CLOSED-FORM SOLUTIONS WITHIN SPARSIFYING TRANSFORM LEARNING

Saiprasad Ravishankar and Yoram Bresler

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

ABSTRACT

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, and medical image reconstruction. In this work, we focus specifically on the learning of orthonormal as well as well-conditioned square sparsifying transforms. The proposed algorithms alternate between a sparse coding step, and a transform update step. We derive the exact analytical solution for each of these steps. Adaptive well-conditioned transforms are shown to perform better in applications compared to adapted orthonormal ones. Moreover, the closed form solution for the transform update step achieves the global minimum in that step, and also provides speedups over iterative solutions involving conjugate gradients. We also present examples illustrating the promising performance and significant speed-ups of transform learning over synthesis K-SVD in image denoising.

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

1. 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. The popular *synthesis model* states that a signal $y \in \mathbb{R}^n$ may be represented as a linear combination of a small number of columns from a dictionary $D \in \mathbb{R}^{n \times K}$ [1, 2], i.e., $y = Dx$, where $x \in \mathbb{R}^K$ is sparse with $\|x\|_0 \ll K$. The l_0 quasi-norm counts the number of non-zeros in x. Given a signal y and synthesis dictionary D, the following *synthesis sparse coding* problem of extracting the sparse representation x has been extensively studied in recent years.

$$\min_x \|y - Dx\|_2^2 \;\; s.t. \;\; \|x\|_0 \leq s \tag{1}$$

Here, s denotes the desired sparsity level, and the signal y is more generally assumed to satisfy $y = Dx + \xi$, where ξ is an error/noise term in the signal domain. Although this problem is NP-hard (Non-deterministic Polynomial-time hard), under certain conditions it can be solved exactly using polynomial-time algorithms [3, 4], which however, tend to be computationally expensive.

An alternative model for sparse representation of data is the *analysis model* [1], which suggests that given the signal y and analysis dictionary $\Omega \in \mathbb{R}^{m \times n}$, the representation $\Omega y \in \mathbb{R}^m$ is sparse, i.e., $\|\Omega y\|_0 \ll m$ [5]. When the signal y is noisy, the analysis model is extended as $y = q + \xi$, with Ωq being sparse, and ξ representing the noise [5]. We refer to the extension as the *noisy signal analysis model*. Specifically, the problem of recovering the clean signal q from the noisy y is formulated as follows [5], and is known as *analysis sparse coding*, with Ωq being the sparse code.

$$\min_q \|y - q\|_2^2 \;\; s.t. \;\; \|\Omega q\|_0 \leq m - t \tag{2}$$

Here, t represents the minimum number of zeros in Ωq (also called co-sparsity). This problem is also NP-hard, just like synthesis sparse coding. However, approximate algorithms have been proposed for solving the analysis sparse coding problem [5], which similar to the synthesis case are also computationally expensive.

In this paper, we focus on a generalized analysis model for sparse representation, which we call the *transform model* [6]. This model suggests that a signal y 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, i.e., $\|x\|_0 \ll m$, and e is the residual in the transform domain. This is a generalization of the analysis model with Ωy exactly sparse. Additionally, unlike the analysis model in which the sparse code Ωy lies in the range space of Ω, the sparse representation x in the transform model is not constrained to lie in the range space of W. This in fact, makes the transform model more general than even the noisy signal analysis model (cf. [6]). Note that the assumption $Wy \approx x$ has been traditionally used in transform coding [7], which pre-dates the analysis/synthesis concepts [8] (hence, the name choice - transform model).

When a sparsifying transform W is known for the signal y, the process of obtaining a sparse code x of given sparsity s involves solving the following problem, which we call transform sparse coding for simplicity.

$$\min_x \|Wy - x\|_2^2 \;\; s.t. \;\; \|x\|_0 \leq s \tag{3}$$

The solution $\hat{x}$ is obtained exactly by thresholding Wy and retaining the s largest coefficients. Conversely, given W and sparse code x, we can recover a least squares estimate of the true signal y by minimizing $\|Wy - x\|_2^2$ over all $y \in \mathbb{R}^n$. The recovered signal is then simply $W^\dagger x$, where $W^\dagger$ is the pseudo-inverse of W. Thus, a sparsifying transform is much simpler and faster to use in practice.

Adapting the sparse model to data can prove advantageous in applications. The idea of learning a synthesis dictionary from training signals has received a lot of attention [9, 10, 11]. Adaptive synthesis dictionaries have been shown to be useful in various applications [12, 13]. However, synthesis dictionary learning is typically non-convex and NP-hard, and algorithms such as K-SVD [10] can get easily caught in local minima or saddle points. The learning of analysis dictionaries, employing either the analysis model or its noisy signal extension, has also received some recent attention [14, 15, 16, 5]. However, this problem too is typically non-convex and NP-hard [5], and no convergence guarantees exist for the analysis dictionary learning algorithms.

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

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

In this paper, we focus on the learning of unitary as well as well-conditioned sparsifying transforms. We will restrict ourselves to square transforms, i.e., $W \in \mathbb{R}^{n \times n}$. While we have very recently developed formulations and algorithms for transform learning [6], in this work we will derive *efficient closed-form solutions* for the update steps, that further enhance the convergence and computational properties of transform learning. The resultant algorithms will be shown to hold promise for compression and denoising.

2. TRANSFORM LEARNING

Given a matrix $Y \in \mathbb{R}^{n \times N}$, whose columns represent training signals, we recently proposed the following formulation for learning square sparsifying transforms for Y [6].

$$\text{(P0)} \quad \min_{W,X} \ \|WY - X\|_F^2 - \lambda \log\det W + \mu \|W\|_F^2 \quad s.t. \ \ \|X_i\|_0 \leq s \ \forall \ i$$

Here, $X \in \mathbb{R}^{n \times N}$ is a matrix, whose columns X_i are the sparse codes of the training signals (columns) in Y. The term $\|WY - X\|_F^2$ in (P0) is called *sparsification error* [6], and denotes the deviation of the data in the transform domain from perfect sparsity.

The $\log\det W$ penalty in (P0) helps enforce full rank on the transform W, and eliminates degenerate solutions (such as those with zero, or repeated rows). The $\|W\|_F^2$ penalty in (P0) helps remove a 'scale ambiguity' [6] in the solution (the scale ambiguity occurs when the data admits an exactly sparse representation), and together with the $-\log\det W$ penalty additionally helps control the condition number of the learnt transform. Badly conditioned transforms typically convey little information and may degrade performance in applications [6]. As the parameter λ is increased in (P0) with fixed ratio μ/λ, the optimal transform(s) become well-conditioned. In the limit $\lambda \to \infty$, their condition number tends to 1 [6].

We have shown [6] that the cost function in (P0) is lower bounded. The objective function in (P0) has log-barriers at W for which $\det W \leq 0$. These log-barriers prevent optimization algorithms that minimize the objective from getting into infeasible regions. The restriction $\det W > 0$, can be made without loss of generality [6] (one can switch from a W with $\det W < 0$ to one with $\det W > 0$ trivially by swapping two rows of W).

We have proposed [6] an alternating algorithm for solving (P0) that alternates between solving for X (*sparse coding step*) and W (*transform update step*). While the sparse coding step has an exact solution, the transform update step was solved using an iterative method such as conjugate gradients (CG) [6]. The alternating algorithm for transform learning has a low computational cost [6] compared to synthesis and analysis dictionary learning.

In this work, we will prove that both steps of transform learning can in fact, be solved exactly and cheaply. We will first consider the learning of a special type of transform, the orthonormal transform, and then consider the more general case involving Problem (P0). We refer to the latter case as 'unconstrained' transform learning to distinguish it from the orthonormal case.

2.1. Orthonormal Transform Learning

There are many well-known examples of analytical orthonormal transforms such as the discrete cosine transform (DCT), discrete fourier transform (DFT), and Wavelets. Orthonormality enforces a constraint of the form $W^T W = I_n$, where I_n is the $n \times n$ identity matrix. When this constraint is used in Problem (P0), it simplifies as follows.

$$\text{(P1)} \quad \min_{W,X} \ \|WY - X\|_F^2 \ \ s.t. \ \ W^T W = I_n, \ \|X_i\|_0 \leq s \ \forall \ i$$

A transform learnt via Problem (P1) can also be used as an orthonormal synthesis dictionary (W^T is a synthesis dictionary). When Problem (P1) is solved using alternating minimization, the sparse coding step remains identical to that for Problem (P0) as follows.

$$\min_{X} \ \|WY - X\|_F^2 \ \ s.t. \ \ \|X_i\|_0 \leq s \ \forall \ i \tag{4}$$

The solution $\hat{X}$ is computed exactly by thresholding WY, and retaining the s largest coefficients (magnitude-wise) in each column.

The transform update step involves the following optimization problem, where we have simplified the objective of (P1).

$$\max_{W} \ tr\left(WYX^T\right) \ \ s.t. \ \ W^T W = I_n \tag{5}$$

Here, 'tr' represents the matrix trace operation, and $(\cdot)^T$ denotes the matrix transpose operation. The above problem is of the form of the orthogonal Procrustes problem [17]. Denoting the full singular value decomposition (SVD) of YX^T by $U\Sigma V^T$ ($U \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{n \times n}$), the optimal solution $\hat{W} = VU^T$ (unique only when the singular values of YX^T are non-degenerate and non-zero, since U and V are not unique otherwise).

2.2. Unconstrained Transform Learning

Problem (P0) is a generalized version of (P1) that allows full control over the condition number of W. While it is sufficient to consider the $\det W > 0$ case [6], by introducing the absolute value, we allow both positive and negative determinants in the following formulation, which makes the derivation of closed-form solutions simpler.

$$\text{(P2)} \quad \min_{W,X} \ \|WY - X\|_F^2 - \lambda \log\,|\det W| + \mu \|W\|_F^2 \quad s.t. \ \ \|X_i\|_0 \leq s \ \forall \ i$$

The sparse coding step in the alternating algorithm for (P2) is exactly as in equation (4). The transform update step involves the following unconstrained non-convex minimization.

$$\min_{W} \ \|WY - X\|_F^2 + \mu \|W\|_F^2 - \lambda \log\,|\det W| \tag{6}$$

The objective function can be re-written as follows.

$$tr\left\{W\left(YY^T + \mu I_n\right)W^T - 2WYX^T + XX^T\right\} - \lambda \log\,|\det W|$$

We decompose the positive-definite matrix $YY^T + \mu I_n$ as LL^T (e.g., Cholesky decomposition). We then employ a change of variables $B = WL$, and use the multiplicativity of the determinant, i.e., $\det B = (\det W)(\det L)$, which implies, $\log\,|\det B| = \log\,|\det W| + C$, where $C = \log\,|\det L|$. The optimization problem (6) then becomes

$$\min_{B} \ tr\left(BB^T\right) - 2\,tr\left(BL^{-1}YX^T\right) - \lambda \log\,|\det B| \tag{7}$$

Next, we let B have a full SVD of $T\Gamma V^T$, and let $L^{-1}YX^T$ have a full SVD of $Q\Sigma R^T$ ($T, \Gamma, V, Q, \Sigma, R$ are all $n \times n$ matrices), with γ_i and σ_i denoting the diagonal entries of Γ and Σ, respectively. The unconstrained minimization (7) then simplifies as follows.

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

For the inner maximization, we use the inequality $tr\left(T\Gamma V^T Q \Sigma R^T\right) \leq tr\left(\Gamma\Sigma\right)$ [18], with the upper bound being attained by setting $T = R$ and $V = Q$. The minimization with respect to Γ is then

$$\min_{\gamma_i} \sum_{i=1}^{n} \gamma_i^2 - 2\sum_{i=1}^{n} \gamma_i \sigma_i - \lambda \sum_{i=1}^{n} \log\, \gamma_i \tag{8}$$

This problem is convex in the γ_i's and the solution is obtained by differentiating the above cost with respect to the γ_i's and setting the derivative to 0. This gives $\gamma_i = \frac{\sigma_i \pm \sqrt{\sigma_i^2 + 2\lambda}}{2}$. Since the singular values are all positive, the solution is $\gamma_i = \frac{\sigma_i + \sqrt{\sigma_i^2 + 2\lambda}}{2}\ \forall i$.

Thus, the closed-form solution or global minimizer for the transform update step (6) can be compactly written as

$$\hat{W} = \frac{R}{2}\left(\Sigma + \left(\Sigma^2 + 2\lambda I_n\right)^{\frac{1}{2}}\right) Q^T L^{-1} \tag{9}$$

where the square root above is the positive-definite square root. It can be verified that the gradient of the objective of (6), is zero at this solution. The solution (9) is also invariant to the specific choice for L in the above derivation (note that if L_1 and L_2 satisfy $YY^T + \mu I_n = L_1 L_1^T = L_2 L_2^T$, then $L_2 = L_1 G$, where G is an orthonormal matrix). The closed-form solution (9) is unique only if the singular values of $L^{-1} Y X^T$ are non-degenerate (distinct) and non-zero.

Note that although CG works well for the non-convex transform update step of (P0) [6] (or equivalently, (P2)), convergence to the global minimum of that step is not guaranteed for CG. Moreover, the closed-form solution for the transform update provides computational speed-ups compared to CG. The closed-form updates in learning also ensure that the objective functions converge for our algorithms for (P1) and (P2). Empirical evidence suggests that the iterates too converge, and that transform learning is insensitive to initialization [6], leading us to conjecture that the algorithms converge to the global minimizers of the learning problems.

The computational cost of the sparse coding step in our algorithms (for (P1), (P2)) scales as $O(Nn^2)$ [6]. This cost is dominated by the computation of the product WY, whereas the projection onto the ℓ_0 ball only requires $O(nN \log n)$ operations, when employing sorting. For the transform update step, the product YX^T needs to be pre-computed, which requires $\alpha N n^2$ multiply-add operations for an X with s-sparse columns, and $s = \alpha n$. When the transform update step is solved using the closed-form solutions, the computational cost scales as $O(n^3)$. On the other hand, when CG is employed [6], the cost of the transform update step scales as $O(Jn^3)$, where J is the number of CG steps (typically, a fixed number of CG iterations works well). Thus, the closed-form solution allows for both a cheap and exact solution to the transform update step.

3. IMAGE DENOISING

We now consider the application of image denoising, which is a widely studied problem of estimating an image $x \in \mathbb{R}^P$ (2D image represented as vector) from its measurement $y = x + h$ corrupted by noise h. We recently presented a simple image denoising technique involving adaptive transforms [19]. Here, we introduce the following denoising formulation, that extends our previous technique.

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

Here, $Q(W) = -\log\det W + \frac{\mu}{\lambda}\|W\|_F^2$. Vector $R_i\, y$ denotes the ith patch of image y (M overlapping patches assumed), with $R_i \in \mathbb{R}^{n \times P}$ being the operator that extracts it. We assume that the noisy patch $R_i\, y$ can be approximated by a noiseless version x_i that is approximately sparsifiable (the *noisy signal transform model* [6]). Vector $\alpha_i \in \mathbb{R}^n$ denotes the sparse code of x_i with s_i non-zeros, and the weight τ is inversely proportional to the noise level σ [12].

The solution to Problem (P3) involves a two step optimization. In the *transform learning* **Step 1**, we fix $x_i = R_i\, y$ and $s_i = s$ (fixed s initially) in (P3), and solve for W and $\alpha_i\ \forall i$, using our proposed learning algorithms. In the *variable sparsity update* **Step 2**, we update the sparsity levels s_i for all i. Note that for fixed W and α_i, (P3) reduces to a least squares problem, that can be solved independently for each x_i. However, we only let α_i be a thresholded version of $WR_i\, y$, and determine the s_i's in Step 2, i.e., $\alpha_i = H_{s_i}(WR_i y)$, with $H_{s_i}(\cdot)$ denoting the operator that retains the s_i largest elements (magnitude-wise) in a vector, while setting the remaining elements to zero. We choose the sparsity s_i for the ith patch such that the error term $\|R_i\, y - x_i\|_2^2$ computed after updating x_i by least squares (with α_i held at $H_{s_i}(WR_i y)$) is below $nC^2\sigma^2$ [12] (the error term decreases to zero, as $s_i \nearrow n$), where C is a fixed parameter. This requires repeating the least squares update of x_i for each i at various sparsity levels incrementally, to determine the level at which the error term falls below the required threshold. However, this process can be done very efficiently (cf. [19] for details).

Once the variable sparsity levels s_i are chosen for all i, we use the new s_i's back in the transform learning Step 1, and iterate over the learning and variable sparsity update steps, which leads to a better denoising performance compared to one iteration. In the final iteration, the x_i's that are computed (satisfying the $\|R_i\, y - x_i\|_2^2 \leq nC^2\sigma^2$ condition) represent the denoised patches. Once, the denoised patches x_i are found, the denoised image x is obtained by averaging the x_i's at their respective locations in the image, and x is then restricted to its range (e.g., 0-255). Note that we work with mean subtracted patches during optimization and typically learn on a subset of all patches (cf. [19]).

4. EXPERIMENTS

For the first experiment, we learn sparsifying transforms from the $\sqrt{n} \times \sqrt{n}$ (zero mean) non-overlapping patches of the image Barbara [6] at various patch sizes n. We study the performance of the proposed algorithms involving closed-form solutions for Problems (P1) and (P2). We compare their performance to the CG-based algorithm [6] that solves (P0), and the fixed DCT. The various parameters are set as $\lambda = \mu = 4 \times 10^5$, $s = 0.17 \times n$ (rounded to nearest integer). The CG-based algorithm is executed with 128 CG iterations, and a fixed step size of 10^{-8}.

We measure the quality of the learnt transforms using the normalized sparsification error, condition number, and recovery PSNR metrics. The normalized sparsification error [6] is defined as $\|WY - X\|_F^2 \,/\, \|WY\|_F^2$, and it measures the fraction of energy lost in sparse fitting in the transform domain, an interesting property to observe for the adapted transforms. The recovery peak signal to noise ratio (recovery PSNR) is defined as $255\sqrt{P} / \left\|Y - W^{-1}X\right\|_F$ in dB, where P is the number of image pixels. It measures the error in recovering the patches Y (or equivalently, the image for non-overlapping patches) as $W^{-1}X$ from their sparse codes X obtained by thresholding WY. Note that the recovery PSNR itself depends on the (trade-off between) sparsification error and condition number of W [6].

Figure 1 plots the various metrics for the transforms learnt using the various algorithms, and for the patch-based 2D DCT [6], as a function of patch size. The run times of the various learning schemes are also plotted. The learnt transforms provide better sparsification

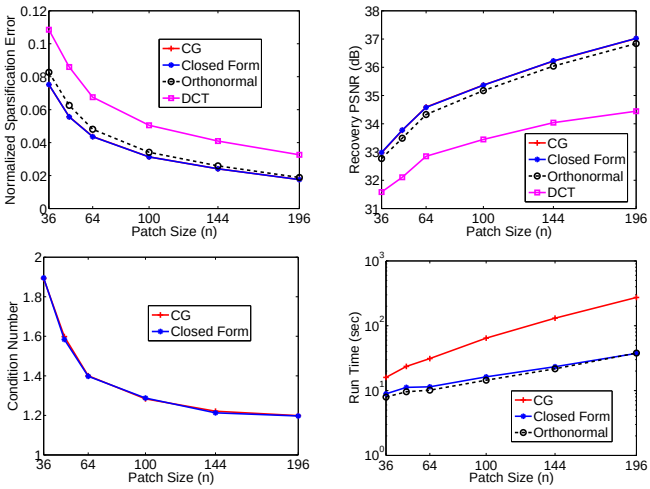

Fig. 1. Comparison of **CG**-based transform learning [6], **Closed Form** transform learning via (P2), **Orthonormal** transform learning via (P1), and **DCT**. Top: Normalized sparsification error vs. patch size (left), Recovery PSNR vs. patch size (right). Bottom: Condition Number vs. patch size (left), Run time vs. patch size (right). Note that the **CG** and **Closed Form** curves overlap in all cases, except for the run times.

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, because the size of the training set also decreases, thereby allowing increasing benefits from adaptivity. The transforms learnt using (P0)/(P2) are well-conditioned.

The performance of the CG-based algorithm [6] is almost identical to that of the proposed algorithm for (P2) involving closed-form solutions. However, the latter is much faster (by 2-7 times) than the CG-based algorithm. The actual speedups depend in general, on how N/n scales with respect to J. The adapted well-conditioned transforms also provide better sparsification and recovery (upto 0.3 dB better recovery) compared to the adapted orthonormal transforms, indicating the usefulness of well-conditioning over the more restrictive unit-conditioning. The performance gap between the adapted well-conditioned and orthonormal transforms can be amplified further at each patch size, by optimal choice of λ (or in other words, optimal choice of condition number).

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

Next, we present preliminary results for our denoising framework, employing the proposed efficient closed-form solutions in transform learning. We add i.i.d. gaussian noise at noise level $\sigma = 10$ to the peppers image. We choose our algorithm parameters as $n = 64$, $\mu = \lambda = 10^6$, initial $s = 0.15 \times n$ (rounded to nearest integer), $C = 1.08$, and $\tau = 0.01/\sigma$. We executed our denoising algorithm for 3 iterations, each with 80 iterations of transform learning. The noisy image (PSNR = 28.1 dB) is shown in Figure 2, along with the denoised image (PSNR = 34.38 dB) obtained using our closed-form-solution-based learning via (P2). The learnt transform in this case is well-conditioned with condition number 2.28. When the learning was done using the CG-based algorithm (256 CG iterations) [6], we obtain similar denoising, but at about 2x slower speed, indicating the efficiency of the proposed exact closed-form solutions. Moreover, when our denoising algorithm is run with

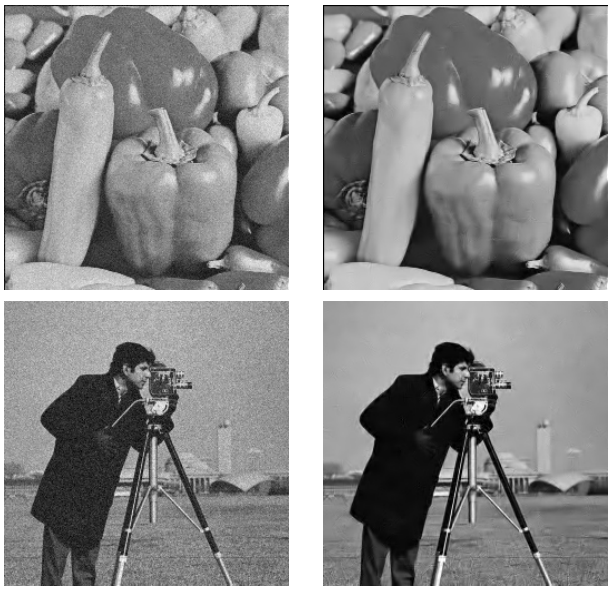

Fig. 2. Noisy Images (Left), Denoised Images (Right).

orthonormal transform learning via (P1), the denoised image has a slightly lower PSNR of 34.33 dB. The denoising gap between non-unitary and unitary adapted transforms typically increases with noise level.

We also compared our denoising performance to that obtained with a 64×256 K-SVD overcomplete synthesis dictionary [12, 20]. The latter provided a lower denoising PSNR of 34.21 dB. Importantly, our denoising algorithm involving closed-form-solution-based learning via (P2) takes only 47s to execute compared to 9.5 minutes for K-SVD, a 12x speeedup. (Note that we used a training set of smaller size for our algorithms compared to K-SVD, since square transforms have fewer free parameters.) The results demonstrate the significant speedups of transform-based denoising.

We repeat the denoising experiment for the cameraman image using $\sigma = 15$. The noisy image (PSNR = 24.6 dB), and its denoised version (PSNR = 31.60 dB) obtained using the transform adapted via our closed-form-solution-based learning (P2), are shown in Figure 2. The denoising PSNR is better than that obtained using the 64×256 K-SVD synthesis dictionary (PSNR = 31.50 dB) [12, 20], while our denoising algorithm is also much (12x) faster.

Our results here indicate the potential of transform-based denoising over synthesis-dictionary-based denoising. Transform-based denoising also performs better than analysis-dictionary-based denoising [21]. While we presented preliminary results in this work, we expect the denoising performance of our algorithms to improve/become comparable to the state of the art (for example [22]) with multiscale and overcomplete extensions of transform learning (similarly to the synthesis case [23]).

5. CONCLUSIONS

In this work, we studied the problem formulations for learning orthonormal as well as well-conditioned sparsifying transforms. The proposed alternating algorithms involve efficient closed-form solutions. The learnt transforms provide better representations than analytical ones such as the DCT for images. Moreover, our algorithm is faster than previous ones involving iterative CG in the transform update step. In the application of image denoising, our algorithms provide comparable or better performance over the synthesis K-SVD, while being much faster.

6. REFERENCES

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

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

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

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

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

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

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

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

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

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

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

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

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

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

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

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

[23] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *SIAM Multiscale Modeling and Simulation*, vol. 7, no. 1, pp. 214–241, 2008.