-1}Y\,(X^*)^T \tag{35}$$

In particular, $\{\Sigma^{q_k}\}$ converges to Σ^*, the full singular value matrix of $L^{-1}Y\,(X^*)^T$. Now, for a convergent subsequence (indexed by q_{n_k}) of $\{Q^{q_k}, \Sigma^{q_k}, R^{q_k}\}$ with limit (Q^*, Σ^*, R^*), using the closed-form formula (5), we have

$$\begin{aligned}W^{**} &\triangleq \lim_{k\to\infty} W^{q_{n_k}+1} \\ &= \lim_{k\to\infty} \frac{R^{q_{n_k}}}{2}\left(\Sigma^{q_{n_k}} + \left((\Sigma^{q_{n_k}})^2 + 2\lambda I\right)^{\frac{1}{2}}\right)(Q^{q_{n_k}})^T L^{-1} \\ &= \frac{R^*}{2}\left(\Sigma^* + \left((\Sigma^*)^2 + 2\lambda I\right)^{\frac{1}{2}}\right)(Q^*)^T L^{-1}\end{aligned} \tag{36}$$

where the last equality in (36) follows from the continuity of the square root function, and the fact that $\lambda > 0$. Equations (35), (36), and (5) imply that

$$W^{**} \in \arg\min_W \|WY - X^*\|_F^2 + \lambda\xi\,\|W\|_F^2 - \lambda \log\,|\det W| \tag{37}$$

Now, applying the same arguments used in (31) and (32) to the sequence $\{W^{q_{n_k}+1}, X^{q_{n_k}}\}$, we get that $g^* = g(W^{**}, X^*)$. Combining with (32), we get $g(W^{**}, X^*) = g(W^*, X^*)$, i.e.,

for a fixed sparse code X^*, W^* achieves the same value of the objective as W^{**}. This result together with (37) proves the required result (33). ■

Next, we use Lemmas 4 and 6 to show that any accumulation point of the iterate sequence in Algorithm A1 is a fixed point of the algorithm.

Lemma 7: Any accumulation point of the iterate sequence $\{W^k, X^k\}$ generated by Algorithm A1 is a fixed point of the algorithm.

Proof: Let $\{W^{q_k}, X^{q_k}\}$ be a subsequence of the iterate sequence converging to some accumulation point (W^*, X^*). Lemmas 4 and 6 then imply that

$$X^* \in \arg\min_X g(W^*, X) \tag{38}$$

$$W^* \in \arg\min_W g(W, X^*) \tag{39}$$

In order to deal with any non-uniqueness of solutions above, we assume for our algorithm that if a certain iterate W^{k+1} satisfies $g(W^{k+1}, X^k) = g(W^k, X^k)$, then we equivalently set $W^{k+1} = W^k$. Similarly, if $W^{k+1} = W^k$ holds, and the iterate X^{k+1} satisfies $g(W^{k+1}, X^{k+1}) = g(W^k, X^k)$, then we set $X^{k+1} = X^k$ [13]. Under the preceding assumptions, equations (39) and (38) imply that if we feed (W^*, X^*) into our algorithm (as initial estimates), the algorithm stays at (W^*, X^*). In other words, the accumulation point (W^*, X^*) is a fixed point of the algorithm. ■

Finally, the following Lemma 9 shows that any accumulation point (i.e., fixed point by Lemma 7) of the iterates is a local minimizer. Since the accumulation points are equivalent in terms of their cost, Lemma 9 implies that they are equally good local minimizers.

We need the following lemma for the proof of Lemma 9.

Lemma 8: The function $f(G) = tr(G) - \log|\det(I+G)|$ for $G \in \mathbb{R}^{n\times n}$, has a strict local minimum at $G = 0$, i.e., there exists an $\epsilon > 0$ such that for $\|G\|_F \leq \epsilon$, we have $f(G) \geq f(0) = 0$, with equality attained only at $G = 0$.

Proof: The gradient of $f(G)$ (when it exists) is given [5] as

$$\nabla_G f(G) = I - (I+G)^{-T} \tag{40}$$

It is clear that $G = 0$ produces a zero (matrix) value for the gradient. Thus, $G = 0$ is a stationary point of $f(G)$. The Hessian of $f(G)$ can also be derived [44] as $H = (I+G)^{-T} \otimes (I+G)^{-1}$, where "$\otimes$" denotes the Kronecker product. The Hessian is I_{n^2} at $G = 0$. Since this Hessian is positive definite, it means that $G = 0$ is a strict local minimizer of $f(G)$. The rest of the lemma is trivial. ■

Lemma 9: Every fixed point (W, X) of our Algorithm A1 is a minimizer of the objective $g(W, X)$ of Problem (P1), in the sense of (15) for sufficiently small dW, and ΔX in the union of the regions R1 and R2 in Theorem 1. Furthermore, if $\|WY_i\|_0 \leq s \,\forall\, i$, then ΔX can be arbitrary.

[13] This rule is trivially satisfied due to the way Algorithm A1 is written, except possibly for $k = 0$. In the latter case, if the rule is applicable, it means that the initial (W^0, X^0) is already a fixed point, and thus, no more iterations are performed. All aforementioned convergence results hold true for this degenerate case too.

Proof: It is obvious that W is a global minimizer of the transform update problem (4) for fixed sparse code X, and it thus provides a gradient value of 0 for the objective of (4). Thus, we have (using gradient expressions from [5])

$$2WYY^T - 2XY^T + 2\lambda\xi W - \lambda W^{-T} = 0 \tag{41}$$

We also have the following optimal property for the sparse code.

$$X_i \in \tilde{H}_s(WY_i) \;\forall\; i \tag{42}$$

Now, given such a fixed point (W, X), we consider perturbations $dW \in \mathbb{R}^{n\times n}$, and $\Delta X \in \mathbb{R}^{n\times N}$. We are interested in the relationship between $g(W+dW, X+\Delta X)$ and $g(W, X)$. It suffices to consider *sparsity preserving* ΔX, that is ΔX such that $X + \Delta X$ has columns that have sparsity $\leq s$. Otherwise the barrier function $\psi(X+\Delta X) = +\infty$, and $g(W+dW, X+\Delta X) > g(W, X)$ trivially. Therefore, in the rest of the proof, we only consider sparsity preserving ΔX.

For sparsity preserving $\Delta X \in \mathbb{R}^{n\times N}$, we have

$$\begin{aligned} g(W+dW, X+\Delta X) &= \|WY - X + (dW)Y - \Delta X\|_F^2 \\ &+ \lambda\xi \,\|W+dW\|_F^2 - \lambda \log|\det(W+dW)| \end{aligned} \tag{43}$$

Expanding the two Frobenius norm terms above using the trace inner product $\langle Q, R\rangle \triangleq \mathrm{tr}(QR^T)$, and dropping the non-negative terms $\|(dW)Y - \Delta X\|_F^2$ and $\lambda\xi\,\|dW\|_F^2$, we obtain

$$\begin{aligned} g(W+dW, X+\Delta X) \geq\; & \|WY - X\|_F^2 + \lambda\xi\,\|W\|_F^2 \\ & + 2\langle WY - X, (dW)Y - \Delta X\rangle + 2\lambda\xi\langle W, dW\rangle \\ & - \lambda \log|\det(W+dW)| \end{aligned} \tag{44}$$

Using (41) and the identity $\log|\det(W+dW)| = \log|\det W| + \log\left|\det(I + W^{-1}dW)\right|$, (44) simplifies to

$$\begin{aligned} g(W+dW, X+\Delta X) \geq\; & g(W, X) + \lambda\left\langle W^{-T}, dW\right\rangle \\ & - 2\langle WY - X, \Delta X\rangle - \lambda \log\left|\det(I + W^{-1}dW)\right| \end{aligned} \tag{45}$$

Define $G \triangleq W^{-1}dW$. Then, the terms $\left\langle W^{-T}, dW\right\rangle - \log\left|\det(I+W^{-1}dW)\right|$ (appearing in (45) with a scaling λ) coincide with the function $f(G)$ in Lemma 8. Therefore, by Lemma 8, we have that there exists an $\epsilon > 0$ such that for $\left\|W^{-1}dW\right\|_F \leq \epsilon$, we have $\left\langle W^{-T}, dW\right\rangle - \log\left|\det(I+W^{-1}dW)\right| \geq 0$, with equality attained here only at $dW = 0$. Since $\left\|W^{-1}dW\right\|_F \leq \|dW\|_F / \sigma_n$, where σ_n is the smallest singular value of W, we have that an alternative sufficient condition (for the aforementioned positivity of $f(W^{-1}dW)$) is $\|dW\|_F \leq \epsilon\sigma_n$. Assuming that dW lies in this neighborhood, equation (45) becomes

$$g(W+dW, X+\Delta X) \geq g(W, X) - 2\langle WY - X, \Delta X\rangle \tag{46}$$

Thus, we have the optimality condition $g(W+dW, X+\Delta X) \geq g(W, X)$ for any $dW \in \mathbb{R}^{n\times n}$ satisfying $\|dW\|_F \leq \epsilon\sigma_n$ (ϵ from Lemma 8), and for any $\Delta X \in \mathbb{R}^{n\times N}$ satisfying $\langle WY - X, \Delta X\rangle \leq 0$. This result defines Region R1.

We can also define a simple local region R2 $\subseteq \mathbb{R}^{n\times N}$, such that any sparsity preserving ΔX in the region results in $\langle WY - X, \Delta X\rangle = 0$. Then, by (46), $g(W+dW, X+\Delta X) \geq g(W, X)$ holds for $\Delta X \in$ R2. As we now show, the region R2 includes all $\Delta X \in \mathbb{R}^{n\times N}$ satisfying $\|\Delta X\|_\infty <$

$\min_i \{\phi_s(WY_i) : \|WY_i\|_0 > s\}$. In the definition of R2, we need only consider the columns of WY with sparsity greater than s. To see why, consider the set $\mathcal{A} \triangleq \{i : \|WY_i\|_0 > s\}$, and its complement $\mathcal{A}^c = \{1, ..., N\} \setminus \mathcal{A}$. Then, we have

$$\begin{aligned}\langle WY - X, \Delta X\rangle &= \sum_{i \in \mathcal{A} \cup \mathcal{A}^c} \Delta X_i^T (WY_i - X_i) \\ &= \sum_{i \in \mathcal{A}} \Delta X_i^T (WY_i - X_i) \end{aligned} \quad (47)$$

where we used the fact that $WY_i - X_i = 0$, $\forall\ i \in \mathcal{A}^c$. It is now clear that $\langle WY - X, \Delta X\rangle$ is unaffected by the columns of WY with sparsity $\leq s$. Therefore, these columns do not appear in the definition of R2. Moreover, if $\mathcal{A} = \emptyset$, then $\langle WY - X, \Delta X\rangle = 0$ for arbitrary $\Delta X \in \mathbb{R}^{n \times N}$, and thus, $g(W + dW, X + \Delta X) \geq g(W, X)$ holds (by (46)) for arbitrary ΔX. This proves the last statement of the Lemma.

Otherwise, assume $\mathcal{A} \neq \emptyset$, $\Delta X \in$ R2, and recall from (42) that $X_i \in \tilde{H}_s(WY_i)$ $\forall\ i$. It follows by the definition of R2, that for $i \in \mathcal{A}$, any $X_i + \Delta X_i$ with sparsity $\leq s$ will have the same sparsity pattern (non-zero locations) as X_i, i.e., the corresponding ΔX_i does not have non-zeros outside the support of X_i. Now, since $X_i \in \tilde{H}_s(WY_i)$, $WY_i - X_i$ is zero on the support of X_i, and thus, $\Delta X_i^T (WY_i - X_i) = 0$ for all $i \in \mathcal{A}$. Therefore, by (47), $\langle WY - X, \Delta X\rangle = 0$, for any sparsity-preserving ΔX in R2. ■

Note that the proof of Theorem 2 also requires Lemma 9, but with the objective $g(W, X)$ replaced by $u(W, X)$. Appendix F briefly discusses how the proof of the Lemma 9 is modified for the case of Theorem 2.

Appendix E
Limit of a Thresholded Sequence

Lemma 10: Consider a bounded vector sequence $\{\alpha^k\}$ with $\alpha^k \in \mathbb{R}^n$, that converges to α^*. Then, every accumulation point of $\{H_s(\alpha^k)\}$ belongs to the set $\tilde{H}_s(\alpha^*)$.

Proof: If $\alpha^* = 0$, then it is obvious that $\{H_s(\alpha^k)\}$ converges to $\tilde{H}_s(\alpha^*) = 0$. Therefore, we now only consider the case $\alpha^* \neq 0$.

First, let us assume that $\tilde{H}_s(\alpha^*)$ (the set of optimal projections of α^* onto the s-ℓ_0 ball) is a singleton and $\phi_s(\alpha^*) > 0$, so that $\phi_s(\alpha^*) - \phi_{s+1}(\alpha^*) > 0$. Then, for sufficiently large k ($k \geq k_0$), we will have $\left\|\alpha^k - \alpha^*\right\|_\infty < (\phi_s(\alpha^*) - \phi_{s+1}(\alpha^*))/2$, and then, $H_s(\alpha^k)$ has the same support set (non-zero locations) Γ as $H_s(\alpha^*) = \tilde{H}_s(\alpha^*)$. As $k \to \infty$, since $\left\|\alpha^k_\Gamma - \alpha^*_\Gamma\right\|_2 \to 0$ (where the subscript Γ indicates that only the elements of the vector corresponding to the support Γ are considered), we have that $\left\|H_s(\alpha^k) - H_s(\alpha^*)\right\|_2 \to 0$. Thus, the sequence $\{H_s(\alpha^k)\}$ converges to $H_s(\alpha^*)$ in this case.

Next, when $\tilde{H}_s(\alpha^*)$ is a singleton, but $\phi_s(\alpha^*) = 0$ (and $\alpha^* \neq 0$), let γ be the magnitude of the non-zero element of α^* of smallest magnitude. Then, for sufficiently large k ($k \geq k_1$), we will have $\left\|\alpha^k - \alpha^*\right\|_\infty < \gamma/2$, and then, the support of $H_s(\alpha^*) = \tilde{H}_s(\alpha^*)$ is contained in the support of $H_s(\alpha^k)$. Therefore, for $k \geq k_1$, we have

$$\left\|H_s(\alpha^k) - H_s(\alpha^*)\right\|_2 = \sqrt{\left\|\alpha^k_{\Gamma_1} - \alpha^*_{\Gamma_1}\right\|_2^2 + \left\|\alpha^k_{\Gamma_2}\right\|_2^2} \quad (48)$$

where Γ_1 is the support set of $H_s(\alpha^*)$, and Γ_2 (depends on k) is the support set of $H_s(\alpha^k)$ excluding Γ_1. (Note that α^* and $H_s(\alpha^*)$ are zero on Γ_2.) As $k \to \infty$, since $\alpha^k \to \alpha^*$, we have that $\left\|\alpha^k_{\Gamma_1} - \alpha^*_{\Gamma_1}\right\|_2 \to 0$ and $\left\|\alpha^k_{\Gamma_2}\right\|_2 \to 0$. Combining this with (48), we then have that the sequence $\{H_s(\alpha^k)\}$ converges to $H_s(\alpha^*)$ in this case too.

Finally, when $\tilde{H}_s(\alpha^*)$ is not a singleton (there are ties), it is easy to show that for sufficiently large k ($k \geq k_2$), the support of $H_s(\alpha^k)$ for each k coincides with the support of one of the optimal codes in $\tilde{H}_s(\alpha^*)$. In this case, as $k \to \infty$ (or, as $\alpha^k \to \alpha^*$), the distance between $H_s(\alpha^k)$ and the set $\tilde{H}_s(\alpha^*)$ converges to 0. Therefore, the accumulation point(s) of $\{H_s(\alpha^k)\}$ in this case, all belong to the set $\tilde{H}_s(\alpha^*)$. ■

In the case of (30), Lemma 10 implies $X_i^* \in \tilde{H}_s(W^* Y_i)$.

Appendix F
Modifications to Proof of Lemma 9 for Theorem 2

The (unconstrained) objective $u(W, X)$ here does not have the barrier function $\psi(X)$, but instead the penalty $\sum_{i=1}^N \eta_i^2 \|X_i\|_0$. Let us consider a fixed point (W, X) of the alternating Algorithm A2 that minimizes this objective. For a perturbation $\Delta X \in \mathbb{R}^{n \times N}$ satisfying $\|\Delta X\|_\infty < \min_i \{\eta_i/2\}$, it is easy to see (since X satisfies $X_i \in \hat{H}_{\eta_i}(WY_i)\ \forall\, i$) that

$$\sum_{i=1}^N \eta_i^2 \|X_i + \Delta X_i\|_0 = \sum_{i=1}^N \eta_i^2 \|X_i\|_0 + \sum_{i=1}^N \eta_i^2 \|\Delta X_i^c\|_0 \quad (49)$$

where $\Delta X_i^c \in \mathbb{R}^n$ is zero on the support (non-zero locations) of X_i, and matches ΔX_i on the complement of the support of X_i. Now, upon repeating the steps in the proof of Lemma 9 for the case of Theorem 2, we arrive at the following counterpart of equation (46).

$$\begin{aligned}u(W + dW, X + \Delta X) \geq & u(W, X) - 2\langle WY - X, \Delta X\rangle \\ & + \sum_{i=1}^N \eta_i^2 \|\Delta X_i^c\|_0 \end{aligned} \quad (50)$$

The term $-2\langle WY - X, \Delta X\rangle + \sum_{i=1}^N \eta_i^2 \|\Delta X_i^c\|_0$ can be easily shown to be ≥ 0 when $\|\Delta X\|_\infty < \min_i \{\eta_i/2\}$. ■

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] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 661–677, 2013.

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

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

[6] 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.

[7] K. Engan, S. Aase, and J. Hakon-Husoy, "Method of optimal directions for frame design," in *Proc. IEEE Int. Conf. Acoust. Speech, Sig. Proc.*, 1999, pp. 2443–2446.

[8] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T.-W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comput.*, vol. 15, no. 2, pp. 349–396, 2003.

[9] 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.

[10] 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.

[11] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2121–2130, 2010.

[12] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.

[13] 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.

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

[15] 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.

[16] M. Yaghoobi, S. Nam, R. Gribonval, and M. Davies, "Analysis operator learning for overcomplete cosparse representations," in *European Signal Processing Conference*, 2011, pp. 1470–1474.

[17] B. Ophir, M. Elad, N. Bertin, and M. Plumbley, "Sequential minimal eigenvalues - an approach to analysis dictionary learning," in *Proc. European Signal Processing Conference*, 2011, pp. 1465–1469.

[18] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Constrained overcomplete analysis operator learning for cosparse signal modelling," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2341–2355, 2013.

[19] S. Ravishankar and Y. Bresler, "Learning doubly sparse transforms for images," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4598–4612, 2013.

[20] M. Zibulevsky and B. A. Pearlmutter, "Blind source separation by sparse decomposition in a signal dictionary," *Neural Computation*, vol. 13, no. 4, pp. 863–882, 2001.

[21] L. Pfister, "Tomographic reconstruction with adaptive sparsifying transforms," Master's thesis, University of Illinois at Urbana-Champaign, Aug. 2013.

[22] L. Pfister and Y. Bresler, "Model-based iterative tomographic reconstruction with adaptive sparsifying transforms," in *SPIE International Symposium on Electronic Imaging: Computational Imaging XII*, vol. 9020, 2014, pp. 90 200H–1–90 200H–11.

[23] L. Mirsky, "On the trace of matrix products," *Mathematische Nachrichten*, vol. 20, no. 3-6, pp. 171–174, 1959.

[24] G. W. Stewart, *Matrix Algorithms: Volume 1: Basic Decompositions*. Philadelphia, PA: SIAM, 1998.

[25] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, 2001.

[26] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[27] ——, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5-6, pp. 629–654, 2008.

[28] S. Ravishankar and Y. Bresler, "Learning doubly sparse transforms for image representation," in *IEEE Int. Conf. Image Process.*, 2012, pp. 685–688.

[29] ——, "Sparsifying transform learning for compressed sensing MRI," in *Proc. IEEE Int. Symp. Biomed. Imag.*, 2013, pp. 17–20.

[30] M. Elad, "Michael Elad personal page," http://www.cs.technion.ac.il/~elad/Various/KSVD_Matlab_ToolBox.zip, [Online; accessed 2014].

[31] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 9, no. 7, pp. 1483–1492, 1997.

[32] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, 1995.

[33] ——, "The "independent components of natural scenes are edge filters," *Vision Research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[34] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York, NY: Wiley-Interscience, 2001.

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

[36] A. J. Ferreira and M. A. T. Figueiredo, "Class-adapted image compression using independent component analysis," in *Proceedings. 2003 International Conference on Image Processing*, vol. 1, 2003, pp. I–625–8 vol.1.

[37] A. Hyvärinen, "The fastica matlab package," http://research.ics.aalto.fi/ica/fastica/code/FastICA_2.5.zip, [Online; accessed August-2014].

[38] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit : recursive function approximation with applications to wavelet decomposition," in *Asilomar Conf. on Signals, Systems and Comput.*, 1993, pp. 40–44 vol.1.

[39] J.-F. Cardoso, "Jade for real-valued data," http://bsp.teithe.gr/members/downloads/Jade, 2005, [Online; accessed August-2014].

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

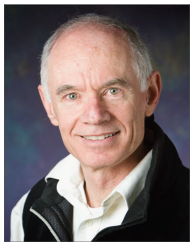
[41] B. Wen, S. Ravishankar, and Y. Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *International Journal of Computer Vision*, pp. 1–31, 2014. [Online]. Available: http://dx.doi.org/10.1007/s11263-014-0761-1

[42] S. Ravishankar and Y. Bresler, "Learning overcomplete sparsifying transforms for signal processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2013, pp. 3088–3092.

[43] P. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[44] M. Zibulevsky, "Blind source separation with relative newton method," in *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, 2003, pp. 897–902.

Saiprasad Ravishankar received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology Madras, in 2008. He received the M.S. and Ph.D. degrees in Electrical and Computer Engineering, in 2010 and 2014 respectively, from the University of Illinois at Urbana-Champaign, where he is currently an Adjunct Lecturer at the Department of Electrical and Computer Engineering, and a postdoctoral research associate at the Coordinated Science Laboratory. His current research interests include signal and image processing, medical imaging, inverse problems, image analysis, dictionary learning, compressed sensing, machine learning, computer vision, and big data applications.



Yoram Bresler received the B.Sc. (cum laude) and M.Sc. degrees from the Technion, Israel Institute of Technology, in 1974 and 1981 respectively, and the Ph.D degree from Stanford University, in 1986, all in Electrical Engineering. In 1987 he joined the University of Illinois at Urbana-Champaign, where he is currently a Professor at the Departments of Electrical and Computer Engineering and Bioengineering, and at the Coordinated Science Laboratory. Yoram Bresler is also President and Chief Technology Officer at InstaRecon, Inc., a startup he co-founded to commercialize breakthrough technology for tomographic reconstruction developed in his academic research. His current research interests include multi-dimensional and statistical signal processing and their applications to inverse problems in imaging, and in particular compressed sensing, computed tomography, and magnetic resonance imaging.

Dr. Bresler has served on the editorial board of a number of journals including the IEEE Transactions on Signal Processing, the IEEE Journal on Selected Topics in Signal Processing, Machine Vision and Applications, and the SIAM Journal on Imaging Science, and on various committees of the IEEE. Dr. Bresler is a fellow of the IEEE and of the AIMBE. He received two Senior Paper Awards from the IEEE Signal Processing society, and a paper he coauthored with one of his students received the Young Author Award from the same society in 2002. He is the recipient of a 1991 NSF Presidential Young Investigator Award, the Technion (Israel Inst. of Technology) Fellowship in 1995, and the Xerox Senior Award for Faculty Research in 1998. He was named a University of Illinois Scholar in 1999, appointed as an Associate at the Center for Advanced Study of the University in 2001-02, and Faculty Fellow at the National Center for Super Computing Applications in 2006.