Int J Comput Vis (2015) 114:137–167
DOI 10.1007/s11263-014-0761-1

Structured Overcomplete Sparsifying Transform Learning with Convergence Guarantees and Applications

Bihan Wen · **Saiprasad Ravishankar** · **Yoram Bresler**

Received: 10 February 2014 / Accepted: 1 September 2014 / Published online: 19 October 2014

Abstract In recent years, sparse signal modeling, especially using the synthesis model has been popular. Sparse coding in the synthesis model is however, NP-hard. Recently, interest has turned to the sparsifying transform model, for which sparse coding is cheap. However, natural images typically contain diverse textures that cannot be sparsified well by a single transform. Hence, in this work, we propose a union of sparsifying transforms model. Sparse coding in this model reduces to a form of clustering. The proposed model is also equivalent to a structured overcomplete sparsifying transform model with block cosparsity, dubbed OCTOBOS. The alternating algorithm introduced for learning such transforms involves simple closed-form solutions. A theoretical analysis provides a convergence guarantee for this algorithm. It is shown to be globally convergent to the set of partial minimizers of the non-convex learning problem. We also show that under certain conditions, the algorithm converges to the set of stationary points of the overall objective. When applied to images, the algorithm learns a collection of well-conditioned square transforms, and a good clustering of patches or textures. The resulting sparse representations for the images are much better than those obtained with a single learned transform, or with analytical transforms. We show the promising performance of the proposed approach in image denoising, which compares quite favorably with approaches involving a single learned square transform or an overcomplete synthesis dictionary, or gaussian mixture models. The proposed denoising method is also faster than the synthesis dictionary based approach.

Keywords Sparsifying transform learning · Dictionary learning · Convergence guarantees · Overcomplete representation · Clustering · Image representation · Sparse representation · Image denoising · Machine learning

1 Introduction

The sparsity of signals and images in a certain transform domain or dictionary has been heavily exploited in signal and image processing. It is well known that natural signals and images have an essentially sparse representation (few significant non-zero coefficients) in analytical transform domains such as discrete cosine transform (DCT) and Wavelets (Mallat 1999). This property has been used in designing various compression algorithms and in compression standards such as JPEG2000 (Marcellin et al. 2000).

Well-known models for sparsity include the synthesis, the analysis (Elad et al. 2007), and the transform models (Pratt et al. 1969; Ravishankar and Bresler 2013c). Each of these models can be used to generate sparse representations, or sparse codes for a signal. However, sparse coding in the transform model is computationally much cheaper than in the other models. Recent research has focused on adaptation of sparse models to data (Aharon et al. 2006; Rubinstein et al. 2012; Ravishankar and Bresler 2012b, 2013c), which turns out to be advantageous in applications. In particular, the learning

Communicated by Julien Mairal, Francis Bach, and Michael Elad.

Saiprasad Ravishankar and Bihan Wen have contributed equally to this work.

B. Wen (✉) · S. Ravishankar · Y. Bresler
Department of Electrical and Computer Engineering
and the Coordinated Science Laboratory, University of Illinois,
Urbana-Champaign, IL 61801, USA
e-mail: bwen3@illinois.edu

S. Ravishankar
e-mail: ravisha3@illinois.edu

Y. Bresler
e-mail: ybresler@illinois.edu

of transform models has been shown to be much cheaper than synthesis, or analysis learning. Adaptive transforms also provide better signal/image reconstruction quality in applications (Ravishankar and Bresler 2013d, b; Pfister 2013; Pfister and Bresler 2014).

In this work, we present a union of transforms model, and show that it can represent the diverse features, or textures seen in natural images much more accurately than a single sparsifying transform. In applications, the learning of this model also turns out to be more advantageous than prior (sparse) learning-based approaches.

In the following, we discuss the various sparse models and their learning in more detail. The discussion will highlight the various drawbacks of prior models. Our own contributions are then discussed.

1.1 Sparse Models and Their Learning

The *synthesis model* suggests that a real-world signal $y \in \mathbb{R}^n$ satisfies $y = Dx + e$ with $D \in \mathbb{R}^{n\times m}$ a synthesis dictionary, $x \in \mathbb{R}^m$ sparse, and e an approximation term in the signal domain (Bruckstein et al. 2009). We say that $x \in \mathbb{R}^m$ is sparse if $\|x\|_0 \ll m$, where the l_0 quasi norm counts the number of non-zero entries in x. When $m = n$ and D is full rank, it forms a basis. When $m > n$, the fat matrix D is said to be an overcomplete dictionary.

Given a signal y and dictionary D, the well-known synthesis sparse coding problem finds the sparse code x that minimizes $\|y - Dx\|_2^2$ subject to $\|x\|_0 \leq s$, where s is the required sparsity level. This synthesis sparse coding problem is however, NP-hard (Non-deterministic Polynomial-time hard) (Natarajan 1995; Davis et al. 1997). Various algorithms (Pati et al. 1993; Mallat and Zhang 1993; Gorodnitsky et al. 1995; Harikumar and Bresler 1996; Chen et al. 1998; Efron et al. 2004; Dai and Milenkovic 2009) have been proposed to solve this problem. While some of these algorithms are guaranteed to provide the correct solution under certain conditions, these conditions are often violated in applications. Furthermore, these algorithms are typically computationally expensive when used for large-scale problems in image processing and computer vision.

Adapting the synthesis model turns out to be advantageous in applications. Learned synthesis dictionaries have been demonstrated to be useful in applications such as denoising, inpainting, deblurring, and demosaicing (Elad and Aharon 2006; Mairal et al. 2008a; Aharon and Elad 2008; Mairal et al. 2008b). Additionally, inverse problems such as those in Tomography (Liao and Sapiro 2008), and magnetic resonance imaging (MRI) (Ravishankar and Bresler 2011a, b) benefit from an adaptive synthesis model.

The synthesis dictionary learning problem has been studied in many recent papers (Olshausen and Field 1996; Engan et al. 1999; Aharon et al. 2006; Yaghoobi et al. 2009). Given a matrix $Y \in \mathbb{R}^{n\times N}$ whose columns represent training signals, the problem of learning an adaptive dictionary D that gives a sparse representation for the signals in Y can be formulated as follows (Aharon et al. 2006).

$$\min_{D,X} \|Y - DX\|_F^2 \;\; s.t. \;\; \|X_i\|_0 \leq s \;\; \forall\, i \tag{P0s}$$

Here, the subscript i indexes the ith column of a matrix. The columns of the matrix $X \in \mathbb{R}^{m\times N}$ denote the sparse codes of the columns (or, signals) of Y, and s denotes the sparsity level allowed for each training signal. Various algorithms have been proposed for synthesis dictionary learning (Engan et al. 1999; Aharon et al. 2006; Yaghoobi et al. 2009; Skretting and Engan 2010; Mairal et al. 2010; Sahoo and Makur 2013; Smith and Elad 2013; Sadeghi et al. 2013), that typically alternate between a *sparse coding step* (solving for, or updating X), and a *dictionary update step* (solving for D[1]). Among these various algorithms, the K-SVD method (Aharon et al. 2006) has been particularly popular and demonstrated to be useful in numerous applications such as denoising, inpainting, deblurring, and MRI. However, since (P0s) is non-convex and NP-hard, methods such as K-SVD[2] can get easily caught in local minima, or saddle points (Rubinstein et al. 2010).

While the synthesis model has received enormous attention, the *analysis* approach (Elad et al. 2007) has also been gaining traction recently. The analysis model suggests that a signal $y \in \mathbb{R}^n$ satisfies $\|\Omega y\|_0 \ll m$, where $\Omega \in \mathbb{R}^{m\times n}$ is known as the analysis dictionary. A more general *noisy signal analysis model* (Rubinstein and Elad 2011; Rubinstein et al. 2012) has also been studied, where the signal $y \in \mathbb{R}^n$ is modeled as $y = z + e$ with $\Omega z \in \mathbb{R}^m$ sparse, i.e., $\|\Omega z\|_0 \ll m$. Here, e is a noise term that is assumed to be small in the signal domain. Given the noisy signal y and analysis dictionary Ω, the *analysis sparse coding* problem (Rubinstein et al. 2012) finds z by minimizing $\|y - z\|_2^2$ subject to $\|\Omega z\|_0 \leq m - l$, where l is referred to as the cosparsity level (number of zeros) (Rubinstein et al. 2012). This problem too is NP-hard and similarly to sparse coding in the synthesis model, approximate algorithms exist for analysis sparse coding (Chambolle 2004; Rubinstein and Elad 2011; Nam et al. 2011; Candès et al. 2011; Rubinstein et al. 2012; Liu et al. 2012; Yaghoobi et al. 2012; Giryes et al. 2014). The learning of analysis dictionaries has also been studied in several recent papers (Peyré and Fadili 2011; Yaghoobi et al. 2011; Ophir et al. 2011; Rubinstein and Elad 2011; Chen et al. 2012a; Rubinstein et al. 2012; Hawe et al. 2013; Yaghoobi et al. 2013). The analysis

[1] Some algorithms (e.g., K-SVD) also update the non-zero coefficients of the sparse code X in the dictionary update step.

[2] In fact, the K-SVD method, although popular, does not have any convergence guarantees.

learning problems are typically non-convex and NP-hard, and the various learning algorithms tend to be computationally expensive.

Very recently (Ravishankar and Bresler 2013c), a generalized analysis model called the *transform model* has been studied, which 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, i.e., $\|x\|_0 \ll m$. Here, e is the approximation error, which is assumed to be small. The distinguishing feature from the synthesis and from the noisy analysis models, is that this approximation error is in the transform rather than in the signal domain.

When $m = n$, the transform $W \in \mathbb{R}^{n\times n}$ is called a square transform. On the other hand, for $m > n$, the transform is called a tall or overcomplete transform. Various analytical transforms are known to approximately sparsify natural signals, such as the discrete cosine transform (DCT), Wavelets (Mallat 1999), Ridgelets (Candès and Donoho 1999), Contourlets (Do and Vetterli 2005), and Curvelets (Candès and Donoho 1999). The transform model has been shown to be more general than both the analysis and the noisy signal analysis models (Ravishankar and Bresler 2013c).

When a sparsifying transform W is known for the signal y, *transform sparse coding* finds a sparse code x of sparsity s by minimizing the sparsification error $\|Wy - x\|_2^2$ subject to $\|x\|_0 \leq s$. This problem is easy and its solution is obtained exactly by zeroing out all but the s coefficients of largest magnitude in the vector Wy. In contrast, sparse coding with synthesis or analysis dictionaries involves solving NP-hard problems approximately. Given W and sparse code x, one can also recover a least squares estimate of the signal y by minimizing the residual $\|Wy - x\|_2^2$ over y. The recovered signal is $W^\dagger x$, with $W^\dagger$ denoting the pseudo-inverse of W.

Adapting the transform to data provides advantages in many applications (Ravishankar and Bresler 2013c, 2012a, 2013b, d; Pfister 2013; Pfister and Bresler 2014). Learnt transforms provide better signal/image representations than analytical transforms such as the DCT or Wavelets. Moreover, compared to the synthesis and analysis models, the transform model allows for exact and extremely fast computations. Transform learning formulations also do not involve highly non-convex functions involving the product of multiple unknown matrices (such as in Problem (P0s)). Thus, in spite of its apparent similarity to the analysis and the noisy analysis model, the transform model enjoys important advantages over the noisy analysis or synthesis models, whether as a fixed or data-driven, adaptive model.

A drawback of the current transform learning problems and algorithms is that they are restricted to the case of square transforms. For natural images with highly complicated structures, a single learned square transform may not provide sufficient sparsification.

1.2 Our Contributions

1.2.1 Structured Overcomplete Transform Model

We investigate a union of square sparsifying transforms model in this work. In this model, we consider a collection (union) of square transforms $\{W_i\}_{i=1}^K$. A candidate signal is said to match (or, belong to) a particular transform in the collection if that transform provides the *best sparsification* for the signal among all the transforms in the collection.

A motivation for the proposed model is that natural signals and images (even if they belong to a single class such as MRI images, music signals, etc.) need not be sufficiently sparsifiable by a single transform. For example, image patches from different regions of a natural image usually contain different features, or textures. Thus, having a union of transforms would allow groups of patches with common features (or, textures) to be better sparsfied by their own texture-specific transform.

We will show that this union of square transforms model can be interpreted as an overcomplete sparsifying transform model with an additional constraint of block cosparsity for the transform sparse code. Here, the overcomplete transform is formed by stacking the transforms in $\{W_i\}_{i=1}^K$ on top of each other. For the sake of brevity, we will also refer to our OverComplete TransfOrm model with BlOck coSparsity constraint as the OCTOBOS model. In the remainder of this paper, we will use the terms ‘union of transforms’, or ‘OCTOBOS’ interchangeably, depending on the context.

1.2.2 Highlights

We enumerate some important features of our work as follows.

(i) Sparse coding in the proposed OCTOBOS model reduces to a form of clustering and is computationally inexpensive.
(ii) In this work, we propose a novel problem formulation and algorithm for learning structured overcomplete sparsifying transforms with block cosparsity constraint. Our algorithm is an alternating minimization algorithm, and each step of the algorithm involves simple (computationally cheap) closed-form solutions.
(iii) We present a novel convergence guarantee for the proposed alternating OCTOBOS learning algorithm. We prove global convergence (i.e., convergence from any initialization) of the algorithm to the set of partial minimizers of the objective defining the problem. We also show that under certain conditions, the algorithm converges to the set of stationary points of the overall objective.
(iv) Our adapted OCTOBOS model provides a better sparse representation of images than adaptive single square

transforms and analytical transforms such as the DCT.

(v) We present an adaptive image denoising formulation and algorithm exploiting the OCTOBOS model in this work. The denoising performance of the proposed approach is better than that obtained using adaptive square transforms, or adaptive overcomplete synthesis dictionaries (K-SVD). Our denoising scheme also performs better than the well-known Gaussian Mixture Model (GMM) approach (Zoran and Weiss 2011), and is comparable to the state-of-the-art BM3D denoising (Dabov et al. 2007) in some cases.

1.2.3 Related Work

A model similar to the union of transforms, but involving instead a union of orthogonal synthesis dictiionaries (PCAs) has been recently used by Peleg and Elad (2014) for the task of single image super-resolution. Also, for the specific task of super-resolution, Wang et al. (2012) learn a union of coupled synthesis dictionaries.

The learning of a union of synthesis dictionaries with the main goal of unsupervised classification has been previously proposed in a number of works (Ramirez et al. 2010; Kong and wang 2012; Chen et al. 2012b). The learning of structured synthesis dictionary models (with block, or group sparsity) for tasks such as classification has also been explored (Sprechmann et al. 2012a, b; Zelnik-Manor et al. 2012; Chi et al. 2013).

Similar to prior work on dictionary learning, these various formulations tend to be highly non-convex, and these approaches suffers from the high computational cost associated with sparse coding in the synthesis model. In contrast, eliminating the NP hard sparse coding or its approximation, our proposed OCTOBOS learning scheme has a low computational cost.

In this work, we only briefly discuss the possibility of classification (or, segmentation) using our proposed transform learning scheme. Indeed, the classification application is not the focus of this work, and a detailed study of this application will be considered for future work. Instead, to illustrate the usefulness of the learned union of transforms/OCTOBOS model, we focus in this work on applications such as image representation and denoising. We also provide convergence guarantees for OCTOBOS learning. Such guarantees are not available for the methods that learn a union of synthesis dictionaries, or block-sparse synthesis models.

1.2.4 Organization

The proposed union of square transforms model and its various alternative interpretations are described in Sect. 2. Section 2 also discusses the prior work on (single) square transform learning and introduces our newly proposed learning formulation. In Sect. 3, we describe our algorithm for learning the proposed structured overcomplete sparsifying transform, and discuss the algorithm's computational properties. In Sect. 4, we provide a convergence analysis for our transform learning algorithm. The application of our transform learning framework to image denoising is discussed in Sect. 5. We then present experimental results demonstrating the convergence behavior, and promising performance of our proposed approach in Sect. 6. In Sect. 7, we conclude with proposals for future work.

2 OCTOBOS Model and Learning Formulation

2.1 The Union of Transforms Model

The square sparsifying transform model has been investigated (Ravishankar and Bresler 2013c) recently. Here, we extend the single square transform model to a union of transforms model, which suggests that a signal $y \in \mathbb{R}^n$ is approximately sparsifiable by a particular transform in the collection $\{W_k\}_{k=1}^K$, where $W_k \in \mathbb{R}^{n \times n} \, \forall \, k$ are themselves square transforms. Thus, there exists a particular W_k such that $W_k y = x + e$, with $x \in \mathbb{R}^n$ sparse, and a transform residual e that is sufficiently small.

Given a signal $y \in \mathbb{R}^n$, and a union (or, collection) of square transforms $\{W_k\}_{k=1}^K$, we need to find the best matching transform (or, model) for the signal, that gives the smallest sparsification error. This can be formulated as the following sparse coding problem:

$$\min_{1 \leq k \leq K} \; \min_{z^k} \left\| W_k y - z^k \right\|_2^2 \quad s.t. \; \left\| z^k \right\|_0 \leq s \;\; \forall \; k \tag{P1}$$

Here, z^k denotes the sparse representation of y in the transform W_k, with the maximum allowed sparsity level being s. We assume that the W_k's are all identically scaled in (P1). Otherwise, they can be rescaled (for example, to unit spectral or Frobenius norm) prior to solving (P1).

In order to solve (P1), we first find the optimal sparse code $\hat{z}^k$ for each[3] k as $\hat{z}^k = H_s(W_k y)$, where the operator $H_s(\cdot)$ is the projector onto the s-ℓ_0 ball, i.e., $H_s(b)$ zeros out all but the s elements of largest magnitude in $b \in \mathbb{R}^n$. If there is more than one choice for the s coefficients of largest magnitude in a vector b, which can occur when multiple entries in b have identical magnitude, then we choose $H_s(b)$ as the projection of b for which the indices of the s largest magnitude elements in b are the lowest possible. Now, Problem (P1) reduces to

[3] For each k, this is identical to the single transform sparse coding problem.

$$\min_{1 \leq k \leq K} \|W_k y - H_s(W_k y)\|_2^2 \tag{1}$$

To solve the above problem, we compute the sparsification error (using the optimal sparse code above) for each k and choose the best transform $W_{\hat{k}}$ as the one that provides the smallest sparsfication error (among all the W_k's). This is an exact solution technique for Problem (P1). Problem (P1) then also provides us with an optimal sparse code $\hat{z}^{\hat{k}} = H_s(W_{\hat{k}} y)$ for y. Given such a sparse code, one can also recover a signal estimate by minimizing $\left\| W_{\hat{k}} y - \hat{z}^{\hat{k}} \right\|_2^2$ over all $y \in \mathbb{R}^n$. The recovered signal is then given by $\hat{y} = W_{\hat{k}}^{-1} \hat{z}^{\hat{k}}$.

Since Problem (P1) matches a given signal y to a particular transform, it can be potentially used to cluster a collection of signals according to their transform models. The sparsification error term in (1) can be viewed as a clustering measure in this setting. This interpretation of (P1) indicates the possible usefulness of the union of transforms model in applications such as classification.

2.2 The Overcomplete Transform Model Interpretation

We now propose an interpretation of the union of transforms model as a structured overcomplete transform model (or, the OCTOBOS model). The 'equivalent' overcomplete transform is obtained from the union of transforms by stacking the square sub-transforms as $W = \left[W_1^T \mid W_2^T \mid \cdots \mid W_K^T \right]^T$. The tall matrix $W \in \mathbb{R}^{m \times n}$, with $m = Kn$, and thus, $m > n$ (overcomplete transform) for $K > 1$.

The signal y is assumed to obey the model $Wy = x + e$, where the $x \in \mathbb{R}^m$ is assumed to be "block cosparse", and e is a small residual. The block cosparsity of x is defined here using the following ℓ_0-type norm:

$$\|x\|_{0,s} = \sum_{k=1}^{K} I\left(\left\| x^k \right\|_0 \leq s \right) \tag{2}$$

Here, $x^k \in \mathbb{R}^n$ is the block of x corresponding to the transform W_k in the tall W, and s is a given sparsity level (equivalently $n - s$ is the given cosparsity level) for block x^k. The operator $I(\cdot)$ above is an indicator function with $I(Q) = 1$ when statement Q is true, and $I(Q) = 0$ otherwise. We say that x is 1-block cosparse if there is exactly one block of x with at least $n - s$ zeros, i.e., $\|x\|_{0,s} = 1$ in this case.

In the proposed overcomplete transform model for signal y, we formulate the following sparse coding problem, to which we refer as the OCTOBOS sparse coding problem.

$$\min_{x} \|Wy - x\|_2^2 \quad s.t. \quad \|x\|_{0,s} \geq 1 \tag{P2}$$

Problem (P2) finds an x with at least one block that has $\geq n - s$ zeros. In particular, we now prove the following proposition, that the Problems (P1) and (P2) are equivalent. This equivalence is the basis for the interpretation of the union of transforms model as an overcomplete transform model.

Proposition 1 *The minimum values of the sparsification errors in Problems (P1) and (P2) are identical. The optimal sparse code(s) in (P1) is equal to the block(s) of the optimal $\hat{x}$ in (P2) satisfying $\left\| \hat{x}^k \right\|_0 \leq s$.*

Proof The objective in (P2) is $\sum_{k=1}^{K} \left\| W_k y - x^k \right\|_2^2$. The constraint in (P2) calls for $\left\| x^k \right\|_0 \leq s$ for at least one k. Assume without loss of generality that in the optimal solution $\hat{x}$ of (P2), the blocks $\hat{x}^k$ satisfying the constraint $\left\| \hat{x}^k \right\|_0 \leq s$ have indices $1, \ldots, J$ (for some $J \geq 1$). Otherwise, we can always trivially permute the blocks $\hat{x}^k$ in $\hat{x}$ (and the corresponding W_k in W) so that the optimal indices are $1, \ldots, J$. Now, for fixed optimal block indices, Problem (P2) reduces to the following Problem.

$$\min_{\{x^k\}} \sum_{k=1}^{K} \left\| W_k y - x^k \right\|_2^2$$
$$s.t. \quad \left\| x^k \right\|_0 \leq s, \ \text{for } k = 1, \ldots, J \tag{3}$$

Here, the notation $\{x^k\}$ denotes the set of x^k for $1 \leq k \leq K$[4]. Since the blocks with indices $k > J$ are not selected by the sparsity constraint, for $k > J$ the optimal $\hat{x}^k = W_k y$, because this setting results in a zero (minimal) contribution to the objective (3) above. Problem (3) then decouples into J independent (square) transform sparse coding problems. It is then obvious that the minimum in (3) is achieved with $J = 1$ (minimum possible J), i.e., we only choose one block as $\hat{x}^k = H_s(W_k y)$, and all other blocks satisfy $\hat{x}^k = W_k y$. This setting leads to a zero contribution to the objective of (3) for all but one block. The chosen active block is the one that provides the smallest (minimizes (3)) individual sparsification error. It is now obvious (by directly comparing to the sparse coding algorithm for (P1)) that the proposition is true.

Note that if $\|W_k y\|_0 \leq s$ holds for one or more k, then the optimal $\hat{x} = Wy$ in (P2). Only in this degenerate case, it is possible for the optimal $\hat{x}$ in (P2) to have more than one block that is s-sparse (i.e., $\left\| \hat{x} \right\|_{0,s} > 1$ occurs if $\|W_k y\|_0 \leq s$ for two or more k). In this case, the optimal sparse code in (P1) can be set to be equal to any of the optimal s-sparse blocks in $\hat{x} = Wy$. The minimum sparsification error is zero for both (P1) and (P2) in this degenerate case. □

The optimal $\hat{x}$ in (P2) by itself cannot be called a sparse code, since (based on the proof of Proposition 1) it typically has many more non-zeros than zeros.[5] However, the particu-

[4] In the remainder of the paper, when certain indexed variables are enclosed within braces, it means that we are considering the set of variables over the range of all the indices.

[5] For example, when vector Wy has no zeros, then the optimal $\hat{x}$ in (P2) has exactly $n - s \ll Kn$ (for large K) zeros—all the zeros are concentrated in a single block of $\hat{x}$.

lar s-sparse block(s) of $\hat{x}$ can be considered as a sparse code, and one could also recover a signal estimate from this code similar to the union of transforms case.[6] Note that the many non-zeros in $\hat{x}$ help keep the overcomplete transform residual small.

The OCTOBOS model enforces a block cosparsity constraint. Alternatively, one could consider the model $Wy = x + e$ with a tall transform $W \in \mathbb{R}^{Kn \times n}$, but without any block cosparsity constraint on x, and assuming that x has at least $n-s$ zeros, i.e., $\|x\|_0 \leq (K-1)n+s$. The sparse coding in this model would be identical to thresholding (zeroing out the $n-s$ elements of smallest magnitude of) Wy. However, it is unclear how to easily combine the non-zeros and zeros to form a length n sparse code.[7] Therefore, we do not pursue this case (non-block cosparse model) in this work.

2.3 An OCTOBOS Optimality Property

Here, we consider two data matrices $Y_1 \in \mathbb{R}^{n \times N}$ and $Y_2 \in \mathbb{R}^{n \times M}$ (columns of the matrices represent signals), each of which is sparsified by a different square transform. We provide a condition under which using just one of the two transforms for both Y_1 and Y_2 will increase the total sparsification error (computed over all signals in Y_1 and Y_2). Thus, when the proposed condition holds, the union of transforms provides a better model for the collection of data compared to any one transform.

The proposed condition is based on the spark property (Donoho and Elad 2003). For a matrix $A \in \mathbb{R}^{n \times r}$, the spark is defined to be the minimum number of columns of A that are linearly dependant.

Proposition 2 *Given two sets of data $Y_1 \in \mathbb{R}^{n \times N}$ and $Y_2 \in \mathbb{R}^{n \times M}$, suppose there exist non-identical and non-singular square transforms $W_1, W_2 \in \mathbb{R}^{n \times n}$, that exactly sparsify the datasets as $W_1 Y = X_1$ and $W_2 Y_2 = X_2$, where the columns of both X_1 and X_2 have sparsity $\leq s$. If* spark $\left[W_1^{-1} \mid W_2^{-1}\right] > 2s$*, then the columns of $W_2 Y_1$ have sparsity $> s$.*

Proof Consider an arbitrary column, say the ith one, of Y_1, which we denote as z. Let $\alpha_i^1 = W_1 z$. We then have that $\left\|\alpha_i^1\right\|_0 \leq s$. Let us denote $W_2 z$ by α_i^2. We then have that

$$\left[W_1^{-1} \mid W_2^{-1}\right] \begin{bmatrix} \alpha_i^1 \\ -\alpha_i^2 \end{bmatrix} = B\alpha_i = 0, \tag{4}$$

where $B = \left[W_1^{-1} \mid W_2^{-1}\right]$, and α_i is the vertical concatenation of α_i^1 and $-\alpha_i^2$. Now, if matrix B has spark $> 2s$, then the linear combination of any $\leq 2s$ of its columns cannot equal zero. Therefore, under the spark assumption, we must have $\|\alpha_i\|_0 > 2s$. Since, $\left\|\alpha_i^1\right\|_0 \leq s$, we must then have $\left\|\alpha_i^2\right\|_0 > s$, under the spark assumption. □

If the spark condition above holds, then the sparsification errors of the columns of Y in W_2 (using sparsity level s) are strictly positive. We can also derive an alternative condition that involves the mutual coherence of $B = \left[W_1^{-1} \mid W_2^{-1}\right]$. The mutual coherence of the matrix B (Bruckstein et al. 2009) is defined as follows.

$$\mu(B) = \max_{1 \leq k, j \leq m, k \neq j} \frac{\left|B_k^T B_j\right|}{\|B_k\|_2 \cdot \left\|B_j\right\|_2} \tag{5}$$

Unlike the spark, the mutual coherence is easy to compute, and characterizes the dependance between the columns (indexed by the subscripts j and k in (5)) of matrix B. It is known that the spark and mutual coherence of a matrix B are related as follows (Bruckstein et al. 2009).

$$\text{spark}(B) \geq 1 + \frac{1}{\mu(B)} \tag{6}$$

Therefore, in Proposition 2, the spark condition can be replaced by the following (more stringent) sufficient condition involving the mutual coherence of B.

$$\mu(B) < \frac{1}{2s-1} \tag{7}$$

If the above condition holds, then by Eq. (6), the spark condition of Proposition 2 automatically holds, and thus we will have that the columns of $W_2 Y_1$ have sparsity $> s$.

The spark-based sufficient condition in Proposition 2 can be interpreted as a similarity measure between the models W_1 and W_2. In the extreme case, when $W_1 = W_2$, the aforementioned matrix B has minimum possible spark (=2). In broad terms, if W_1 and W_2 are sufficiently different, as measured by the spark condition in Proposition 2, or the coherence condition in (7), then the union of transforms model, or OCTOBOS provides a better model than either one of the transforms alone.

The difference between W_1 and W_2 as measured by the spark, or coherence conditions is invariant to certain transformations. In particular, if W_1 is an exact full rank sparsifier of matrix Y_1, then one can also obtain equivalent transforms by permuting the rows of W_1, or by pre-multiplying W_1 with a diagonal matrix with non-zero diagonal entries. All these equivalent transforms sparsify Y_1 equally (i.e., provide the same sparsity level of s) well. It is easy to see that if the condition spark $\left[W_1^{-1} \mid W_2^{-1}\right] > 2s$ (or, alternatively, the mutual coherence-based condition) holds with respect to a particular W_1 and W_2 in Proposition 2, then it also automatically holds with respect to any other equivalent W_1 and equivalent W_2.

[6] More precisely, the index of the sparse block is also part of the sparse code. This adds just $\log_2 K$ bits per index to the sparse code.

[7] We need a length n code in a square and invertible sub-transform of W, in order to perform signal recovery uniquely.

2.4 Square Transform Learning

Consider a training data matrix $Y \in \mathbb{R}^{n \times N}$ whose columns are the given training signals, and a sparse code matrix $X \in \mathbb{R}^{n \times N}$ whose columns are the sparse representations (or, sparse codes) of the corresponding columns of Y in a sparsifying transform $W \in \mathbb{R}^{n \times n}$. Previous work (Ravishankar and Bresler 2013c) proposed to learn a (single) square sparsifying transform W and sparse code X that minimize the sparsification error given by $\|WY - X\|_F^2$. The sparsification error is the modeling error in the transform model, and hence we minimize it in order to learn the best possible transform model. Analytical transforms such as the Wavelets and DCT are known to provide low sparsification errors for natural images. The single square transform learning problem was proposed as follows

$$\min_{W,X} \|WY - X\|_F^2 + \lambda Q(W) \qquad \text{(P3)}$$
$$s.t. \;\; \|X_i\|_0 \leq s \;\; \forall \; i,$$

where $Q(W) = -\log\,|\det W| + \|W\|_F^2$. Here, the subscript i denotes the ith column of the sparse code matrix X. Problem (P3) has $Q(W)$[8] as an additional regularizer in the objective to prevent trivial solutions. The log determinant penalty enforces full rank on W and eliminates degenerate solutions such as those with zero, or repeated rows. The $\|W\|_F^2$ penalty helps remove a ‘scale ambiguity in the solution (the scale ambiguity occurs when the data admits an exactly sparse representation). Together, the log determinant and Frobenius norm penalty terms help control the condition number of the learnt transform. Badly conditioned transforms typically convey little information and may degrade performance in applications.

It was shown (Ravishankar and Bresler 2013c) that the condition number $\kappa(W)$ can be upper bounded by a monotonically increasing function of $Q(W)$. Hence, minimizing $Q(W)$ encourages reduction of condition number. Given a normalized transform W and a scalar $a \in \mathbb{R}$, $Q(aW) \to \infty$ as the scaling $a \to 0$ or $a \to \infty$. Thus, $Q(W)$ also penalizes bad scalings. Furthermore, when $\lambda \to \infty$ in (P3), the condition number of the optimal transform(s) tends to 1, and the spectral norm (or, scaling) tends to $1/\sqrt{2}$.

We set $\lambda = \lambda_0 \|Y\|_F^2$ in (P3), where λ_0 is a constant. This setting makes Problem (P3) invariant to the scaling of the data Y as follows. When the data Y is replaced with aY ($a \in \mathbb{R}$, $a \neq 0$) in (P3), we can set $X = aX'$. Then, the objective function for this case becomes $a^2 \left(\|WY - X'\|_F^2 + \lambda_0 \|Y\|_F^2 \; Q(W) \right)$. Since, this is just a scaled version of the objective in (P3), the minimization of it over (W, X') (with sparse X') yields the same solution as (P3). Thus, the learnt transform for data aY is the same as for Y, while the the learnt sparse code for aY is a times that for Y. This makes sense since the sparsifying transform for a dataset is not expected to change, when the data is trivially scaled. Thus, the setting $\lambda = \lambda_0 \|Y\|_F^2$ achieves scale invariance for the solution of (P3).[9]

2.5 OCTOBOS Learning Formulations and Properties

2.5.1 *Problem Formulations*

Similar to the square sparsifying transform learning problem, we propose the following OCTOBOS learning formulation that learns a tall sparsifying transform $W \in \mathbb{R}^{Kn \times n}$ and sparse code matrix $X \in \mathbb{R}^{Kn \times N}$ from training data $Y \in \mathbb{R}^{n \times N}$.

$$\min_{W,X} \|WY - X\|_F^2 + Q'(W) \qquad \text{(P4)}$$
$$s.t. \;\; \|X_i\|_{0,s} \geq 1 \;\; \forall \; i$$

Here, $\|X_i\|_{0,s}$ is defined as in Eq. (2). The function $Q'(W)$ is defined as follows.

$$Q'(W) = \sum_{k=1}^{K} \lambda_k Q(W_k) \qquad (8)$$

The regularizer $Q'(W)$ controls the condition number of the sub-blocks of W, and λ_k are positive weights.

One can also formulate the transform learning problem in the union of transforms model as follows.

$$\min_{\{W_k, X_i, C_k\}} \sum_{k=1}^{K} \left\{ \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2 + \lambda_k Q(W_k) \right\} \qquad \text{(P5)}$$
$$s.t. \;\; \|X_i\|_0 \leq s \;\; \forall \;\; i, \;\; \{C_k\} \in G$$

Here, $X_i \in \mathbb{R}^{n \times n}$ and the set $\{C_k\}_{k=1}^K$ indicates a clustering of the training signals $\{Y_i\}_{i=1}^N$. The cluster C_k contains the indices i corresponding to the signals Y_i in the kth cluster. The signals in the kth cluster are matched to transform W_k. The set G is the set of all possible partitions of the set of integers $[1 : N] \triangleq \{1, 2, ..., N\}$, or in other words, G is the set of all possible $\{C_k\}$, and is defined as follows.

$$G = \left\{ \{C_k\} : \bigcup_{k=1}^{K} C_k = [1 : N], \; C_j \bigcap C_k = \emptyset, \; \forall \, j \neq k \right\}$$

The constraint involving G thus enforces the various C_k in $\{C_k\}_{k=1}^K$ to be disjoint, and their union to contain

[8] The weights on the log-determinant and Frobenius norm terms are set to the same value in this paper.

[9] On the other hand, if λ is a fixed constant, there is no guarantee that the optimal transforms for scaled and un-scaled Y in (P3) are related.

the indices for all training signals. Note that the term $\sum_{k=1}^{K}\sum_{i\in C_k}\|W_k Y_i - X_i\|_2^2$ in (P5) is the sparsification error for the data Y in the union of transforms model.

The weights λ_k is (P4) and (P5) are chosen as $\lambda_k = \lambda_0 \left\| Y_{C_k} \right\|_F^2$, where Y_{C_k} is a matrix whose columns are the signals of Y in the kth cluster. The rationale for this choice of λ_k is similar to that presented earlier for the λ weight in (P3). Specifically, when the clusters $\{C_k\}_{k=1}^{K}$ are fixed to their optimal values in (P5), the optimization problem (P5) reduces to K square transform learning problems of the form of (P3), each involving a particular data matrix Y_{C_k}. Thus, the setting $\lambda_k = \lambda_0 \left\| Y_{C_k} \right\|_F^2$ achieves scale invariance for the solutions of these K problems. The setting also implies that λ_k itself is a function of the unknown C_k (function of the signal energy in cluster C_k) in the optimization problem (P5). When the $\{W_k\}$ are fixed, the $Q'(W)$ penalty in (P5) encourages a larger concentration of data energy ($\left\| Y_{C_k} \right\|_F^2$) in the cluster corresponding to a smaller $Q(W_k)$ (i.e., corresponding to smaller condition number and reasonable scaling).

2.5.2 Properties of Formulations (P4) and (P5)

The following result implies that the learning Problems (P4) and (P5) are equivalent.

Proposition 3 *The minimum values of the objectives in Problems (P4) and (P5) are identical. Moreover, any optimal union of transforms in (P5) can be vertically concatenated to form an optimal overcomplete W for (P4). Similarly, any optimal W in (P4) can be used to generate an optimal union of transforms for (P5).*

Proof Let $\left\{W_k, C_k, X_{C_k}\right\}$ be a global minimizer of (P5). Then, we can form an equivalent overcomplete W by vertically stacking the W_k's. Moreover, we can form/construct a tall sparse code matrix $X' \in \mathbb{R}^{Kn\times n}$ by letting each column X_i' be equal to X_i on one block (according to the clustering C_k), and equal to $W_k Y_i$ on the other blocks. The constructed (W, X') is feasible for (P4), and provides a value for the (P4) objective that is equal to the minimum objective value attained in (P5). Thus, the minimum objective value in (P4) can only be lower than the minimum value in (P5).

Similarly, given an optimal minimizer (W, X) for (P4), we can form $\{W_k\}$ as the blocks of W. The $\{C_k\}$ and $\{X_i\}$ parts of (P5) can also be constructed from X using Proposition 1. The constructed $\{W_k\}$, $\{C_k\}$, $\{X_i\}$ is feasible for (P5), and provides a value for the (P5) objective that is clearly equal to the minimum objective value obtained in (P4). Since, the minimum in (P5) is computed over all feasible $\{W_k\}$, $\{C_k\}$, $\{X_i\}$, it can be only lower than the minimum objective value in (P4).

By the preceding arguments, it is clear that the minimum values of the objectives in (P4) and (P5) must in fact, be identical. The rest of the proposition also follows from the above arguments and construction techniques. □

Although (P4) and (P5) are equivalent, Problem (P5) is more intuitive and amenable to alternating optimization schemes (see Sect. 3 for such a scheme). If we were to alternate between updating X and W in (P4), we would not be able to directly maintain (without additional constraints) the property that the transform domain residual for each Y_i is zero in all but (at most) one block of W, during the update of W. This is not a problem for (P5), since its objective only considers the residual of each Y_i in one (best) block.

The following result indicates that the minimum objective in the union of transforms Problem (P5) is always lower than the minimum objective value in the single transform learning problem (P3). This means that either the optimal sparsification error in (P5) is lower than the corresponding value in (P3), or the optimal regularizer (that controls the condition number(s) of the transform block(s)) in (P5) is smaller than the corresponding value in (P3), or both of these conditions hold.

Proposition 4 *The minimum value of the objective in (P5) can only be lower than the minimum objective value in (P3).*

Proof Let $(\hat{W}, \hat{X})$ denote an optimal minimizer of (P3), i.e., it provides the minimum value of the objective of (P3). Now, in (P5), we can set $W_k = \hat{W}\ \forall\, k$, and $X_i = \hat{X}_i\ \forall\, i$ in the objective. For this setting, the objective in (P5) becomes identical (using the fact that $\sum_{k=1}^{K}\lambda_k = \lambda$) to the minimum value of the objective in (P3). This result is invariant to the specific choice of the C_k's. Now, since the minimum value of the objective in (P5) is attained over all feasible $\{W_k\}$, $\{X_i\}$, $\{C_k\}$, it can only be lower ($\leq$) than the value obtained with the specific settings above. □

We will empirically illustrate in Sect. 6 that our algorithm for (P5) (discussed in Sect. 3) provides a lower value of both the objective and sparsification error compared to the algorithm for (P3).

It was shown by Ravishankar and Bresler (2013c) that the objective of Problem (P3) is lower bounded. The following lemma confirms that the objectives of the proposed learning formulations are lower bounded too.

Lemma 1 *The objectives in Problems (P4) and (P5) are both lower bounded by $\lambda Q_0 = \lambda_0 Q_0 \|Y\|_F^2$, where $Q_0 = \frac{n}{2} + \frac{n}{2}\log(2)$.*

Proof The objectives in (P4) and (P5) are the summation of a sparsification error term (net error over all signals) and a $Q'(W) = \sum_{k=1}^{K}\lambda_k Q(W_k)$ regularizer term. The sparsification error term is lower bounded by 0. Each $Q(W_k)$ regularizer is bounded as $Q(W_k) \geq Q_0 = \frac{n}{2} + \frac{n}{2}\log(2)$ (cf. Ravishankar and Bresler 2013c for proof of this). Thus,

$Q'(W) \geq Q_0 \sum_{k=1}^{K} \lambda_k = \lambda_0 \, Q_0 \, \|Y\|_F^2$, where we used the setting $\lambda_k = \lambda_0 \left\|Y_{C_k}\right\|_F^2$. Thus, the objectives in (P4) and (P5) are both lower bounded by $\lambda Q_0 = \lambda_0 \, Q_0 \, \|Y\|_F^2$. □

We use the preceding lemma to prove the following proposition, which pertains to the identifiability of good models (models that sparsify well and are well-conditioned) by our Problem (P5). Proposition 5 also pertains to the case when the lower bounds in Lemma 1 are achievable.

Proposition 5 *Given a training data matrix $Y \in \mathbb{R}^{n \times N}$, let $\left\{Y_{C_k}\right\}$ be a collection of data matrices formed according to a clustering rule $\{C_k\}$. Suppose that $\{W_k\}$ is a collection of unit conditioned square transform models, each with spectral norm $1/\sqrt{2}$,*[10] *that exactly sparsifies the clustered data $\left\{Y_{C_k}\right\}$ as $W_k Y_{C_k} = X_{C_k} \; \forall \, k$, with each X_{C_k} having s-sparse columns. Then, the set $\left\{W_k, \; C_k, \; X_{C_k}\right\}$ is a global minimizer of* (P5)*, i.e., the underlying model is identifiable by solving* (P5).

Proof The objective in (P5) is the summation of a sparsification error term and a $Q'(W)$ regularizer term. Since, $\{W_k\}$ exactly sparsify the clustered data $\left\{Y_{C_k}\right\}$, the sparsification error in (P5) is zero (minimum) at the given $\left\{W_k, \; C_k, \; X_{C_k}\right\}$. The regularizer term $Q'(W) = \sum_{k=1}^{K} \lambda_k Q(W_k)$ only depends on the $\{W_k\}$. It was shown by Ravishankar and Bresler (2013c) that $Q(W_k) \geq Q_0$, with equality if and only if W_k is unit conditioned, and the singular values of W_k are all equal to $\sqrt{\frac{1}{2}}$. Since, each W_k considered here achieves $Q(W_k) = Q_0$, we have that the regularizer $Q'(W)$ attains its lower bound $\lambda Q_0 = \lambda_0 \, Q_0 \, \|Y\|_F^2$ mentioned in Lemma 1, for the considered $\{W_k\}$. Thus, we have shown that the objective in (P5) attains its lower bound for the given $\left\{W_k, \; C_k, \; X_{C_k}\right\}$. In other words, the objective attains its global minimum in this case. □

Thus, when an "error-free" union of transforms model exists for the data, and the transforms are all unit conditioned, Proposition 5 guarantees that such a union of transforms model is a global minimizer of the proposed Problem (P5). Therefore, it makes sense to solve (P5) in order to find such good OCTOBOS models.

We now show that the role of the λ_0 weight in (P5) is to control the condition number and scaling of the transform blocks W_k $(1 \leq k \leq K)$. If we were to minimize only the $\hat{Q}(W) = Q'(W)/\lambda_0 = \sum_{k=1}^{K} \left\|Y_{C_k}\right\|_F^2 \, Q(W_k)$ regularizer in Problem (P5) with respect to the unknowns, then the minimum value would be $Q_0 \, \|Y\|_F^2$ according to Lemma 1. This minimum is achieved with W_k's that are unit conditioned, and with spectral norm of $1/\sqrt{2}$ (i.e., transforms with identical scaling). Thus, similar to Corollary 2 in (Ravishankar and Bresler 2013c), we have that as $\lambda_0 \to \infty$ in (P5), the condition number of the optimal transforms in (P5) tends to 1, and their spectral norm (scaling) tends to $1/\sqrt{2}$. Therefore, as $\lambda_0 \to \infty$, our formulation (P5) approaches a union of unit-conditioned transforms learning problem. We also empirically show in Sect. 6 that when λ_0 is properly chosen (but finite), the condition numbers and norms of the learnt W_k's in (P5) are very similar. Note that we need the W_k's to be similarly scaled for the sparsification error in (P5) to be fully meaningful (since otherwise, a certain W_k with a very small scaling can trivially give the best sparsification error for a signal).

Another interesting fact about OCTOBOS learning is that both (P4) and (P5) admit an equivalence class of solutions similar to (P3). For example, one can permute the rows within an optimal block W_k (along with a permuation of the corresponding sparse codes), or pre-multiply W_k by a diagonal ±1 sign matrix (and multiply the sparse codes accordingly), without affecting its optimality. In (P4), one can also permute the blocks W_k within an optimal W (and correspondingly permute the sparse codes) to produce equivalent optimal solutions.

We note that in spite of sharing the common theme of a mixture of models, our OCTOBOS model and learning formulation are quite different from the Gaussian Mixture Model (GMM) approach of Zoran and Weiss (2011), and Yu et al. (2012). In the GMM-based models, the signal can be thought of (cf. Zoran and Weiss 2011) as approximated by a linear combination of a few (orthonormal) eigenvectors of the covariance matrix of the mixture component to which it belongs. In contrast, in the OCTOBOS approach, the transform blocks W_k (equivalently, the class-conditional square sparsifying transforms) are not eigenvectors of some covariance matrices. Instead they are directly optimized (via (P5)) for transform-domain sparsity of the training data. Our OCTOBOS learning also enforces well-conditioning rather than exact orthonormality of the transform blocks. These features distinguish our OCTOBOS framework from the GMM-based approach.

3 Transform Learning Algorithm and Properties

3.1 Algorithm

We propose an alternating algorithm to solve the joint minimization Problem (P5). In one step of our proposed algorithm called the *sparse coding and clustering step*, we solve for $\{C_k\}$, $\{X_i\}$ with fixed $\{W_k\}$ in (P5). In the other step of the algorithm called the *transform update step*, we solve for the transforms $\{W_k\}$ in (P5) with fixed sparse codes.

[10] If the transforms have a different spectral norm, they can be trivially scaled to have spectral norm $1/\sqrt{2}$.

3.1.1 Sparse Coding and Clustering

Given the training matrix Y, and fixed transforms $\{W_k\}$ (or, the equivalent overcomplete W), we solve the following Problem (P6) (which is just (P5) with fixed transforms) to determine the sparse codes and clusters. As before, the clusters are disjoint and every training signal belongs to exactly one cluster.

$$\min_{\{C_k\},\{X_i\}} \sum_{k=1}^{K} \sum_{i \in C_k} \left\{ \|W_k Y_i - X_i\|_2^2 + \eta_k \|Y_i\|_2^2 \right\} \tag{P6}$$
$$s.t. \;\; \|X_i\|_0 \leq s \;\; \forall \;\; i, \quad \{C_k\} \in G$$

The weight $\eta_k = \lambda_0 \, Q(W_k)$ above. This is a fixed weight, since W_k is fixed in this step. We refer to the term $\|W_k Y_i - X_i\|_2^2 + \eta_k \|Y_i\|_2^2$, with $X_i = H_s(W_k Y_i)$ (i.e., the optimal sparse code of Y_i in transform W_k) as a clustering measure corresponding to the signal Y_i. This is a modified version of the measure in (P1), and includes the additional penalty $\eta_k \|Y_i\|_2^2$ determined by the regularizer (i.e., determined by the conditioning of W_k[11]). It is easy to observe that the objective in (P6) involves the summation of only N such 'clustering measure' terms (one for each signal). Since every training signal is counted exactly once (in one cluster) in the double summation in Problem (P6), we can construct the equivalent optimization problem as follows.

$$\sum_{i=1}^{N} \min_{1 \leq k \leq K} \left\{ \|W_k Y_i - H_s(W_k Y_i)\|_2^2 + \eta_k \|Y_i\|_2^2 \right\} \tag{9}$$

The minimization over k for each Y_i above determines the cluster C_k (in (P6)) to which Y_i belongs. For each Y_i, the optimal cluster index $\hat{k}_i$[12] is such that $\left\| W_{\hat{k}_i} Y_i - H_s(W_{\hat{k}_i} Y_i) \right\|_2^2 + \eta_{\hat{k}_i} \|Y_i\|_2^2 \leq \left\| W_j Y_i - H_s(W_j Y_i) \right\|_2^2 + \eta_j \|Y_i\|_2^2, \forall j \neq \hat{k}_i$. The optimal $\hat{X}_i$ in (P6) is then $H_s\left(W_{\hat{k}_i} Y_i\right)$. There is no coupling between the sparse coding/clustering problems in (9) for the different training signals $\{Y_i\}_{i=1}^{N}$. Thus, the training signals can be sparse coded and clustered, in parallel.

3.1.2 Transform Update Step

Here, we solve for $\{W_k\}$ in (P5) with fixed $\{C_k\}$, $\{X_i\}$. Although this is an unconstrained joint minimization problem over the set of transforms, the optimization problem is actually separable (due to the objective being in summation form) into K unconstrained problems, each involving only a particular square transform W_k. Thus, the transform update problem becomes

$$\min_{W_k} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2 + \lambda_k Q(W_k) \tag{P7}$$

Here, $\lambda_k = \lambda_0 \left\| Y_{C_k} \right\|_F^2$ is a fixed weight. Problem (P7) is solved separately for each k, which can be done in parallel. Problem (P7) is similar to the transform update problem encountered for (P3) (Ravishankar and Bresler 2013a), and can thus be solved similarly.

Let U be the matrix whose columns are the training signals Y_i belonging to the kth cluster (i.e., $i \in C_k$). Let V be the corresponding (fixed) sparse code matrix. Problem (P7) can then be solved exactly and efficiently by using the simple closed-form solution proposed by Ravishankar and Bresler (2013a). First, we decompose the positive-definite matrix $UU^T + \lambda_k I$ as $UU^T + \lambda_k I = LL^T$ (e.g., by Cholesky decomposition, or taking the eigen value decomposition (EVD) square root), where I is the $n \times n$ identity. Next, we obtain the full singular value decomposition (SVD) of the matrix $L^{-1} U V^T$ as $B \Sigma R^T$, where B, Σ, and R are all $n \times n$ matrices. Then, the optimal transform $\hat{W}_k$ in (P7) is given as

$$\hat{W}_k = \frac{R}{2} \left(\Sigma + \left(\Sigma^2 + 2\lambda_k I \right)^{\frac{1}{2}} \right) B^T L^{-1}, \tag{10}$$

where the $(\cdot)^{\frac{1}{2}}$ operation above denotes the positive definite square root. The closed-form solution (10) is guaranteed to be a global optimum of Problem (P7). Compared to iterative optimization methods such as conjugate gradients (CG), (10) allows for both cheap and exact computation of the solution of the transform update problem. The OCTOBOS learning Algorithm A1 is summarized in Fig. 1. Algorithm A1 assumes that an initial estimate $\left\{ \hat{W}_k^0, \hat{X}_i^0, \hat{C}_k^0 \right\}$ is available (see Section 6.2 for examples of initializations). The initial $\left\{ \hat{W}_k^0 \right\}$ is only used by the algorithm in a degenerate scenario mentioned later (see footnote 19).

3.2 Computational Cost

The algorithm for (P5) consists of the sparse coding and clustering step, and the transform update step. We derive the computational cost of each of these steps.

Sparse coding and clustering. First, the sparse code of every training signal with respect to every transform W_k is computed. The computation of $W_k Y$ requires $n^2 N$ multiply-add operations. The projection of $W_k Y$ on to the s-ℓ_0 ball if done by full sorting would require $O(nN \log n)$ operations. Thus, the cost of finding the sparse representation of the training matrix in a particular W_k is dominated by $O(n^2 N)$. Since this needs to be done for each k, the total number of oper-

[11] This clustering measure will encourage the shrinking of clusters corresponding to any badly conditioned, or badly scaled transforms.

[12] When two or more clusters are equally optimal, then we pick the one corresponding to the lowest cluster index k.

```
OCTOBOS Learning Algorithm A1
Input : $Y$ - training matrix with $N$ signals, $s$ - sparsity level, $\lambda_0$ - constant, $K$ - number of clusters, $J$ - number of iterations.
Output : $\left\{\hat{W}_k\right\}$ - learnt transforms, $\left\{\hat{X}_i\right\}$ - learnt sparse codes, $\left\{\hat{C}_k\right\}$ - learnt clusters.
Initial Estimates: $\left\{\hat{W}_k^0, \hat{X}_i^0, \hat{C}_k^0\right\}$.
For $t = 1$ : J Repeat
1) Transform Update: For each $1 \leq k \leq K$, do
    (a) Let $\Psi \triangleq \hat{C}_k^{t-1}$. Compute $\lambda_k = \lambda_0 \|Y_\Psi\|_F^2$.
    (b) Compute $L^{-1} = \left(Y_\Psi Y_\Psi^T + \lambda_k I\right)^{-1/2}$.
    (c) Compute full SVD of $L^{-1} Y_\Psi (\hat{X}_\Psi^{t-1})^T$ as $B \Sigma R^T$.
    (d) $\hat{W}_k^t = \frac{R}{2}\left(\Sigma + (\Sigma^2 + 2\lambda_k I)^{\frac{1}{2}}\right) B^T L^{-1}$.
2) Sparse Coding and Clustering: For $1 \leq i \leq N$,
    (a) If $i = 1$, set $\hat{C}_k^t = \emptyset \ \forall \ k$, and compute $\eta_k = \lambda_0 \, Q(\hat{W}_k^t)$, $1 \leq k \leq K$.
    (b) Compute $\gamma_k = \eta_k \|Y_i\|_2^2 + \left\|\hat{W}_k^t Y_i - H_s(\hat{W}_k^t Y_i)\right\|_2^2$, $1 \leq k \leq K$. Set $\hat{k} = \min\{k : \gamma_k = \min_k \gamma_k\}$. Set $\hat{C}_{\hat{k}}^t \leftarrow \hat{C}_{\hat{k}}^t \cup \{i\}$.
    (c) $\hat{X}_i^t = H_s\left(\hat{W}_{\hat{k}}^t Y_i\right)$.
End
```

Fig. 1 Algorithm A1 for OCTOBOS learning via (P5). A superscript of t is used to denote the iterates in the algorithm

ations scales as $O(Kn^2N)$. Denoting $m = Kn$ this cost is $O(mnN)$.

Next, for each training signal, in order to perform clustering, the clustering measure, which is the sum of the sparsification error and weighted signal energy, is computed with respect to all the transforms. The total cost of computing the sparsification error (taking into account that the sparsification error is only computed using the transform domain residual on the complement of the support of the sparse code) for all training signals in all clusters is $O((n-s)KN)$. Since the weighted signal energy term $\lambda_0 \|Y_i\|_2^2 \; Q(W_k)$ needs to be computed for all i, k, we first compute $\lambda_0 \, Q(W_k)$ for all k at a cost of $O(Kn^3)$ (computing determinants dominates this cost). Next, the term $\|Y_i\|_2^2$ is computed for all i at a cost of $O(nN)$. Then, computing $\lambda_0 \|Y_i\|_2^2 \; Q(W_k)$ and adding it to the corresponding sparsification error term for all i, k, requires $O(KN)$ operations. Finally, to compute the best cluster for each signal requires $O(K-1)$ comparisons (between the clustering measure values for different transforms), and thus $O((K-1)N)$ operations for the entire Y matrix. Assuming, $N \gg n$, it is clear based on the preceding arguments that the computation of $\{W_kY\}$ dominates the computations in the sparse coding and clustering step, with a cost of $O(mnN)$ (or, $O(Kn^2N)$).

Table 1 Computational cost per-iteration for square sparsifying transform, OCTOBOS, and KSVD learning

	Square trans.	OCTOBOS	KSVD
Cost	$O(n^2N)$	$O(mnN)$	$O(mn^2N)$

Transform update. In this step, we compute the closed-form solution for the transform in each cluster using (10). First, the matrix $UU^T + \lambda_k I_n$ (notations defined in Sect. 3.1.2) needs to be computed for each (disjoint) cluster. This requires $O(n^2N)$ multiply-add operationas totally (over all clusters). (Note that the computation of all the λ_k's requires only $O(nN)$ operations.) The computation of L and L^{-1} requires $O(n^3)$ operations for each cluster, and thus about $O(Kn^3)$ operations for K clusters. Next, the matrix UV^T is computed for each cluster. Since V has s-sparse columns, this matrix multiplication gives rise to a total (over all clusters) of $\alpha n^2 N$ multiply-add operations (assuming $s = \alpha n$, with $\alpha < 1$). Finally, the computation of $L^{-1}UV^T$, its SVD, and the closed-form update (10) require $O(n^3)$ operations per cluster, or about $O(Kn^3)$ operations for K clusters. Since, $N \gg m = Kn$ typically, we have that the cost of the transform update step scales as $O(n^2N)$. Thus, for $K \gg 1$, the transform update step is cheaper than the sparse coding step for the proposed algorithm.

Based on the preceding arguments, it is clear that the computational cost per iteration (of sparse coding and transform update) of our algorithm scales as $O(mnN)$.[13] This is much lower (in order) than the per-iteration cost of learning an $n \times m$ overcomplete synthesis dictionary D using K-SVD (Aharon et al. 2006), which, (assuming, as in the transform model, that the synthesis sparsity level $s = \beta n$ with $\beta < 1$[14]), scales as $O(mn^2N)$. Our transform learning also holds a similar (per-iteration) computational advantage over analysis dictionary learning schemes such as analysis K-SVD. The computational costs per-iteration of square transform, OCTOBOS, and KSVD learning are summarized in Table 1.

As illustrated in our experiments in Sect. 6, both the OCTOBOS and square transform learning algorithms converge in few iterations in practice. Therefore, the per-iteration computational advantages for OCTOBOS over K-SVD typically translate to a net computational advantage in practice.

OCTOBOS learning could be used for a variety of purposes including clustering (classification), denoising, and

[13] Setting $m = n$ for the case $K = 1$, this agrees with previous cost analysis for square transform learning using (P3), which has per-iteration cost of $O(n^2N)$ (Ravishankar and Bresler 2013c).

[14] The notion that sparsity s scales with the signal dimension n is rather standard. For example, while $s = 1$ may work for representing the 4×4 patches of an image in a DCT dictionary with $n = 16$, the same sparsity level of $s = 1$ for an $n = 256^2$ DCT dictionary for a 256×256 (vectorized) image would lead to very poor image representation. Therefore, the sparsity s must increase with the size n. A typical assumption is that the sparsity s scales as a fraction (e.g., 5 or 10 %) of the image or, patch size n. Otherwise, if s were to increase only sub-linearly with n, it would imply that larger (more complex) images are somehow better sparsifiable, which is not true in general.

sparsity-based signal compression. In the latter case, we also need to compute $\hat{Y}_i = W_k^{-1} X_i$, for all $i \in C_k$, and $\forall\, k$, in order to recover estimates of the signals from their (compressed) transform codes. Computing W_k^{-1}, $1 \leq k \leq K$, has $O(Kn^3)$ computational cost. However, this cost does not depend on the number of training signals $N \gg Kn$ (typically), and is therefore negligible compared to the total cost $O(snN)$ $(= O(n^2 N)$ for $s \propto n)$ of multiplying the once computed W_k^{-1} for all k, with the corresponding sparse codes. The latter cost is the same as for multiplying a synthesis dictionary D with its sparse codes.

4 Convergence Analysis

In this section, we analyze the convergence behavior of the proposed OCTOBOS learning algorithm, that solves (P5). While some recent works (Spielman et al. 2012; Agarwal et al. 2013; Arora et al. 2013; Xu and Yin 2013; Bao et al. 2014; Agarwal et al. 2014) study the convergence of (specific) synthesis dictionary learning algorithms,[15] none of them consider the union of dictionaries case. The prior works also typically require many restrictive assumptions (e.g., noiseless data) for their results to hold. In contrast, we present a novel convergence theory here for learning a union of (transform) sparse models. Importantly, although our proposed Problem formulation (P5) is highly non-convex (due to the ℓ_0 quasi norm on the sparse codes and the log-determinant penalty), our results hold with few or no assumptions.

Very recently, a convergence result (Ravishankar and Bresler 2014) has been derived for the algorithm for the (single) square transform learning Problem (P3). Here, we derive the results for the general overcomplete OCTOBOS case.

4.1 Main Results

The convergence results are more conveniently stated in terms of an unconstrained form of (P5). Problem (P5) has the constraint $\|X_i\|_0 \leq s\, \forall\, i$, which can equivalently be added as a penalty in the objective by using a barrier function $\phi(X)$ (where $X \in \mathbb{R}^{n\times N}$ is the matrix of all sparse codes), which takes the value $+\infty$ when the constraint is violated, and is zero otherwise. Then, we denote the objective of (P5) as

$$g\left(W, X, \Gamma\right) = \sum_{k=1}^{K} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2 + \phi(X) + \sum_{k=1}^{K} \lambda_k Q(W_k), \quad (11)$$

where $W \in \mathbb{R}^{Kn\times n}$ is obtained by stacking the W_k's on top of one another, the matrix $X \in \mathbb{R}^{n\times N}$ contains the sparse codes X_i as its columns, and the row vector $\Gamma \in \mathbb{R}^{1\times N}$ is such that its ith element $\Gamma_i \in \{1, \ldots, K\}$ denotes the cluster index (label) corresponding to the signal Y_i. As discussed in Sect. 2.5, the clusters $\left\{C_k\right\}$ form a disjoint partitioning of $[1 : N]$.

Problem (P5) is to find the best possible union of sparsifying transforms model for a given set of data Y, by minimizing the sparsification error, and controlling the condition numbers (avoiding trivial solutions) of the cluster-specific transforms. Proposition 5 in Sect. 2.5 established the identifiability of good models by solving Problem (P5). Here, we discuss the convergence behavior of our algorithm A1 that solves (P5).

For our convergence results, we make only the following mild assumption.

Assumption 1 Our proposed solution involves computing SVDs, EVDs (of small $n \times n$ matrices), and scalar square roots. Although in practice these quantities are computed using iterative methods, we assume, for the theoretical analysis, that they are computed exactly. In practice, standard numerical methods are guaranteed to quickly provide machine precision accuracy for these computations.

Since the training matrix $Y \in \mathbb{R}^{n\times N}$, the training signals are also bounded, i.e., $\max_i \|Y_i\|_2 < \infty$. For a vector z, let $\beta_j(z)$ denote the magnitude of the jth largest element (magnitude-wise) of z. For a matrix B, $\|B\|_\infty \triangleq \max_{i,j} \left|B_{ij}\right|$. We say that a sequence $\left\{b^t\right\}$ has an accumulation point b, if there is a subsequence that converges to b. For our iterative Algorithm A1 (in Fig. 1), we denote the iterates (or, outputs) at each iteration t by the set (W^t, X^t, Γ^t), where W^t denotes the matrix obtained by stacking the cluster-specific transforms W_k^t $(1 \leq k \leq K)$, X^t is the sparse code matrix with the sparse codes X_i^t $(1 \leq i \leq N)$ as its columns, and Γ^t is a row vector containing the cluster indices Γ_i^t $(1 \leq i \leq N)$ as its elements. Each Γ_i^t contains the cluster index corresponding to signal Y_i. The following theorem provides a convergence result for the OCTOBOS learning Algorithm A1.

Theorem 1 *Let $\left\{W^t, X^t, \Gamma^t\right\}$ denote the iterate sequence generated by Algorithm A1 with training data Y and initial (W^0, X^0, Γ^0). Then, the objective sequence $\left\{g^t\right\}$ with $g^t \triangleq g\left(W^t, X^t, \Gamma^t\right)$ is monotone decreasing, and converges to a finite value, say $g^* = g^*(W^0, X^0, \Gamma^0)$. The iterate sequence is bounded, and all its accumulation points are equivalent in the sense that they achieve the same value g^* of the objective. Finally, every accumulation point (W, X, Γ) is a fixed point of Algorithm A1 satisfying the following partial global optimality conditions*

[15] Most of these (synthesis dictionary learning) algorithms have not been demonstrated to be practically useful in applications such as denoising. Bao et al. (2014) show that their method denoises worse than the K-SVD method (Elad and Aharon 2006).

$$(X, \Gamma) \in \underset{\tilde{X}, \tilde{\Gamma}}{\arg\min}\ g\left(W, \tilde{X}, \tilde{\Gamma}\right) \tag{12}$$

$$W \in \underset{\tilde{W}}{\arg\min}\ g\left(\tilde{W}, X, \Gamma\right) \tag{13}$$

as well as the following partial local optimality property.

$$g\,(W + dW, X + \Delta X, \Gamma) \geq g\,(W, X, \Gamma) \tag{14}$$

Property (14) *holds for all* dW *with sufficiently small perturbations to the individual cluster-specific transforms* $dW_k \in \mathbb{R}^{n\times n}$ *satisfying* $\|dW_k\|_F \leq \epsilon_k$ *for some* $\epsilon_k > 0$ *that depends on the specific* W_k*, and the following condition on* ΔX*. For every* $1 \leq k \leq K$*, let* $\Delta X_{C_k} \in \mathbb{R}^{n\times|C_k|}$ *be the matrix with columns* $\Delta X_i \in \mathbb{R}^n$ *for* $i \in C_k$*. Then,* ΔX *is such that* $\Delta X_{C_k} \in \mathrm{R1_k} \cup \mathrm{R2_k}$ *for all* k*, where*

R1$_k$*: The half-space tr* $\left\{(W_k Y_{C_k} - X_{C_k})\Delta X_{C_k}^T\right\} \leq 0.$
R2$_k$*: The* local region *defined by*
$\left\|\Delta X_{C_k}\right\|_\infty < \min_{i\in C_k}\{\beta_s(W_k Y_i) : \|W_k Y_i\|_0 > s\}.$

Furthermore, if we have $\|W_k Y_i\|_0 \leq s\, \forall\, i \in C_k$*, then the corresponding* ΔX_{C_k} *can be arbitrary.*

The local region R2$_k$ in Theorem 1 that determines the size of the local perturbation ΔX_{C_k} in class k, is determined by the scalar $\gamma_k = \min_{i\in C_k}\{\beta_s(W_k Y_i) : \|W_k Y_i\|_0 > s\}$. This scalar is computed by (i) taking the columns of $W_k Y$ corresponding to the kth cluster; (ii) choosing only the vectors with sparsity $> s$; (iii) finding the sth largest magnitude element of those vectors; and (iv) picking the smallest of those values.

Theorem 1 indicates that for a particular starting point (W^0, X^0, Γ^0), the iterate sequence in our algorithm converges to an equivalence class of accumulation points. Every accumulation point has the same cost $g^* = g^*(W^0, X^0, \Gamma^0)$[16], and is a fixed point of the algorithm, as well as a partial local minimizer (with respect to the cluster transforms and sparse code variables) of the objective $g\,(W, X, \Gamma)$. Since Algorithm A1 minimizes the objective $g\,(W, X, \Gamma)$ by alternating between the minimization over (X, Γ) and W, and obtains the global optimum in each of these minimizations, it follows that the algorithm's fixed points satisfy the partial global optimality conditions (12) and (13).

Thus, we can also say that, for each initial (W^0, X^0, Γ^0), the iterate sequence in OCTOBOS converges to an equivalence class of fixed points, or an equivalence class of partial local/global minimizers satisfying (12)–(14). This is summarized by the following Corollary.

Corollary 1 *For a particular initial* (W^0, X^0, Γ^0)*, the iterate sequence in Algorithm A1 converges to an equivalence class of fixed points, that are also partial minimizers satisfying* (12)–(14).

The following corollary summarizes the convergence of Algorithm A1 for any starting point (the phrase "globally convergent" refers to convergence from any initialization).

Corollary 2 *The iterate sequence in Algorithm A1 is globally convergent to the set of partial minimizers of the non-convex objective* $g\,(W, X, \Gamma)$.

We would like to emphasize that unlike results in previous work (for synthesis learning), Theorem 1 that shows the convergence of the proposed non-convex OCTOBOS learning algorithm is free of any restrictive conditions or requirements. Theorem 1 also holds for any choice of the parameter λ_0 in (P5), which controls the condition number of the cluster transforms. The condition (14) in Theorem 1 holds true not only for local (or small) perturbations in the sparse codes, but also for arbitrarily large perturbations of the sparse codes in a half space, as defined by region R1$_k$.

While Theorem 1 shows partial local/global optimality for Algorithm A1, the following Theorem 2 establishes that, under certain conditions, every accumulation point of the iterate sequence in Algorithm A1 is a stationary point of the overall objective. Algorithm A1 then converges to the set of stationary points of the overall problem.

Equation (9) indicates that the objective that is minimized in Problem (P5) can be equivalently written as

$$f\,(W) = \sum_{i=1}^{N} \min_k \left\{ \|W_k Y_i - H_s(W_k Y_i)\|_2^2 \right. \\ \left. + \lambda_0\, Q(W_k)\, \|Y_i\|_2^2 \right\} \tag{15}$$

This equivalent objective is now only a function of the transforms $\{W_k\}$ (with the cluster assignment being implicit). Our OCTOBOS learning algorithm can be thought of as an alternating minimization algorithm to minimize the function $f(W)$, that also involves the optimization with respect to the additionally introduced sparse code and cluster index variables.

We now state Theorem 2 in terms of the equivalent objective $f(W)$.

Theorem 2 *Let* $\left\{W^t, X^t, \Gamma^t\right\}$ *denote the iterate sequence generated by Algorithm A1 with training data* Y *and initial* (W^0, X^0, Γ^0)*. Let each accumulation point* (W, X, Γ) *of the iterate sequence be such that* (X, Γ) *is the* unique *minimizer of* $g\left(W, \tilde{X}, \tilde{\Gamma}\right)$ *for fixed* W*. Then, every accumulation point of the iterate sequence is a stationary point of the objective* $f(W)$.

Theorem 2 establishes that the iterates converge to the set of stationary points of $f(W)$. It assumes that for every

[16] The exact value of g^* may vary with initialization. We will empirically illustrate in Sect. 6.2 that our algorithm is also insensitive to initialization.

accumulation point (W, X, Γ), the pair (X, Γ) is the *unique* minimizer of $g\left(W, \tilde{X}, \tilde{\Gamma}\right)$ for fixed W. Note that the condition $(X, \Gamma) \in \arg\min_{\tilde{X},\tilde{\Gamma}} g\left(W, \tilde{X}, \tilde{\Gamma}\right)$ is already guaranteed by Theorem 1. Only the uniqueness of the sparse coding and clustering solution is additionally assumed in Theorem 2, i.e., we assume that there are no ties in assigning the clusters or sparse codes.

Although the result in Theorem 2 depends on the uniqueness condition, the following conjecture postulates that provided the following Assumption 2 (that uses a probabilistic model for the data) holds, the uniqueness condition holds with probability 1, i.e., the probability of a tie in assigning the cluster or sparse code is zero.

Assumption 2 The training signals $Y_i \in \mathbb{R}^n$ for $1 \leq i \leq N$, are drawn independently from an absolutely continuous probability measure over the n-dimensional ball $\hat{S} \triangleq \left\{y \in \mathbb{R}^n : \|y\|_2 \leq c_0\right\}$ for some $c_0 > 0$.

Conjecture 1 *Let Assumption 2 hold. Then, with probability 1, every accumulation point* (W, X, Γ) *of Algorithm A1 is such that* (X, Γ) *is the* unique *minimizer of* $g\left(W, \tilde{X}, \tilde{\Gamma}\right)$ *for fixed* W.

Conjecture 1 is motivated in Sect. 4.2.3. If Conjecture 1 holds, then every accumulation point of the iterate sequence in Algorithm A1 is immediately a stationary point of $f(W)$ with probability 1.

4.2 Proofs

We 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 defined as $\{x \in \mathbb{R}^n : \|x\|_0 \leq s\}$.

We now prove the following properties of Algorithm A1 one-by-one.

(i) The objective sequence in Algorithm A1 converges.
(ii) The sequence of iterates is bounded.
(iii) The iterate sequence has an accumulation point.
(iv) All the accumulation points of the iterate sequence have a common objective value.
(v) Every accumulation point of the iterate sequence is a fixed point of the algorithm satisfying the partial global optimality conditions (12) and (13).
(vi) Every fixed point of the algorithm is a local minimizer of $g(W, X, \Gamma)$ with respect to the transforms $\{W_k\}$ and sparse codes $\{X_i\}$.
(vii) Under the uniqueness condition stated in Theorem 2, every accumulation point is a stationary point of the equivalent objective $f(W)$.

Items (i)–(vi) above pertain to Theorem 1 and establish the various results in Theorem 1, while item (vii) pertains to Theorem 2.

4.2.1 *Proof of Theorem 1*

Our first lemma shows the convergence of the objective sequence in Algorithm A1.

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

Proof In the transform update step, we solve K independent unconstrained problems. For each k, we obtain a global minimizer with respect to W_k in (P7). The closed-form solution is given in (10). Since, by Assumption 1, we obtain an exact solution in the transform update step, the objective decreases in this step, i.e., $g\left(W^{t+1}, X^t, \Gamma^t\right) \leq g\left(W^t, X^t, \Gamma^t\right)$. Furthermore, as discussed in Sect. 3.1.1, in the sparse coding and clustering step too, we obtain an exact solution with respect to $\{C_k\}$ and $\{X_i\}$ (for fixed cluster transforms). Therefore, $g\left(W^{t+1}, X^{t+1}, \Gamma^{t+1}\right) \leq g\left(W^{t+1}, X^t, \Gamma^t\right)$. Combining the results for the two steps, we get

$$g\left(W^{t+1}, X^{t+1}, \Gamma^{t+1}\right) \leq g\left(W^t, X^t, \Gamma^t\right) \tag{16}$$

By Lemma 1 in Sect. 2.5, the objective in (P5) is lower bounded. Since, the objective is monotone decreasing and lower bounded, it converges. □

The next lemma establishes the boundedness of the iterate sequence.

Lemma 3 *The iterate sequence generated by Algorithm A1 is bounded.*

Proof For each t, the iterate is $\left(W^t, X^t, \Gamma^t\right)$. Clearly, $1 \leq \Gamma_i^t \leq K$. Therefore, it is obvious that Γ_i^t is bounded by K for any i and all t.

Now, consider the triplet $\left(W^t, X^{t-1}, \Gamma^{t-1}\right)$. Since $g(W^t, X^{t-1}, \Gamma^{t-1}) = \sum_{k=1}^{K} \sum_{i \in C_k^{t-1}} \left\| W_k^t Y_i - X_i^{t-1} \right\|_2^2 + \sum_{k=1}^{K} \lambda_0 \left\| Y_{C_k^{t-1}} \right\|_F^2 Q(W_k^t)$ (note that $\phi(X) = 0$ for the iterates) is a sum of non-negative terms[17], we have that for any k,

$$\lambda_0 \left\| Y_{C_k^{t-1}} \right\|_F^2 Q(W_k^t) \leq g\left(W^t, X^{t-1}, \Gamma^{t-1}\right) \leq g^0, \tag{17}$$

[17] The regularizer $Q(W_k^t)$ is non-negative by the arguments in the proof of Lemma 1 in Sect. 2.5.

where the last inequality is due to the monotonic decrease of the objective (Lemma 2). Now, it could happen that at a particular iteration, the cluster C_k^{t-1} (the output of the clustering step) is empty. In such cases, we assume that in the subsequent transform update step, the transform W_k^t for the kth cluster remains fixed at W_k^{t-1}. This transform is still used in the following clustering step, and may produce a non-empty cluster C_k^t. Since W_k^t remains fixed whenever C_k^{t-1} is empty, we only need to bound it for the iterations where C_k^{t-1} is non-empty.

Now, assuming C_k^{t-1} is non-empty, we can further consider two sub-cases: (1) $\left\| Y_{C_k^{t-1}} \right\|_F = 0$; and (2) $\left\| Y_{C_k^{t-1}} \right\|_F \neq 0$. Now, when $\left\| Y_{C_k^{t-1}} \right\|_F = 0$, it means that the signals in the kth cluster are all zero. In this case, the corresponding sparse codes $X^{t-1}_{C_k^{t-1}}$ (obtained by thresholding zero signals) will also be all zero. Therefore, the objective for the kth cluster in the tth transform update step is identically zero in this case. This objective is minimized by any (transform) matrix in $\mathbb{R}^{n \times n}$. Therefore, in this case, we assume that the optimal W_k^t is set to W_k^{t-1}. Since W_k^t remains fixed in this case, we only need to consider the case when $\left\| Y_{C_k^{t-1}} \right\|_F \neq 0$. In the latter case, we have by (17) that

$$Q(W_k^t) \leq \frac{g^0}{\lambda_0 \left\| Y_{C_k^{t-1}} \right\|_F^2} \leq \frac{g^0}{\lambda_0 \eta^2}, \tag{18}$$

where the last inequality follows from $\left\| Y_{C_k^t} \right\|_F \geq \eta$, with $\eta \triangleq \min_{i:\, \|Y_i\|_2 \neq 0} \|Y_i\|_2$ being a fixed strictly positive number.

The function $Q(W_k^t) = \sum_{i=1}^n (\alpha_i^2 - \log \alpha_i)$, where α_i $(1 \leq i \leq n)$ are the (all positive) singular values of W_k^t, is a coercive function of the singular values, and therefore it has bounded lower level sets.[18] Combining this fact with (18), we get that there is a constant c (that depends only on g^0, λ_0, η) such that $\|W_k^t\|_F = \sqrt{\sum_{i=1}^n \alpha_i^2} \leq c$. Thus, the sequence $\{W^t\}$ of cluster transforms is bounded.

We now bound the sparse codes X_i^t $(1 \leq i \leq N)$ for all t. First, for each iteration t and index i, we have that there exists a k $(1 \leq k \leq K)$ such that $X_i^t = H_s\left(W_k^t Y_i\right)$ (see Fig. 1). Therefore, by the definition of $H_s(\cdot)$, we have

$$\left\|X_i^t\right\|_2 = \left\|H_s\left(W_k^t Y_i\right)\right\|_2 \leq \left\|W_k^t Y_i\right\|_2 \leq \left\|W_k^t\right\|_2 \left\|Y_i\right\|_2 \tag{19}$$

Since W_k^t (by aforementioned arguments) and Y_i are both bounded (by constants that do not depend on t), (19) implies that the sequence $\{X^t\}$ of sparse codes is also bounded. □

Proposition 6 *The iterate sequence in Algorithm A1 has at least one convergent subsequence, or in other words, it has at least one accumulation point.*

Proof Since the iterate sequence is bounded, the existence of a convergent subsequence (for a bounded sequence) is a standard result. □

The following property of the accumulation points of the iterates in our algorithm will be used to prove that all accumulation points are equivalent.

Lemma 4 *Any accumulation point (W^*, X^*, Γ^*) of the iterate sequence generated by Algorithm A1 satisfies*

$$X_i^* \in \tilde{H}_s\left(W_{\Gamma_i^*}^* Y_i\right) \quad \forall\ i \tag{20}$$

Proof Consider the subsequence $\{W^{q_t}, X^{q_t}, \Gamma^{q_t}\}$ (indexed by q_t), that converges to the accumulation point (W^*, X^*, Γ^*). We then have for each i $(1 \leq i \leq N)$

$$X_i^* = \lim_{t\to\infty} X_i^{q_t} = \lim_{t\to\infty} H_s\left(W_{\Gamma_i^{q_t}}^{q_t} Y_i\right) \tag{21}$$

where $\Gamma_i^{q_t}$ is the cluster index for Y_i at iteration q_t. Now, for each i, since the integer sequence $\left\{\Gamma_i^{q_t}\right\}$ converges to Γ_i^* (which is also an integer in $\{1, \ldots, K\}$), this implies that this convergence takes place in a finite number of iterations, i.e., there exists a $t_0 \in \mathbb{N}$ such that $\forall\ t \geq t_0$, we have $\Gamma_i^{q_t} = \Gamma_i^*$. Using this result in (21) yields

$$X_i^* = \lim_{t\to\infty} H_s\left(W_{\Gamma_i^*}^{q_t} Y_i\right) \in \tilde{H}_s\left(W_{\Gamma_i^*}^* Y_i\right) \tag{22}$$

where the containment on the right hand side of (22) follows from Lemma 10 of Appendix. Indeed, since the vector sequence $\left\{W_{\Gamma_i^*}^{q_t} Y_i\right\}$ converges to $W_{\Gamma_i^*}^* Y_i$, by Lemma 10 the accumulation point of the sequence $\left\{H_s\left(W_{\Gamma_i^*}^{q_t} Y_i\right)\right\}$ above must lie in the (possibly non-singleton) set $\tilde{H}_s\left(W_{\Gamma_i^*}^* Y_i\right)$. □

For the following lemma, we define $\hat{g}\left(W, X, \Gamma\right) \triangleq g\left(W, X, \Gamma\right) - \phi(X)$.

Lemma 5 *All the accumulation points of the iterate sequence generated by Algorithm A1 with a given initialization correspond to a common objective value g^*. Thus, they are equivalent in that sense.*

Proof Consider the subsequence $\{W^{q_t}, X^{q_t}, \Gamma^{q_t}\}$ (indexed by q_t), that converges to the accumulation point (W^*, X^*, Γ^*).

First, for each k, the matrix W_k^* is non-singular. This follows from (18) which implies that

$$-\log\left|\det W_k^t\right| \leq \frac{g^0}{\lambda_0 \eta^2} \tag{23}$$

and (23) further implies that $\left|\det W_k^t\right|$ is bounded away from zero for all t. Hence, due to the continuity of the determinant function, W_k^* (the limit point of a subsequence) is non-singular. We then have that

$$\begin{aligned}\lim_{t\to\infty} \hat{g}\left(W^{q_t}, X^{q_t}, \Gamma^{q_t}\right) &= \lim_{t\to\infty} \hat{g}\left(W^{q_t}, X^{q_t}, \Gamma^*\right) \\ &= \hat{g}\left(W^*, X^*, \Gamma^*\right), \end{aligned} \tag{24}$$

[18] The lower level sets of a function $\hat{f} : A \subset \mathbb{R}^n \mapsto \mathbb{R}$ (where A is unbounded) are bounded if $\lim_{t\to\infty} \hat{f}(x^t) = +\infty$ whenever $\left\{x^t\right\} \subset A$ and $\lim_{t\to\infty} \left\|x^t\right\| = \infty$.

where for the first equality in (24), we used the previously mentioned fact that, for each i, the convergence of the integer sequence $\left\{\Gamma_i^{q_t}\right\}$ implies that $\Gamma_i^{q_t} = \Gamma_i^*$ for sufficiently large t. The second equality in (24) follows because, for fixed cluster indices, the function $\hat{g}$, is continuous with respect to the cluster transforms and sparse code vectors at (W^*, X^*). The continuity is obvious from the fact that the sparsification error term in $\hat{g}$ is a quadratic, and the regularizer term in $\hat{g}$ is continuous at non-singular matrices W_k^*.

Next, by Lemma 4, the accumulation point X_i^* (the ith column of X^*) is s-sparse for all i. Furthermore, Algorithm A1 guarantees that $X_i^{q_t}$ is s-sparse for each t and all i. Therefore, the barrier function $\phi(X)$ is zero for the sparse code subsequence and for its accumulation point. It follows that (24) is equivalent to

$$\lim_{t\to\infty} g\left(W^{q_t}, X^{q_t}, \Gamma^{q_t}\right) = g\left(W^*, X^*, \Gamma^*\right) \tag{25}$$

because g and $\hat{g}$ coincide in these equations. Finally, by Lemma 2, the left hand side limit above is in fact g^*. Therefore, we have that for any accumulation point (W^*, X^*, Γ^*) of the iterate sequence,

$$g\left(W^*, X^*, \Gamma^*\right) = g^* \tag{26}$$

□

For a particular accumulation point (W^*, X^*, Γ^*) of the iterate sequence in our algorithm, the following result shows that the cluster index Γ_i^* is the optimal cluster index for signal Y_i with respect to the set of transforms $\left\{W_k^*\right\}$.

Lemma 6 *Any accumulation point (W^*, X^*, Γ^*) of the iterate sequence generated by Algorithm A1 satisfies for each $1 \le i \le N$ the inequality*

$$\begin{aligned}&\left\| W_{\Gamma_i^*}^* Y_i - H_s\left(W_{\Gamma_i^*}^* Y_i\right)\right\|_2^2 + \lambda_0 \left\|Y_i\right\|_2^2 Q\left(W_{\Gamma_i^*}^*\right)\\ &\quad\le \lambda_0 \left\|Y_i\right\|_2^2 Q\left(W_j^*\right) + \left\|W_j^* Y_i - H_s\left(W_j^* Y_i\right)\right\|_2^2 \;\; \forall\, j \neq \Gamma_i^*.\end{aligned} \tag{27}$$

Proof Consider the subsequence $\{W^{q_t}, X^{q_t}, \Gamma^{q_t}\}$ (indexed by q_t), that converges to the accumulation point (W^*, X^*, Γ^*). Let us pick a specific index i. The convergence of the integer sequence $\left\{\Gamma_i^{q_t}\right\}$ then implies that $\Gamma_i^{q_t} = \Gamma_i^*$ for sufficiently large $t \ge t_0$. For each $t \ge t_0$, the sparse coding and clustering step of Algorithm A1 guarantees that $\Gamma_i^{q_t} = \Gamma_i^*$ is the optimal cluster index for signal Y_i, i.e.,

$$\begin{aligned}&\left\| W_{\Gamma_i^*}^{q_t} Y_i - X_i^{q_t}\right\|_2^2 + \lambda_0 \left\|Y_i\right\|_2^2 Q\left(W_{\Gamma_i^*}^{q_t}\right)\\ &\quad\le \lambda_0 \left\|Y_i\right\|_2^2 Q\left(W_j^{q_t}\right) + \left\|W_j^{q_t} Y_i - H_s\left(W_j^{q_t} Y_i\right)\right\|_2^2 \;\; \forall\, j \neq \Gamma_i^*.\end{aligned} \tag{28}$$

We would like to take the limit $t \to \infty$ on both sides of the above inequality. Notice that the sequence $\left\{W_j^{q_t}\right\}$ converges to W_j^* for every j, and by Lemma 4, the limit point of the sequence $\left\{X_i^{q_t}\right\}$ satisfies $X_i^* \in \tilde{H}_s\left(W_{\Gamma_i^*}^* Y_i\right)$. This implies that

$$\begin{aligned}\lim_{t\to\infty} \left\| W_{\Gamma_i^*}^{q_t} Y_i - X_i^{q_t}\right\|_2^2 &= \left\| W_{\Gamma_i^*}^{*} Y_i - X_i^{*}\right\|_2^2\\ &= \left\| W_{\Gamma_i^*}^{*} Y_i - H_s\left(W_{\Gamma_i^*}^{*} Y_i\right)\right\|_2^2\end{aligned} \tag{29}$$

where the last equality in (29) follows because every sparse code in the set $\tilde{H}_s\left(W_{\Gamma_i^*}^{*} Y_i\right)$ provides the same sparsification error, or in other words, X_i^* and $H_s\left(W_{\Gamma_i^*}^{*} Y_i\right)$ provide the same sparsification error in (29). Furthermore, for a fixed j, since, by Lemma 10 of Appendix, every accumulation point of the sequence $\left\{H_s\left(W_j^{q_t} Y_i\right)\right\}$ lies in the set $\tilde{H}_s\left(W_j^* Y_i\right)$, we also easily have that

$$\lim_{t\to\infty} \left\| W_j^{q_t} Y_i - H_s\left(W_j^{q_t} Y_i\right)\right\|_2^2 = \left\| W_j^{*} Y_i - H_s\left(W_j^{*} Y_i\right)\right\|_2^2$$

Now, since W_j^* is non-singular, the regularizer term $Q\left(W_j^{q_t}\right)$ converges to $Q\left(W_j^*\right)$ for any j.

Thus, taking the limit $t \to \infty$ on both sides of (28), and using the above limits for each of the terms in (28), we immediately get the result (27) of the lemma. □

The following property of the accumulation points in our algorithm will be used to prove that every accumulation point is a fixed point. In the following lemma, C_k^* denotes the set of indices belonging to the kth cluster, for an accumulation point (W^*, X^*, Γ^*) of the iterate sequence in Algorithm A1.

Lemma 7 *Any accumulation point (W^*, X^*, Γ^*) of the iterate sequence generated by Algorithm A1 satisfies*

$$W^* \in \underset{W}{\arg\min}\; g\left(W, X^*, \Gamma^*\right) \tag{30}$$

Specifically, for each $1 \le k \le K$, we have

$$W_k^* \in \underset{W_k}{\arg\min} \sum_{i \in C_k^*} \left\|W_k Y_i - X_i^*\right\|_2^2 + \lambda_0 \left\|Y_{C_k^*}\right\|_F^2 Q(W_k) \tag{31}$$

Proof Consider the subsequence $\{W^{q_t}, X^{q_t}, \Gamma^{q_t}\}$ (indexed by q_t), that converges to the accumulation point (W^*, X^*, Γ^*). Assume without loss of generality that the sequence $\left\{W^{q_t+1}\right\}$ converges to say W^{**}. Otherwise we can work with a convergent subsequence (exists since the sequence is bounded) $\left\{W^{q_{r_t}+1}\right\}$, and the following proof technique still holds by considering the subsequence $\{W^{q_{r_t}}, X^{q_{r_t}}, \Gamma^{q_{r_t}}\}$. Note that the subsequence $\{W^{q_{r_t}}, X^{q_{r_t}}, \Gamma^{q_{r_t}}\}$ converges to the same limit (W^*, X^*, Γ^*) as the original $\{W^{q_t}, X^{q_t}, \Gamma^{q_t}\}$ above.

For (31) to hold, we need only consider k for which C_k^* is non-empty and $\left\|Y_{C_k^*}\right\|_F \neq 0$. For any other k, (31) is trivially true. Let us now pick a specific such k. Since the integer vector sequence $\{\Gamma^{q_t}\}$ converges, it follows that, there exists a $t_0 \in \mathbb{N}$ such that $\Gamma^{q_t} = \Gamma^*$ for all $t \geq t_0$. Hence,

$$C_k^{q_t} = C_k^* \quad \forall \quad t \geq t_0, \quad \forall\, k \tag{32}$$

In the remainder of this proof, we consider only $t \geq t_0$. Because of (32), the data in the kth cluster does not change over the subsequence iterations q_t for $t \geq t_0$. Hence, in the transform update step (of Algorithm A1) at iteration $q_t + 1$ for $t \geq t_0$, the computed (unique) inverse EVD square root matrix has the form

$$L^{-1} = \left(Y_{C_k^*} Y_{C_k^*}^T + \lambda_0 \left\| Y_{C_k^*} \right\|_F^2 I \right)^{-1/2}, \tag{33}$$

where the subscript C_k^* is used to denote the matrix whose columns are the signals corresponding to the indices in C_k^*.

Let $B_k^{q_t} \Sigma_k^{q_t} \left(R_k^{q_t}\right)^T$ denote the full SVD of $L^{-1} Y_{C_k^*} \left(X_{C_k^*}^{q_t}\right)^T$. The transform $W_k^{q_t+1}$ is computed in (10) in terms of the above SVD by the expression

$$W_k^{q_t+1} = \frac{R_k^{q_t}}{2} \left(\Sigma_k^{q_t} + \left(\left(\Sigma_k^{q_t}\right)^2 + 2\lambda_k I \right)^{\frac{1}{2}} \right) \left(B_k^{q_t}\right)^T L^{-1},$$

where $\lambda_k = \lambda_0 \left\| Y_{C_k^*} \right\|_F^2$.

For L^{-1} defined as in (33), and $W_k^{q_t+1}$ defined as above, and recalling the assumption that $\left\{ W_k^{q_t+1} \right\}$ converges to W_k^{**} and $\left\{ W_k^{q_t}, X_{C_k^*}^{q_t} \right\}$ converges to $\left(W_k^*, X_{C_k^*}^* \right)$ for the chosen subsequence, we have that all conditions for Lemma 11 in Appendix are satisfied. Therefore, we have

$$W_k^{**} \in \underset{W_k}{\arg\min} \sum_{i \in C_k^*} \left\| W_k Y_i - X_i^* \right\|_2^2 + \lambda_k Q(W_k) \tag{34}$$

The above result holds for every k. Therefore, since the objective $g\,(W, X^*, \Gamma^*)$ is the sum of the objectives corresponding to each cluster-wise transform, we have

$$W^{**} \in \underset{W}{\arg\min}\ g\left(W, X^*, \Gamma^*\right) \tag{35}$$

Now, by Lemma 5, we have that $g\,(W^*, X^*, \Gamma^*) = g^*$. Furthermore, applying the same arguments as previously used in (24)–(26) to the (convergent) sequence $\left\{ W^{q_t+1}, X^{q_t}, \Gamma^{q_t} \right\}$ ($\Gamma^{q_t} = \Gamma^*$ for $t \geq t_0$), we also get that $g\,(W^{**}, X^*, \Gamma^*) = g^*$. Since W^* achieves the same value of the objective (with fixed sparse codes and cluster indices) as W^{**}, and using (35), we immediately have that (30) holds. Equation (31) then trivially holds due to the separability of the objective in (30). □

Lemma 8 *Every accumulation point of the iterate sequence generated by Algorithm A1 is a fixed point of the algorithm.*

Proof Consider the subsequence $\{W^{q_t}, X^{q_t}, \Gamma^{q_t}\}$ (indexed by q_t), that converges to the accumulation point (W^*, X^*, Γ^*). We then have by Lemma 6 and Lemma 4 that

$$(X^*, \Gamma^*) \in \underset{X, \Gamma}{\arg\min}\ g\left(W^*, X, \Gamma\right) \tag{36}$$

To see this, note that Lemma 6 provides the optimality of the cluster index Γ_i^* for signal Y_i for each i, for given W^*. Now, for a given (optimal) cluster index Γ_i^*, the set of optimal sparse codes for signal Y_i is given by $\tilde{H}_s\left(W_{\Gamma_i^*}^* Y_i\right)$. Since, by Lemma 4, $X_i^* \in \tilde{H}_s\left(W_{\Gamma_i^*}^* Y_i\right)$, we obtain (36).

Next, we have the result of Lemma 7 that

$$W^* \in \underset{W}{\arg\min}\ g\left(W, X^*, \Gamma^*\right) \tag{37}$$

In order to deal with any non-uniqueness of solutions in (37), we assume for Algorithm A1 that if a certain iterate W^{t+1} (fixed t) satisfies $g\left(W^{t+1}, X^t, \Gamma^t\right) = g\left(W^t, X^t, \Gamma^t\right)$, then we equivalently set $W^{t+1} = W^t$. Similarly, in order to deal with any non-uniqueness of solutions in (36), we assume that if $W^{t+1} = W^t$ holds (fixed t) and the iterate $\left(X^{t+1}, \Gamma^{t+1}\right)$ satisfies $g\left(W^{t+1}, X^{t+1}, \Gamma^{t+1}\right) = g\left(W^t, X^t, \Gamma^t\right)$, then we equivalently set $\Gamma^{t+1} = \Gamma^t$ and $X^{t+1} = X^t$[19]. These assumptions and equations (37) and (36) imply that if we provide the accumulation point (W^*, X^*, Γ^*) into Algorithm A1 as the initial iterate, the algorithm stays at the point. Therefore, the accumulation point is a fixed point. □

The fixed point property implies that every accumulation point (W^*, X^*, Γ^*) is a global optimum of $g\,(W, X, \Gamma)$ with respect to either $\{W_k\}$, or (X, Γ), with the other variables fixed. The following lemma establishes the local optimality (jointly with respect to the transform and sparse code variables) of the accumulation points.

Lemma 9 *Every fixed point* (W, X, Γ) *of Algorithm A1 is a local optimum of the objective* $g\,(W, X, \Gamma)$ *with respect to* (W, X).

Proof Since (W, X, Γ) is a fixed point of Algorithm A1, we have that

$$W \in \underset{\tilde{W}}{\arg\min}\ \ g\left(\tilde{W}, X, \Gamma\right) \tag{38}$$

The above optimization problem (over the cluster transforms) involves an unconstrained objective, that is separable into the component-wise objectives corresponding to each $\tilde{W}_k$, i.e., we can write

$$g\left(\tilde{W}, X, \Gamma\right) = \sum_{k=1}^{K} g_k\left(\tilde{W}_k, X_{C_k}\right), \tag{39}$$

[19] This rule is trivially satisfied due to the way Algorithm A1 is written, except perhaps for the case when the superscript $t = 0$. In the latter case, if the rule is applicable, it means that the algorithm has already reached a fixed point (the initial $\left(W^0, X^0, \Gamma^0\right)$ is a fixed point), and therefore, no more iterations are performed. All aforementioned convergence results hold true for this degenerate case.

where we denoted the set of indices i for which $\Gamma_i = k$ by C_k. Specifically, the component-wise objective is given as

$$g_k\left(\tilde{W}_k, X_{C_k}\right) = \left\|\tilde{W}_k Y_{C_k} - X_{C_k}\right\|_F^2 + \lambda_k \left\|\tilde{W}_k\right\|_F^2 - \lambda_k \log\left|\det \tilde{W}_k\right| + \phi(X_{C_k}), \tag{40}$$

where $\lambda_k = \lambda_0 \left\|Y_{C_k}\right\|_F^2$.

Each W_k in (38) is a global minimizer of the corresponding component-wise objective. Therefore, it provides a gradient value of 0 (necessary condition) for that objective. Thus, we have that

$$2W_k Y_{C_k} Y_{C_k}^T - 2X_{C_k} Y_{C_k}^T + 2\lambda_k W_k - \lambda_k W_k^{-T} = 0 \tag{41}$$

Since (W, X, Γ) is a fixed point of Algorithm A1, we also have that

$$(X, \Gamma) \in \underset{\tilde{X}, \tilde{\Gamma}}{\arg\min}\ g\left(W, \tilde{X}, \tilde{\Gamma}\right) \tag{42}$$

Therefore, we have for any k that

$$X_i \in \tilde{H}_s(W_k Y_i)\ \forall\ i \in C_k \tag{43}$$

Now, keeping the cluster indices Γ fixed, and using the definition of g_k in (40) along with (41) and (43), and applying Lemma 12 in Appendix , we get that the following condition holds for each component-wise objective g_k.

$$g_k(W_k + dW_k, X_{C_k} + \Delta X_{C_k}) \geq g_k\left(W_k, X_{C_k}\right) \tag{44}$$

The condition holds for all sufficiently small $dW_k \in \mathbb{R}^{n\times n}$ satisfying $\|dW_k\|_F \leq \epsilon_k$ for some $\epsilon_k > 0$ that depends on the specific W_k, and $\Delta X_{C_k} \in \mathbb{R}^{n\times|C_k|}$ in the union of the following regions.

R1$_k$. The half-space $tr\left\{(W_k Y_{C_k} - X_{C_k})\Delta X_{C_k}^T\right\} \leq 0$.
R2$_k$. The *local region* defined by
$\left\|\Delta X_{C_k}\right\|_\infty < \min_{i\in C_k}\{\beta_s(W_k Y_i) : \|W_k Y_i\|_0 > s\}$.

If we have $\|W_k Y_i\|_0 \leq s\,\forall\ i \in C_k$, then ΔX_{C_k} can be arbitrary.

Finally, summing the result in (44) over all k, we get that

$$g\,(W + dW, X + \Delta X, \Gamma) \geq g\,(W, X, \Gamma) \tag{45}$$

The above condition (45) holds for $\|dW_k\|_F \leq \epsilon_k$ and ΔX_{C_k} in R1$_k\cup$ R2$_k$ for all k. Thus, the fixed point (W, X, Γ) is a local minimum of the objective $g\,(W, X, \Gamma)$, with respect to the cluster transform and sparse code variables. □

This concludes the proof of Theorem 1.

4.2.2 Proof of Theorem 2

Let (W, X, Γ) be an accumulation point of the iterate sequence in Algorithm A1. The function $f(W)$ involves the summation of N terms of the form

$$f_i(W) = \min_k \left\{\|W_k Y_i - H_s(W_k Y_i)\|_2^2 + \lambda_0\, Q(W_k)\, \|Y_i\|_2^2\right\}$$

The function $f_i(W)$ computes the minimum value of the clustering measure for Y_i with respect to the set of transforms $\{W_k\}$. Recall that W is obtained the stacking the W_k's on top of each other. Let us denote the clustering measure corresponding to signal Y_i in a fixed transform $B \in \mathbb{R}^{n\times n}$ by the function $\tilde{f}_i(B) = \|BY_i - H_s(BY_i)\|_2^2 + \lambda_0\, Q(B)\, \|Y_i\|_2^2$. Then,

$$f_i(W) = \min_k\quad \tilde{f}_i(W_k) \tag{46}$$

Lemma 3 established the boundedness of the iterate sequence in Algorithm A1. This implies that the accumulation (W, X, Γ) is also bounded (same bound works as for the iterates). By Lemma 8, the accumulation point is a fixed point. Since $Y \in \mathbb{R}^{n\times N}$, the training signals are all bounded. We can now use Lemma 13 of Appendix (which shows Lipschitz continuity, and therefore continuity of the sparsification error function), and the fact that $Q(B)$ is continuous at full rank matrices B, to conclude that the function $\tilde{f}_i(B)$ is continuous at the point W_k for any k.

Now, the optimal cluster index for a signal Y_i with respect to the set of transforms (fixed point) $\{W_k\}$ is (assumed) unique, i.e., there is a non-zero separation between the smallest value of the clustering measure $\tilde{f}_i(W_k)$ (minimum over k) and the second smallest value. For signal Y_i, by the fixed point Eq. (42), the optimal cluster index with respect to the fixed point $\{W_k\}$ is $k = \Gamma_i$. If we perturb $\{W_k\}$ by sufficiently small $\{dW_k\}$, the minimum cluster index for Y_i with respect to $\{W_k + dW_k\}$ remains at (the unique) Γ_i due to the continuity of the function $\tilde{f}_i(B)$ at $B = W_k\ \forall\ k$ in (46), and because $\arg\min_k\ \tilde{f}_i(W_k)$ is unique ($= \Gamma_i$).

Therefore, for a particular Y_i, we have that $f_i(W) = \tilde{f}_i(W_{\Gamma_i})$ and $f_i(W + dW) = \tilde{f}_i(W_{\Gamma_i} + dW_{\Gamma_i})$ for all sufficiently small $\{dW_k\}$ (dW is formed by stacking the dW_k on top of each other). Further, since X is the (assumed) unique minimizer of $g\left(W, \tilde{X}, \Gamma\right)$ for fixed W and Γ, we also have that $\tilde{H}_s(W_{\Gamma_i} Y_i)$ is the singleton $H_s(W_{\Gamma_i} Y_i)$. It is then easy to obtain (using the definition of the derivative as a limit) the following two equations

$$\nabla_{W_{\Gamma_i}} f_i(W) = 2W_{\Gamma_i} Y_i Y_i^T - 2H_s(W_{\Gamma_i} Y_i) Y_i^T + \lambda_0 \|Y_i\|_2^2 \left(2W_{\Gamma_i} - W_{\Gamma_i}^{-T}\right) \tag{47}$$

$$\nabla_{W_k} f_i(W) = 0\ \forall\ k \neq \Gamma_i. \tag{48}$$

Therefore, for a particular index i, the derivative of $f_i(W)$ at the fixed point W exists and is given by the above expressions.

We can now show that the fixed point (W, X, Γ) of Algortihm A1 is a stationary point of $f(W)$. First, by the assumption that X is the unique minimizer of $g\left(W, \tilde{X}, \Gamma\right)$ for fixed W and Γ, we have that $H_s(W_k Y_i) = \tilde{H}_s(W_k Y_i)\ \forall\ i \in C_k$ in (43), or that $X_i = H_s(W_k Y_i)\ \forall\ i \in C_k$. Then, using (47) and (48), we have that for any k for which C_k

is non-empty, the derivative $\nabla_{W_k} f(W)$ at the fixed point (W, X, Γ) is $2W_k Y_{C_k} Y_{C_k}^T - 2X_{C_k} Y_{C_k}^T + 2\lambda_k W_k - \lambda_k W_k^{-T}$, where $\lambda_k = \lambda_0 \left\| Y_{C_k} \right\|_F^2$. When C_k is empty, the aforementioned derivative is 0.

By the fixed point Eq. (41), we then have that $\nabla_{W_k} f(W) = 0$ at each k, i.e., the fixed point is a stationary point of f. This concludes the proof of Theorem 2. □

4.2.3 Motivation for Conjecture 1

Conjecture 1 says that every accumulation point (W, X, Γ) of the iterate sequence generated by Algorithm A1 is such that (X, Γ) is the *unique* minimizer of $g\left(W, \tilde{X}, \tilde{\Gamma}\right)$ for fixed W. By Lemma 8, we know that every accumulation point (W, X, Γ) is a fixed point of Algorithm A1.

Firstly, Conjecture 1 says that the clustering indices Γ computed with respect to the set $\{W_k\}$ are uniquely optimal, i.e., for each Y_i, there is a non-zero separation between the smallest value of the clustering measure $\tilde{f}_i(W_k)$ (minimum over k) and the second smallest value.[20] From the fixed point equations (41) and (43), we can see that each transform W_k is essentially a (non-linear) function of the signals in cluster k. Since the clusters are disjoint, and the signals in each cluster are independent and continuously distributed (by Assumption 2), we conjecture that the event that any two transforms W_k and W_j (for $k \neq j$) achieve the exact same (minimum) value of the clustering measure for a signal Y_i has probability 0.

Secondly, Conjecture 1 says that the set $\tilde{H}_s(W_{\Gamma_i} Y_i)$ of optimal sparse codes for signal Y_i is a singleton $\forall\, i$. In order for this to fail, the vector $W_{\Gamma_i} Y_i$ must have two entries of identical magnitude. However, because the full rank W_{Γ_i} is a function of the training signals in class k, and since the training signals are continuously distributed with an absolutely continuous probability measure (by Assumption 2), we conjecture that the event that two entries of $W_{\Gamma_i} Y_i$ have identical magnitude has probability 0.

5 Image Denoising

There are numerous applications that benefit from a good sparse model. Image denoising is an important and classical application that has been widely studied. 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 $h \in \mathbb{R}^P$ is a noise vector. Here, we consider h whose entries are i.i.d. Gaussian with zero mean and variance σ^2. We propose an adaptive image denoising framework in this section that exploits the proposed union of transforms model, or OCTOBOS model.

5.1 Problem Formulation

Similar to previous work on dictionary-based image denoising (Elad and Aharon 2006), we work with image patches. We model them as sparse in a transform domain. We allow overlapping patches, which provide an additional averaging effect that reduces noise. The patches considered can be vectorized to form the columns of a training matrix, allowing us to utilize the proposed schemes such as (P5) to learn an adaptive transform for patches.

Similar to the previous formulation (Ravishankar and Bresler 2013b) for adaptive square transform-based denoising, we propose the following image denoising problem formulation that exploits the union of transforms model.

$$\min_{\{W_k, x_i, \alpha_i, C_k\}} \sum_{k=1}^{K} \sum_{i \in C_k} \left\{ \| W_k x_i - \alpha_i \|_2^2 + \lambda_i' \, Q(W_k) \right\}$$
$$+ \tau \sum_{i=1}^{N} \| R_i \, y - x_i \|_2^2$$
$$s.t. \;\; \|\alpha_i\|_0 \leq s_i \;\; \forall \;\; i, \quad \{C_k\} \in G \qquad \text{(P8)}$$

Here, $R_i \in \mathbb{R}^{n \times P}$ is defined to be a patch extraction operator, i.e., $R_i y \in \mathbb{R}^n$ denotes the ith patch of the image y as a vector. We assume a total of N overlapping patches. Compared with Problem (P5), the denoising problem includes the additional, yet important data fidelity term $\tau \sum_{i=1}^N \| R_i \, y - x_i \|_2^2$. The assumption in (P8) is that there exist noiseless $x_i \in \mathbb{R}^n$ that approximate $R_i y$, and are approximately sparsifiable by the learned model. The weight τ for the fidelity term is typically inversely proportional to the noise level σ, that is assumed known apriori. Vector $\alpha_i \in \mathbb{R}^n$ in (P8) denotes the sparse representation of x_i in a specific cluster transform W_k, with an apriori unknown sparsity level s_i. The weight λ_i' is set based on the given noisy data $R_i y$ as $\lambda_0 \| R_i y \|_2^2$. The net weight on the $Q(W_k)$ regularizer in (P8) is then $\lambda_k = \sum_{i \in C_k} \lambda_i'$. Thus, similar to Problem (P5), the weight λ_k here varies depending on C_k.

Since $\tau \propto 1/\sigma$, we have the result that when $\sigma \rightarrow 0$, the optimal $x_i \rightarrow R_i y$ in (P8). In the limit, (P8) reduces to the transform learning problem (P5). Since the patch-based framework is used in formulation (P8), the denoised image x is obtained by averaging the learned x_i's at their respective locations in the image (Ravishankar and Bresler 2013b).

Problem (P8) has the disadvantage that it involves apriori unknown sparsity levels s_i. These sparsity levels have to be estimated in practice. An alternative version of (P8) would replace the penalty $\tau \sum_{i=1}^N \| R_i \, y - x_i \|_2^2$ by constraints $\| R_i \, y - x_i \|_2^2 \leq n C^2 \sigma^2 \; \forall \, i$, where C is a constant.

[20] The uniqueness of the cluster index for each signal Y_i in the iterations of Algorithm A1 for various data sets was empirically observed.

Furthermore, the sparsity constraints in (P8) can be converted to a penalty of the form $\sum_{i=1}^{N} \gamma_i \, \|\alpha_i\|_0$. Although this modification eliminates the issue of unknown sparsity levels s_i, it introduces another new set of unknown parameters $\gamma_i > 0$.[21] In this paper, we will work with the formulation (P8) that uses the (simple) data fidelity penalty and sparsity constraints. Our algorithm for (P8) will additionally estimate the (minimum) sparsity levels for which the condition $\|R_i \, y - x_i\|_2^2 \leq nC^2\sigma^2 \; \forall \, i$ is satisfied (similar to (Elad and Aharon 2006)).

5.2 Algorithm for (P8)

The proposed iterative algorithm is aimed at solving the nonconvex Problem (P8). While one could solve (P8) with fixed s_i (e.g., s_i set to 10 % of the patch size), we observed that the denoising performance is better when the s_i's are tuned adaptively as discussed above. Each iteration of our algorithm involves intra-cluster transform learning, variable sparsity level update, and clustering steps. The denoised patches x_i are updated in the final iteration. The denoised image is reconstructed when the iterations complete.

5.2.1 Intra-cluster Transform Learning

Given $\{x_i\}$, $\{s_i\}$, and the clusters C_k, we solve for the cluster transforms $\{W_k\}$ and the corresponding sparse codes $\{\alpha_i\}$ in (P8). This problem separates out into K different single transform learning problems (similar to (P3)). The kth problem is as follows.

$$\min_{\{W_k, \alpha_i\}} \sum_{i \in C_k} \left\{ \|W_k x_i - \alpha_i\|_2^2 + \lambda_i' \quad Q(W_k) \right\}$$
$$s.t. \;\; \|\alpha_i\|_0 \leq s_i \;\; \forall \;\; i \in C_k \tag{49}$$

This problem is solved by alternating between sparse coding and transform update steps. For each of these steps, we use closed-form solutions (Ravishankar and Bresler 2013a).

5.2.2 Intra-cluster Sparsity Level Update

Now, we update the sparsity levels s_i for all i. We adopt a similar method for updating the sparsity levels as introduced by Ravishankar and Bresler (2013b).

With a fixed cluster transform W_k and α_i $(i \in C_k)$, one can solve for x_i in (P8) in the least squares sense as follows.

$$x_i = \begin{bmatrix} \sqrt{\tau} \, I \\ W_k \end{bmatrix}^{\dagger} \begin{bmatrix} \sqrt{\tau} R_i \, y \\ \alpha_i \end{bmatrix} = G_1 R_i \, y + G_2 \alpha_i, \tag{50}$$

where I is the $n \times n$ identity, and the matrices G_1 and G_2 are given as $G_1 = \tau \left(\tau I + W_k^T W_k\right)^{-1}$ and $G_2 = \left(\tau I + W_k^T W_k\right)^{-1} W_k^T$. Both G_1 and G_2 are computed once for each cluster.

With α_i held at $H_{s_i}(W_k R_i \, y)$, we choose s_i to be the smallest integer such that the x_i in (50) satisfies the error condition $\|R_i \, y - x_i\|_2^2 \leq nC^2\sigma^2$. This can be done efficiently by precomputing $m_i = G_1 R_i \, y$ and adding to it one scaled column of G_2 at a time (corresponding to incrementing s_i by 1 in $\alpha_i = H_{s_i}(W_k R_i \, y)$), until the error condition is met.

We only update the s_i's in this step, except in the final algorithm iteration, when the x_i's computed above satisfying the $\|R_i \, y - x_i\|_2^2 \leq nC^2\sigma^2$ condition represent the final denoised patches. Note that the sparse code is also further updated here for each $i \in C_k$ as $\alpha_i = H_{s_i}(W_k R_i \, y)$, using the optimal s_i.

5.2.3 Clustering

With fixed $\{s_i\}$, $\{W_k\}$, and $\{x_i\}$, we solve (P8) with respect to the clusters $\{C_k\}$ and sparse codes $\{\alpha_i\}$. This problem is similar to (P6). For each i, we solve a sparse coding and clustering problem in the union of transforms model. We calculate $\tilde{\alpha}_i^k = H_{s_i}(W_k R_i \, y) \, \forall \, k$, and choose the cluster $C_{\hat{k}_i}$ if we have that $\left\| W_{\hat{k}_i} x_i - \tilde{\alpha}_i^{\hat{k}_i} \right\|_2^2 + \eta_{\hat{k}_i} \, \|x_i\|_2^2 \leq \left\| W_j x_i - \tilde{\alpha}_i^j \right\|_2^2 + \eta_j \, \|x_i\|_2^2 \;\; \forall \, j \neq \hat{k}_i$, where $\eta_j = \lambda_0 \, Q(W_j)$. Then, the optimal $\alpha_i = \tilde{\alpha}_i^{\hat{k}_i}$. Note that the clustering step is not performed in the final iteration of our algorithm for (P8).

5.2.4 Computing Denoised Image Estimate

The denoised image patches $\{x_i\}$ obtained from the iterative scheme for (P8) are restricted to their range (e.g., 0–255 for unsigned 8-bit integer class). We output the denoised image by averaging the denoised patches at their respective image locations. The summary of the method for (P8) is presented in Fig. 2.

In order to enhance the method's efficiency, we typically perform the intra-cluster transform learning in our algorithm using only a subset of patches that are selected uniformly at random in each cluster. The patches are all mean-subtracted in our algorithm, and the means are added back to the final denoised patch estimates.

Our algorithm learns a union of transforms using noisy patches, and updates the sparsity levels s_i adaptively during the iterations. One could use the final s_i's output from the algorithm and re-solve (P8) with fixed s_i's by alternating between the intra-cluster transform learning, x_i update (by least squares), and clustering steps. However, we observed in our experiments that such additional iterations produce

[21] The γ_i's need to be set accurately for the modified formulation to work well in practice.

```
Algorithm for (P8)
Input : $y$ - noisy image, $s$ - initial fixed sparsity, $K$ - number of clusters, $L$ - number of iterations of algorithm for (P8), $\sigma^2$ - an estimate of noise variance, $J$ - number of transform learning iterations, $\lambda_0$ - a constant.
Output : $x$ - Denoised image estimate
Initialization : Patches $x_i = R_i y$ and $s_i = s$ for $i = 1, 2, ...., N$. Initial $W_k = W_0 \, \forall \, k$, $C_k$ - random cluster selection for each $i \in \{1, ..., N\}$.
For l = 1:L Repeat
  1. For $k = 1...K$, update $W_k$ and the corresponding $\alpha_i$ alternatingly (to solve problem (49)), with fixed clusters $C_k$ and $x_i = R_i y$. The number of alternations for each $k$ is $J$.
  2. Update $s_i$ for all $i = 1, 2, ...., N$ : Increase $s_i$ in $\alpha_i = H_{s_i}(W_k R_i y)$ in (50) where $i \in C_k$, until the error condition $\|R_i \, y - x_i\|_2^2 \leq nC^2\sigma^2$ is reached.
  3. For each $i \in \{1, ..., N\}$, perform clustering and sparse coding with $x_i = R_i y$: calculate $\tilde{\alpha}_i^k = H_{s_i}(W_k x_i) \, \forall \, k$ and assign $i$ to cluster $C_{\hat{k}}$ if $\hat{k}$ is the smallest integer in $\{1, ..., K\}$ such that $\left\|W_{\hat{k}} x_i - \tilde{\alpha}_i^{\hat{k}}\right\|_2^2 + \eta_{\hat{k}} \|x_i\|_2^2 \leq \left\|W_j x_i - \tilde{\alpha}_i^j\right\|_2^2 + \eta_j \|x_i\|_2^2 \;\; \forall j \neq \hat{k}$ holds with $\eta_j = \lambda_0 \, Q(W_j) \, \forall \, j$. The optimal code $\alpha_i = \tilde{\alpha}_i^{\hat{k}}$.
End
Update $x$ : Obtain the denoised patches $x_i$ satisfying the error condition in step 2 above, and average them at their respective image locations.
```

Fig. 2 Algorithm to solve (P8), and obtain a denoised image estimate x. A particular initialization is mentioned above for $\{W_k, C_k, s_i\}$, for simplicity. The W_0 above can be chosen to be the DCT, Wavelets, etc

at most a minor additional improvement in denoising performance. Hence, to save run time, we do not include them.

Note that similar to previous work (Ravishankar and Bresler 2013b), we do not enforce the constraint $R_i x = x_i$ explicitly in (P8), but rather treat extracted patches as individual data points. Although the final denoised image estimate is computed by averaging all patches (at respective locations), the approach may be sub-optimal (Zoran and Weiss 2011), but results in an effective algorithm with low computational cost. Numerical results presented in Sect. 6 demonstrate this.

5.3 Improved Denoising by Iteratively Resolving (P8)

Our aforementioned algorithm obtains a denoised image estimate by solving (P8) once. We propose an improved iterative denoising scheme that makes multiple passes through (P8), each time replacing y by its latest denoised version, setting the noise level to an estimate of the remaining noise in the denoised image produced in the previous pass. Each iteration of this denoising scheme uses the same algorithm (for (P8)) as in Sect. 5.2.

6 Numerical Experiments

6.1 Framework

6.1.1 Overview

In this section, we present results demonstrating the promise of the proposed adaptive union of transforms, or OCTOBOS framework in image representation, and in image denoising. First, we illustrate the convergence behavior of our alternating transform learning algorithm. We consider the behavior of our algorithm with various initializations to empirically check whether the algorithm is sensitive to initializations. We will also provide examples showing the clustering/classification ability of our approach. Then, we illustrate the promise of the proposed transform learning approach for representing various images. We compare the image representation quality provided by our learnt OCTOBOS transforms to that provided by learnt single square transforms, and by fixed analytical transforms such as the DCT[22]. Finally, we demonstrate the potential of the proposed adaptive OCTOBOS transform-based image denoising scheme. We will show that the proposed approach performs better than methods involving learnt single square transforms or learnt overcomplete synthesis dictionaries (K-SVD[23]). The computational advantage of the proposed approach over the synthesis dictionary-based approach will also be indicated. Furthermore, we demonstrate that our method denoises better than GMM denoising (Zoran and Weiss 2011), and is competitive with the state-of-the-art BM3D (Dabov et al. 2007) for some images and/or noise levels.

The data in our experiments are generated as the patches of natural images. We employ our proposed transform learning Problem formulation (P5) for learning adaptive sparse representations of such patches. The means (DC values) of the patches are removed and we only sparsify the mean-subtracted patches which, after reshaping as vectors, are stacked as columns of the training matrix Y. The means are added back for image display. Mean removal is typically adopted in image processing applications such as com-

[22] The DCT is a popular analytical transform that has been extensively used in compression standards such as JPEG.

[23] The K-SVD method is a highly popular scheme that has been applied to a wide variety of image processing applications (Elad and Aharon 2006; Mairal et al. 2008a). Mairal et al. (2009) have proposed a non-local method for denoising, that also exploits learned dictionaries. A similar extension of OCTOBOS learning-based denoising using non-local means methodology may potentially provide enhanced performance for OCTOBOS. However, such an extension would distract from the focus on the OCTOBOS model in this work. Hence, we leave its investigation for future work. For the sake of simplicity, we compare our overcomplete transform learning scheme to the corresponding overcomplete synthesis dictionary learning scheme K-SVD in this work.

pression and image denoising (Elad 2009; Ravishankar and Bresler 2013b).

All our (unoptimized) implementations were coded in Matlab version R2013b. The corresponding Matlab implementation of K-SVD denoising (Elad and Aharon 2006) available from Elad's website (Elad 2009) was used in our comparisons. For K-SVD denoising, we used the built-in parameter settings of the author's implementation. We also used the publicly available implementations of GMM (Weiss 2011) and BM3D denoising (Dabov et al. 2011). All computations were performed with an Intel Core i7 CPU at 2.9GHz and 4GB memory, employing a 64-bit Windows 7 operating system.

6.1.2 Quality/Performance Metrics:

Several metrics have been introduced previously to evaluate the quality of learnt transforms(Ravishankar and Bresler 2013c, b). The *normalized sparsification error* (NSE) for a single 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. The NSE measures the fraction of energy lost in sparse fitting in the transform domain, and is an interesting property to observe for the adaptive transforms. For our proposed approach, since we have a union of square transforms and clustered patches, we compute the normalized sparsification error as follows.

$$\text{NSE} = \frac{\sum_{k=1}^{K} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2}{\sum_{k=1}^{K} \sum_{i \in C_k} \|W_k Y_i\|_2^2} \tag{51}$$

Here, the numerator is the net sparsification error (i.e., the total transform domain residual), and the denominator denotes the total transform domain energy. We have $0 \leq \text{NSE} \leq 1$. The sparse codes in the kth cluster above are $X_i = H_s(W_k Y_i)$. For the proposed NSE definition to be meaningful, we assume that the W_k's are all normalized (e.g., they have unit spectral norm). When $K = 1$, the above definition is identical to the previously proposed NSE (Ravishankar and Bresler 2013c) for a single transform.

For image representation, a useful performance metric is the recovery peak signal to noise ratio (or *recovery PSNR* (RP)), which for the case of a single transform W was previously defined as $20 \log_{10} \left(255\sqrt{P} / \left\|Y - W^{-1}X\right\|_F\right)$ in decibels (dB), where P is the number of image pixels and X is again the transform sparse code of data Y. The recovery PSNR measures the error in recovering the patches Y (or equivalently the image, in the case of non-overlapping patches) as $W^{-1}X$ from their sparse codes X obtained by projecting WY onto the ℓ_0 ball. The recovery PSNR serves as a simple surrogate for the performance of a learnt transform in a compression application. For our proposed union of transforms approach, the recovery PSNR is redefined in terms of the clusters as follows.

$$\text{RP} = 20 \log_{10} \left(\frac{255\sqrt{P}}{\sqrt{\sum_{k=1}^{K} \sum_{i \in C_k} \left\|Y_i - W_k^{-1} X_i\right\|_2^2}} \right) \tag{52}$$

Note that each patch Y_i belongs to exactly one cluster C_k above.

For image denoising, similar to previous work (Elad and Aharon 2006), we measure the image reconstruction PSNR computed between the true noiseless reference and the noisy or denoised images.

6.2 Convergence and Learning Behavior

Here, we illustrate the convergence behavior of our alternating OCTOBOS learning algorithm for image data. We extract the $\sqrt{n} \times \sqrt{n} = 8 \times 8$ $(n = 64)$ non-overlapping mean-subtracted patches from the 512×512 Barbara image (shown later in Fig. 8). The data matrix $Y \in \mathbb{R}^{n \times N}$ in this case has 4096 vectorized training patches. We learn a union of transforms (or, equivalently an OCTOBOS transform) for this data by solving (P5). The parameters are set as $\lambda_0 = 3.1 \times 10^{-3}$, $s = 11$ (which is roughly 1/6th of the data dimension), and the number of clusters $K = 2$. The choice of λ_0 here ensures well-conditioning of the blocks of the learnt overcomplete transform. Badly conditioned transforms degrade performance in applications (Ravishankar and Bresler 2013c). Hence, we focus our investigations here only on the well-conditioned scenario.

In the experiments of this paper, we assume an initialization (or, initial estimates) for the clusters $\{C_k\}$ and cluster transforms $\{W_k\}$ in (P5). The initial sparse codes in (P5) are then computed for the initialization as $X_i = H_s(W_k Y_i)$, $\forall i \in C_k$, and for each k, and the alternating algorithm A1 is executed beginning with the transform update step. Note that the initial $\{X_i\}$ are fully determined by the initial estimates for the cluster-specific transforms.

Here, we study the convergence behavior of the algorithm for various initializations. We consider two different scenarios. In Scenario A, we fix the initial clusters $\{C_k\}$ (each patch is assigned uniformly at random to one of $K = 2$ clusters), and vary the initialization for the cluster transforms $\{W_k\}$. Four different initializations for the $\{W_k\}$ are considered: (i) the 64×64 2D DCT matrix (obtained as the Kronecker product of two 8×8 1D DCT matrices); (ii) the Karhunen-Loève Transform (KLT) initialization, obtained by inverting (transposing) the left singular matrices of the data in each cluster; (iii) the identity matrix; and (iv) a random matrix with i.i.d. gaussian entries (zero mean and standard deviation 0.2), respectively.[24] In Scenario B, we fix the initial

[24] Note that for the case of the DCT, identity, and random initializations, the same matrix is used to initialize all the W_k's.

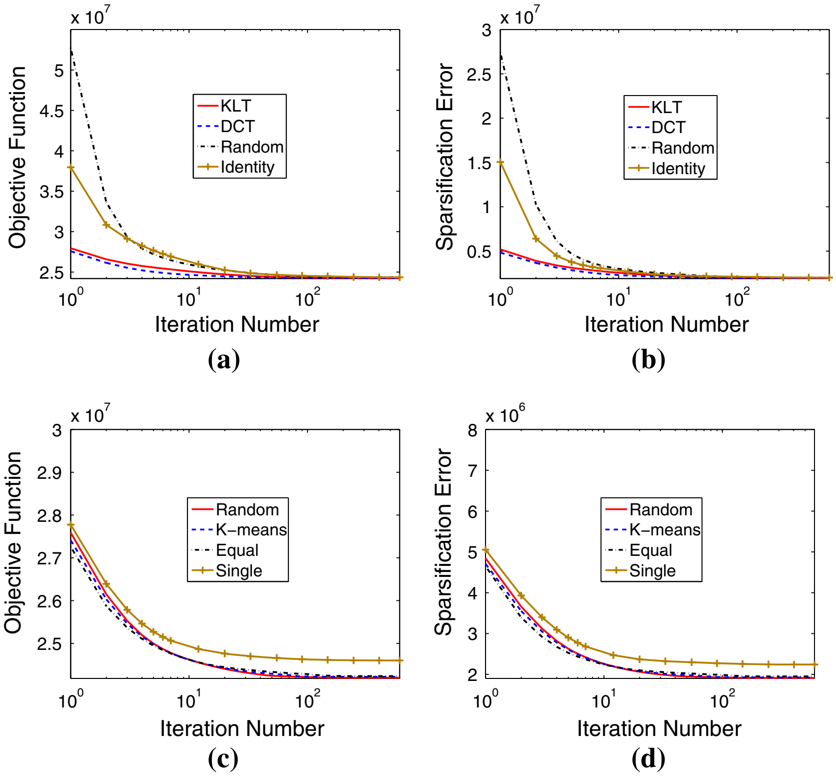

Fig. 3 Behavior of the OCTOBOS learning algorithm for (P5): **a** Objective function with different transform initializations; **b** Sparsification error with different transform initializations; **c** Objective function with different cluster initializations for $K = 2$, along with the objective for single square transform ($K = 1$) learning; **d** Sparsification error with different cluster initializations for $K = 2$, along with the sparsification error for single square transform ($K = 1$) learning

cluster transforms $\{W_k\}$ to be the 2D DCT, and vary the initialization for the $\{C_k\}$ in (P5). We consider three initializations for the clusters here: (i) the initialization obtained by using the well-known k-means algorithm; (ii) random clustering, where each patch is assigned uniformly at random (a different random clustering is used here, than the one in the aforementioned Scenario A) to one of the clusters; and (iii) the patches on the left half of the image in one cluster, and the remaining patches in the second cluster. We will refer to the initialization (iii) as 'equal' initialization, for simplicity.

Figure 3 shows the progress of the algorithm over iterations for the various initializations of $\{W_k\}$ (Scenario A in Fig. 3a, b), and $\{C_k\}$ (Scenario B in Fig. 3c, d). Both the objective function (Fig. 3a, c) and sparsification error (Fig. 3b, d) converge quickly for our algorithm. Importantly, the final values of the objective (similarly, the sparsification error) are nearly identical for all the initializations. This indicates that our learning algorithm is reasonably robust, or insensitive to initialization. Good initializations for the $\{W_k\}$ such as the DCT and KLT lead to faster convergence of learning (Fig. 3a, b).

For comparison, we also plot in Fig. 3c, d, the behavior of the algorithm for $K = 1$. In this case it reduces to the single square transform learning algorithm via (P3) (Ravishankar and Bresler 2013c, a). The parameters such as s and λ_0 are set to the same values as for $K = 2$. Figure 3c shows the objective for single square transform learning converging to a larger value compared to OCTOBOS learning. Likewise, the sparsification error for OCTOBOS for $K = 2$ (Fig. 3d) is 0.67 dB better than that provided by the learnt single square transform. This confirms our expectation based on Proposition 4 in Sect. 2.

The learned square blocks of the overcomplete transform here have similar condition numbers (≈1.4) and Frobenius norms (≈5) for all initializations. This confirms that an appropriate choice of λ_0 allows the learnt W_k's to be similarly scaled, and ensures that the sparsification error term in (P5) is fully meaningful.

Next, we plot in Fig. 4, the cluster sizes over iterations for two different initializations of $\{C_k\}$—random (Fig. 4a) and 'equal' (Fig. 4b). The final values of $|C_1|$ (alternatively

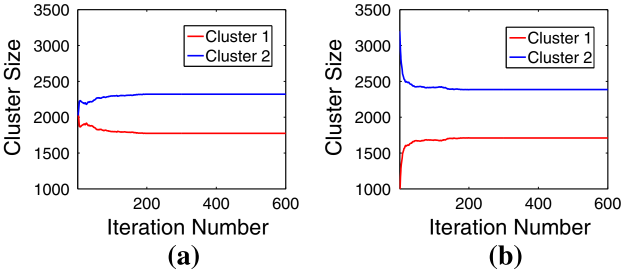

Fig. 4 Cluster size convergence for (P5) corresponding to Fig. 3c: **a** Case of random cluster initialization; **b** Case of 'equal' cluster initialization. Note that the cluster sizes change dramatically in the first iteration in **b**

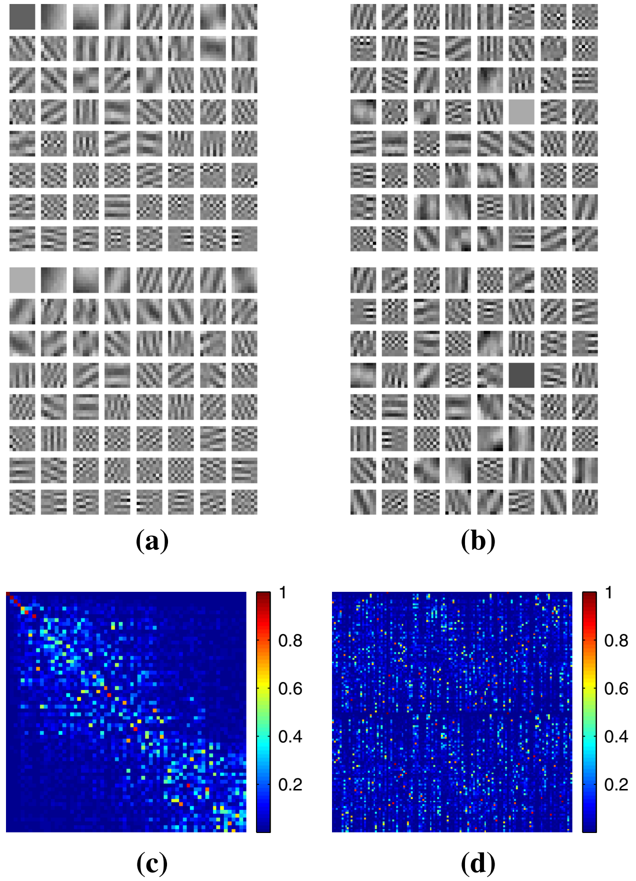

Fig. 5 Learned OCTOBOS transforms corresponding to Fig. 3a: rows of the learned overcomplete transform W shown as patches (the two square blocks of W are separated by a *white space*) for the case of **a** KLT initialization, and **b** random matrix initialization; Magnitude of the cross-gram matrix computed: **c** between the two learnt (row-normalized) square blocks in **a**; and **d** between the two (row-normalized) overcomplete transforms in **a** and **b**

$|C_2|$) for the two initializations are similar. We observed that the (learnt) clusters themselves can be similar (although, not necessarily identical) for various initializations.

Figure 5 visualizes the transforms learnt by our alternating algorithm with the KLT and with random initializations for $\{W_k\}$ (the aforementioned Scenario A). The rows, or atoms of the learnt overcomplete transforms are shown as patches. The learnt transforms exhibit geometric and texture-like features, achieved by adaptation to the patches of the Barbara image. In order to gauge the similarity, or difference between the learnt OCTOBOS transforms with different initializations, we show in Fig. 5d, the magnitude of the cross-gram matrix[25] computed between the transforms in Fig. 5a, b. We normalize the rows of the transforms prior to computing their cross-gram matrix. The cross-gram matrix then indicates the coherence between every pair of rows from the two overcomplete transforms. For the 128 × 128 cross-gram matrix in this case, there are only 15 entries with amplitude above 0.9. This indicates that the two learnt OCTOBOS transforms are not similar, i.e., they are not related by just row permutations and sign changes. However, interestingly, both still sparsify the data Y equally well. Therefore, as far as sparsification is concerned, the two different overcomplete transforms can be considered essentially equivalent (Ravishankar and Bresler 2013c).

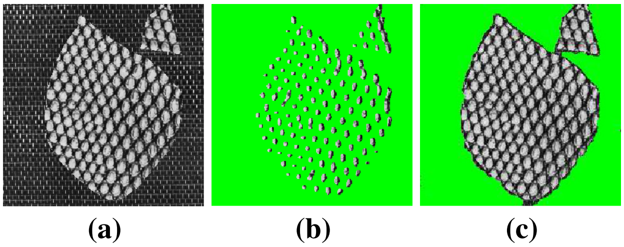

Fig. 6 $K = 2$ clustering example: **a** Input image; Input image with pixels clustered into Class 1 shown in green for **b** the K-means initialization, and **c** OCTOBOS (Color figure online)

How similar are the square blocks of the same overcomplete transform? In Fig. 5c, we show the magnitude of the 64 × 64 cross-gram matrix computed between the (row normalized) blocks in Fig. 5a. In this case, there are only 5 entries with amplitude above 0.9, indicating that the two learnt square blocks are quite different. This is not surprising, since the two blocks here correspond to disjoint clusters.

Although we considered the image Barbara in our convergence study here, we observed similar behavior for our algorithm for other images as well.

6.3 Clustering Behavior

In this subsection, we briefly illustrate the clustering behavior of our OCTOBOS learning scheme. First, we consider the 251 × 249 input image shown in Fig. 6a. The image was formed by combining two textures from the Brodatz database (Brodatz 1966; He and Safia 2013). The goal is to cluster the pixels of the image into one of two classes. In order to do so, we adopt the following strategy. We consider all overlapping mean-subtracted patches from the input image, and employ formulaton (P5) to learn an adaptive clustering of the patches. Since overlapping patches are used, each pixel in the image typically belongs to many overlapping patches. We cluster a pixel into a particular class C_k if the majority of the patches to which it belongs, are clustered into that class by (P5).

We use 9 × 9 (overlapping mean-subtracted) patches ($n = 81$), and set $s = 10$, $K = 2$, and λ_0 is set as in Section 6.2. We initialize OCTOBOS learning using the clustering result of the k-means algorithm. The two cluster transforms

[25] For two matrices A and B of same size, the cross-gram matrix is computed as AB^T.

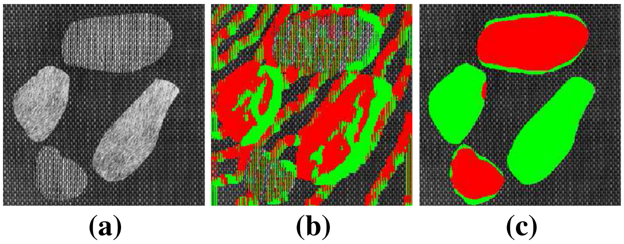

Fig. 7 $K = 3$ clustering example: **a** Input image, **b** Input image with pixels clustered into Classes 1 and 2 shown in *green* and *red*, respectively, for the K-means initialization, **c** Input image with pixels clustered into Classes 1 and 2 shown in *green* and *red*, respectively, for OCTOBOS (Color figure online)

are initialized with the DCT. We now use the aforementioned strategy to cluster the pixels of the input two-texture image into one of two classes using OCTOBOS. Figure 6c shows the clustering result obtained using OCTOBOS. As a comparison, Fig. 6b shows the image pixels clustered into each class for the k-means initialization. The proposed scheme is seen to improve over the k-means result. Alternative initializations for the OCTOBOS clusters such as random initialization also provide similar final clustering results, but typically require more iterations to converge.

Figure 7 shows clustering results for a 256 × 256 three texture image (Fig. 7a). The parameters for OCTOBOS are $K = 3$, and n, s, λ_0 are set just as for the case of Fig. 6. OCTOBOS (Fig. 7c) is again seen to improve over the k-means initialization (Fig. 7b).

The clustering examples here illustrate some *preliminary* potential for the OCTOBOS scheme in classification. We also observed reasonable clustering results with other texture images. Note that unlike prior work in synthesis dictionary-based classification (e.g., (Ramirez et al. 2010)), we do not have additional penalties in (P5) that discriminate (e.g., by enforcing incoherence) between the learnt transform blocks. An extension of our OCTOBOS scheme by incorporating such classification-specific penalties (and other classification-specific heuristics) may be useful for the classification application. We leave the detailed investigation of the classification application (for example, the study of potential discriminative OCTOBOS learning methods) for future work.

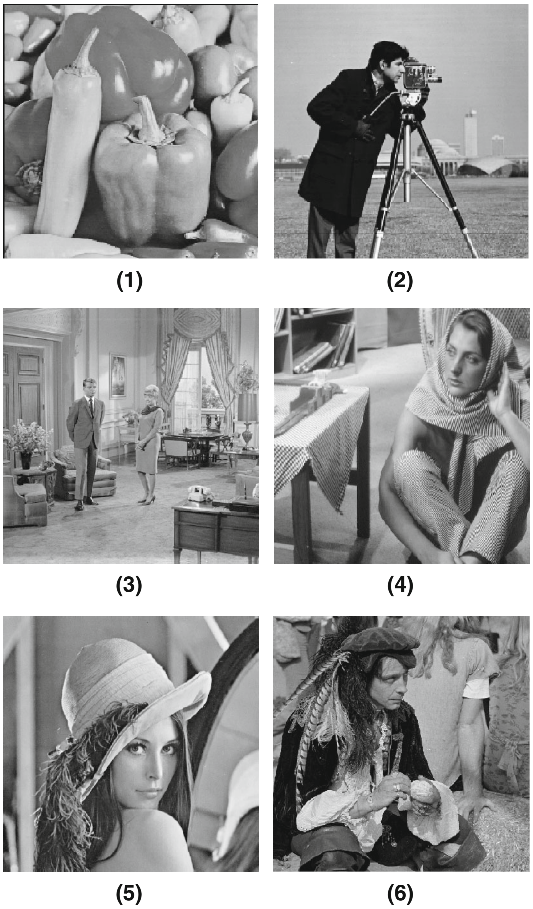

Fig. 8 Images used for sparse representation and denoising experiments: **1** Peppers, **2** Cameraman, **3** Couple, **4** Barbara, **5** Lena, **6** Man. These images are numbered 1 through 6 in our results

Table 2 Recovery PSNR for the learnt OCTOBOS transforms with different K, and for the learnt single square ($K = 1$) transforms (SQ) (Ravishankar and Bresler 2013a), and for DCT, and KLT

Image	DCT	KLT	SQ	OCTOBOS			
				$K = 2$	$K = 4$	$K = 8$	$K = 16$
1	33.2	32.8	35.2	36.0	38.0	41.1	**45.9**
2	30.0	29.7	32.0	33.3	35.9	40.9	**47.8**
3	34.0	33.5	34.5	35.0	35.6	36.5	**37.9**
4	32.9	31.7	34.5	35.4	36.0	36.6	**38.1**
5	36.9	36.8	37.6	38.2	38.8	39.8	**41.4**
6	32.5	32.3	33.1	33.5	33.9	34.5	**35.3**
Av.	33.2	32.8	34.5	35.2	36.4	38.2	**41.1**

The best PSNR values are marked in bold. The last row of the table provides average PSNR values computed over the six images

6.4 Sparse Representation of Images

In this section, we study the potential of the proposed OCTOBOS learning scheme for sparsely representing various images. We consider the six images shown in Fig. 8. The images Cameraman and Peppers have 256 × 256 pixels. The image Man has 768 × 768 pixels, and Couple, Barbara, and Lena have 512 × 512 pixels. We learn overcomplete transforms for each of the images by solving (P5). We consider 8 × 8 non-overlapping mean-subtracted patches, and set λ_0, s to the same values as in Sect. 6.2. Different levels of overcompleteness (K) are considered. We initialize the OCTOBOS learning with random clustering (each patch assigned uniformly at random to one of K clusters) and the

Table 3 NSE metric (in percentage) for the learnt OCTOBOS transforms with different K, and for the learnt single square ($K = 1$) transforms (SQ) (Ravishankar and Bresler 2013a), and for DCT, and KLT

Image	DCT	KLT	SQ	OCTOBOS			
				$K = 2$	$K = 4$	$K = 8$	$K = 16$
1	4.5	4.9	2.4	2.0	1.3	0.6	**0.2**
2	9.0	9.7	4.7	3.5	1.9	0.7	**0.1**
3	6.6	7.3	5.0	4.3	3.8	3.0	**2.3**
4	6.8	8.9	4.3	3.5	3.0	2.6	**1.8**
5	4.7	4.8	3.2	2.8	2.4	2.0	**1.4**
6	9.1	9.5	6.4	5.7	5.2	4.6	**3.8**
Av.	6.8	7.5	4.3	3.6	2.9	2.2	**1.6**

The best NSE values are marked in bold. The last row of the table provides average values computed over the six images

Table 4 Swapping experiment: the learnt OCTOBOS blocks for the $K = 2$ case are swapped between the two learnt clusters, and the NSE (in percentage) and recovery PSNR metrics are recomputed for various images

Image	1	2	3	4	5	6
RP_a	32.3	29.2	33.3	32.3	36.4	32.2
RP_b	36.0	33.3	35.0	35.4	38.2	33.5
RP_c	35.2	32.0	34.5	34.5	37.6	33.1
NSE_a	4.8	9.9	6.6	7.3	4.3	8.1
NSE_b	2.0	3.5	4.3	3.5	2.8	5.7
NSE_c	2.4	4.7	5.0	4.3	3.2	6.4
$\kappa(W_1)$	1.44	1.59	1.58	1.48	1.51	1.67
$\kappa(W_2)$	1.32	1.57	1.29	1.35	1.33	1.38

For comparison, we also include the corresponding metrics for the $K = 2$ (no swapping) and learnt square transform ($K = 1$) cases. The subscripts a, b, and c, are used to denote the following scenarios: a, Swapping experiment result; b, OCTOBOS (K = 2) result, and c, square (K = 1) result. The condition numbers of the two learnt OCTOBOS blocks ($\kappa(W_1)$ and $\kappa(W_2)$) are also provided for all images

2D DCT transform for the W_k's. For comparison, we also learn a square transform (i.e., the $K = 1$ case) for the images, which is equivalent to solving (P3) (Ravishankar and Bresler 2013a). All the transform learning schemes are run for 100 iterations.

Tables 2 and 3 list the NSE and recovery PSNR metrics for the various learnt OCTOBOS transforms, along with the corresponding values for the learnt (single) square transforms, the analytical transform of 2D DCT, and KLT. The learnt transforms (both OCTOBOS and square) provide significantly better sparsification and recovery compared to the DCT and KLT. Importantly, as K increases, the learnt OCTOBOS transforms provide increasingly better image representation compared to the learnt square transform. The recovery PSNR increases monotonically, and NSE decreases likewise, as K increases. This clearly indicates that different groups/clusters of patches in the images are better sparsified by different (rather than identical) adaptive transforms. Another interesting observation is regarding the amount of improvement in NSE and recovery PSNR that OCTOBOS provides compared to the learnt square transform for different images. Images that have more diverse features, or more patches (e.g., Barbara, Man) require a larger value of K to achieve a certain amount of improvement (for OCTOBOS vs. square) than images with less diverse features, or fewer patches (e.g., peppers, cameraman).

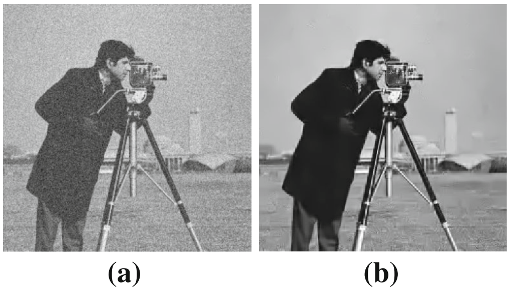

Fig. 9 Denoising result: **a** Noisy Cameraman (PSNR = 22.10 dB), **b** Denoised cameraman (PSNR = 30.24 dB) obtained using the OCTOBOS scheme

In order to further explore the relevance of the OCTOBOS model for the various images, we consider the following experiment with $K = 2$. For each image, we swap the two learnt OCTOBOS blocks between the two learnt clusters. We then recompute the NSE and recovery PSNR metrics for the images using the learnt clusters, but swapped transforms. The results are shown in Table 4. We observe that the metrics have significantly worsened (compared to the result without swapping) with the swapping of the transforms. This now clearly indicates that the two learnt clusters (or, equivalently, the two learnt transforms) are quite different from each other for the images. In fact, with swapping, the NSE and recovery PSNR results are worse than those obtained with a single learnt square transform, since in the latter case, the transform is at least learnt over all the image patches. Note that the learnt OCTOBOS blocks all have similar and good condition numbers in Table 4, as expected.

6.5 Image Denoising

We present preliminary results for our adaptive OCTOBOS-based image denoising framework (based on (P8)). We work with the six images shown in Fig. 8, and simulate i.i.d. Gaussian noise at 5 different noise levels ($\sigma = 5, 10, 15, 20, 100$) for each of the images. We compare the denoising results obtained by our proposed algorithm in Sect. 5, with those obtained by the adaptive overcomplete K-SVD denoising scheme (Elad and Aharon 2006), the GMM-based denoising method (Zoran and Weiss 2011), and the BM3D method (Dabov et al. 2007), which is a state-of-the-art image denois-

Table 5 PSNR values for denoising with 256 × 64 OCTOBOS transform, along with the corresponding values for denoising using BM3D (Dabov et al. 2007), the 64 × 256 overcomplete K-SVD (Elad and Aharon 2006), and the GMM method (Zoran and Weiss 2011). The PSNR values of the noisy images are also shown

Image	σ	Noisy PSNR	BM3D	K-SVD	GMM	OCTOBOS
Peppers	5	34.14	38.09	37.78	37.95	38.09
	10	28.10	34.66	34.24	34.51	34.57
	15	24.58	32.69	32.18	32.54	32.43
	20	22.12	31.33	30.80	31.18	30.97
	100	8.11	23.17	21.79	22.97	22.23
Cameraman	5	34.12	38.21	37.81	38.06	38.19
	10	28.14	34.15	33.72	34.00	34.15
	15	24.61	31.91	31.50	31.85	31.94
	20	22.10	30.37	29.82	30.21	30.24
	100	8.14	23.15	21.76	22.89	22.24
Couple	5	34.16	37.48	37.28	37.35	37.40
	10	28.11	34.01	33.51	33.79	33.73
	15	24.59	32.08	31.46	31.84	31.71
	20	22.11	30.78	30.02	30.51	30.34
	100	8.13	23.46	22.57	23.30	22.88
Barbara	5	34.15	38.30	38.08	37.59	38.31
	10	28.14	34.97	34.41	33.61	34.64
	15	24.59	33.05	32.33	31.28	32.53
	20	22.13	31.74	30.83	29.74	31.05
	100	8.11	23.61	21.87	22.13	22.41
Lena	5	34.16	38.70	38.61	38.55	38.71
	10	28.12	35.88	35.49	35.56	35.64
	15	24.63	34.26	33.74	33.87	33.92
	20	22.11	33.01	32.41	32.60	32.59
	100	8.14	25.75	24.51	25.24	25.17
Man	5	34.15	36.76	36.47	36.75	36.73
	10	28.13	33.18	32.71	33.14	32.98
	15	24.63	31.32	30.78	31.32	31.07
	20	22.11	30.03	29.40	30.02	29.74
	100	8.14	23.83	22.76	23.65	22.92

ing method. Note that as opposed to the K-SVD scheme, our OCTOBOS method is quite constrained due to the block cosparsity of the sparse code.

We work with 8×8 ($n = 64$) overlapping image patches in our experiments. For OCTOBOS-based denoising, we consider a 256×64 transform, i.e., $K = 4$. A corresponding 64×256 synthesis dictionary is used in the synthesis K-SVD denoising method. We fixed the initial sparsity levels s_i to 6 for all patches in our algorithm for (P8). We chose $C = 1.08$, and $\lambda_0 = 3.1 \times 10^{-2}$. We perform multiple passes through (P8), as discussed in Sect. 5.3. For each noise level (σ) of the original noisy image, the number of times that (P8) is solved, and the corresponding noise levels (for each pass through (P8)) were determined empirically[26]. These same parameters were used for all the images in our experiments. Other parameters in our algorithm such as the number of iterations (L, J in Fig. 2) for (P8) were set empirically. An example of OCTOBOS denoising is shown in Fig. 9.

Table 5 lists the PSNRs obtained by denoising with OCTOBOS, overcomplete K-SVD, GMM, and BM3D. First, the OCTOBOS scheme clearly provides better PSNRs than K-SVD for all images and noise levels. Comparing the PSNR values obtained by the 256×64 OCTOBOS to those of the 64×256 synthesis K-SVD dictionary for each image and noise level, we obtain an average PSNR improvement (average computed over all images and noise levels) of 0.30 dB for OCTOBOS over K-SVD. The improvement over K-SVD for individual examples is up to 0.66 dB in Table 5. Thus, the OCTOBOS method outperforms K-SVD despite using a constrained (block cosparse) transform. We also obtain an average speedup of 2.8× for OCTOBOS denoising over K-SVD

[26] The noise level estimates decrease over the iterations (passes through (P8)). We also found empirically that underestimating the noise standard deviation (during each pass through (P8)) led to better performance.

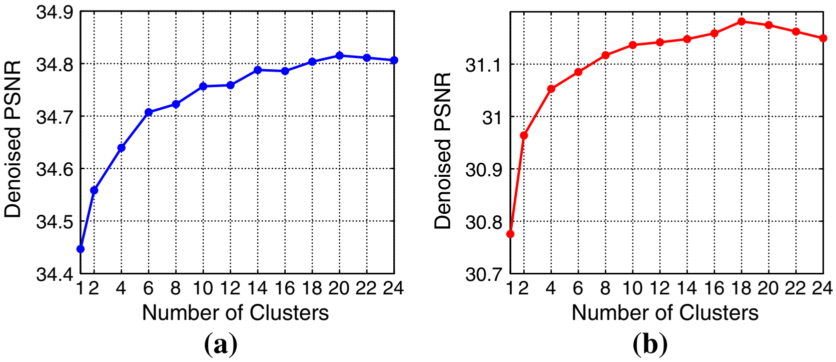

Fig. 10 Denoising PSNR for Barbara as a function of the number of clusters K: **a** $\sigma = 10$, **b** $\sigma = 20$

denoising.[27] This is because the various steps of OCTOBOS-based denoising such as the sparse coding and clustering step are computationally very cheap.

Our OCTOBOS denoising scheme is also 0.05 dB better on an average (over all images and noise levels) compared to GMM-based denoising in Table 5. Although the state-of-the-art BM3D method is quite better than OCTOBOS at $\sigma = 100$, OCTOBOS denoising is only 0.22 dB worse than BM3D on the average at other noise levels ($\sigma \leq 20$ in Table 5). OCTOBOS denoising also performs comparably to BM3D (at lower noise levels) for certain images such as Cameraman and Peppers.

Next, using the same parameters as in the preceding experiments, we study the behavior of OCTOBOS denoising as a function of the overcompleteness K of the transform. Figures 10a, b plot the denoising PSNRs for Barbara as a function of the number of clusters K for $\sigma = 10$ and $\sigma = 20$, respectively. In both cases, the denoising PSNR increases with K up to an optimal value of K, beyond which the PSNR begins to slowly drop. Initially, as K increases, the OCTOBOS model becomes richer, and thus, provides increasingly better denoising. However, when K becomes too large,[28] one cannot reliably learn all the OCTOBOS square blocks from the limited number of noisy training data associated with each block, without overfitting the noise. Thus, the PSNR begins to drop for very large K. This effect is more pronounced the higher the noise level, as seen in Fig. 10, where the optimal K where the plot peaks is lower for $\sigma = 20$, than for $\sigma = 10$. The same trend continues at $\sigma = 100$ (not shown in Fig. 10). The plots in Fig. 10 also illustrate the advantage (up to 0.4 dB improvement for this example) of OCTOBOS-based denoising over the single square transform-based ($K = 1$) denoising. This gap increases when the OCTOBOS parameters are better tuned for larger K.

Thus, our results for OCTOBOS-based denoising are quite comparable to, or better than the results obtained by previous image denoising schemes such as GMM denoising, BM3D, K-SVD denoising, and adaptive square transform denoising. The learned OCTOBOS (square) blocks for all images and noise levels in our experiments, are well-conditioned (condition numbers of 1–2). We expect the denoising PSNRs for OCTOBOS to improve further with optimal parameter tuning. Our method is limited at very high noise (such as $\sigma = 100$) due to the fact that the learning is done using corrupted data. Therefore, in the high noise setting, using a fixed OCTOBOS transform (learnt over a database of images that share similar properties to the image being denoised) may provide better denoising. This topic is worthy of further future investigation. Moreover, since the state-of-the-art BM3D is a non-local method, we believe that a non-local extension to the OCTOBOS scheme could lead to even better OCTOBOS denoising performance. We plan to investigate such an extension in the near future.

7 Conclusions

In this paper, focusing on the transform model for sparse representations, we presented a novel union of sparsifying transforms model. We showed that this model can also be interpreted as an overcomplete sparsifying transform model with an extra block cosparsity constraint (OCTOBOS) on the sparse code. The sparse coding in the proposed model can be interpreted as a form of clustering. We presented a novel problem formulation and algorithm for learning the proposed OCTOBOS transforms. Our algorithm involves simple closed-form solutions, and is thus computationally very efficient. Our theoretical analysis established global convergence (i.e., convergence from any initialization) of the algorithm to the set of partial minimizers of the non-convex learning problem. For natural images, our learning scheme gives rise to a union of well-conditioned transforms, and clustered patches or textures. It is also usually insensitive to initialization. The adapted model provides better sparsification errors and recovery PSNRs for images compared to

[27] Our MATLAB implementation of OCTOBOS denoising is not currently optimized for efficiency. Therefore, the speedup here is computed by comparing our unoptimized MATLAB implementation to the corresponding MATLAB implementation (Elad 2009) of K-SVD denoising.

[28] Compare this behavior to the monotone increase with K of the recovery PSNR for image representation (see Sect. 6.4).

learnt single square transforms, and analytical transforms. In the application of image denoising, the proposed scheme typically provides better image reconstruction quality compared to adaptive (single) square transforms, and adaptive overcomplete synthesis dictionaries. These results suggest that the proposed OCTOBOS learning produces effective models adapted to the data. The usefulness of our OCTOBOS learning scheme in these applications merits detailed study and extensive evaluation. Likewise, other applications, e.g., inverse problems such as MRI (Ravishankar and Bresler 2013d) and CT (Pfister 2013; Pfister and Bresler 2014), and classification merit further study.

Acknowledgments Part of this work was supported by the National Science Foundation (NSF) under Grants CCF-1018660 and CCF-1320953.

Appendix: Useful Lemmas

Here, we list three results (from (Ravishankar and Bresler 2014)) that are used in our convergence proof. The following result is from the Appendix of (Ravishankar and Bresler 2014).

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

The following result is based on the proof of Lemma 6 of (Ravishankar and Bresler 2014).

Lemma 11 *Let* $\{W^{q_t}, X^{q_t}\}$ *with* $W^{q_t} \in \mathbb{R}^{n\times n}$, $X^{q_t} \in \mathbb{R}^{n\times N}$, *be a subsequence of* $\left\{W^t, X^t\right\}$ *converging to the accumulation point* (W^*, X^*). *Let* $Z \in \mathbb{R}^{n\times N}$ *and* $L^{-1} = \left(ZZ^T + \lambda I\right)^{-1/2}$, *with* $\lambda > 0$. *Further, let* $Q^{q_t}\Sigma^{q_t}(R^{q_t})^T$ *denote the full singular value decomposition of* $L^{-1}Z(X^{q_t})^T$. *Let*

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

and suppose that $\left\{W^{q_t+1}\right\}$ *converges to* W^{**}. *Then,*

$$W^{**} \in \arg\min_{W} \left\| WZ - X^*\right\|_F^2 + \lambda \left\|W\right\|_F^2 - \lambda \log\,|\det W| \tag{53}$$

The following result is based on the proof of Lemma 9 of (Ravishankar and Bresler 2014). Note that $\phi(X)$ is the barrier function defined in Section 4.

Lemma 12 *Given* $Z \in \mathbb{R}^{n\times N_1}$, $\lambda > 0$, *and* $s \geq 0$, *consider the function* $g : \mathbb{R}^{n\times n} \times \mathbb{R}^{n\times N_1} \mapsto \mathbb{R}$ *defined as* $g(W, X) = \|WZ - X\|_F^2 + \lambda\|W\|_F^2 - \lambda \log\,|\det W| + \phi(X)$ *for* $W \in \mathbb{R}^{n\times n}$ *and* $X \in \mathbb{R}^{n\times N_1}$. *Further, let* $(\hat{W}, \hat{X})$ *be a pair in* $\mathbb{R}^{n\times n} \times \mathbb{R}^{n\times N_1}$ *satisfying*

$$2\hat{W}ZZ^T - 2\hat{X}Z^T + 2\lambda\hat{W} - \lambda\hat{W}^{-T} = 0 \tag{54}$$

$$\hat{X}_i \in \tilde{H}_s(\hat{W}Z_i), \quad \forall\, 1 \leq i \leq N_1 \tag{55}$$

Then, the following condition holds at $(\hat{W}, \hat{X})$.

$$g(\hat{W} + dW, \hat{X} + \Delta X) \geq g(\hat{W}, \hat{X}) \tag{56}$$

The condition holds for all sufficiently small $dW \in \mathbb{R}^{n\times n}$ *satisfying* $\|dW\|_F \leq \epsilon'$ *for some* $\epsilon' > 0$ *that depends on* $\hat{W}$, *and all* $\Delta X \in \mathbb{R}^{n\times N}$ *in the union of the following regions.*

R1. *The half-space* $tr\left\{(\hat{W}Z - \hat{X})\Delta X^T\right\} \leq 0$.
R2. *The local region defined by*
$\|\Delta X\|_\infty < \min_i \left\{\beta_s(\hat{W}Z_i) : \left\|\hat{W}Z_i\right\|_0 > s\right\}$.

Furthermore, if we have $\left\|\hat{W}Z_i\right\|_0 \leq s\,\forall i$, *then* ΔX *can be arbitrary.*

The following lemma is a slightly modified version of the one in (Ravishankar et al. 2014). We only state the minor modifications to the previous proof, for the following lemma to hold.

The lemma implies Lipschitz continuity (and therefore, continuity) of the function $u(B) \triangleq \|By - H_s(By)\|_2^2$ on a bounded set.

Lemma 13 *Given* $c_0 > 0$, *and* $y \in \mathbb{R}^n$ *satisfying* $\|y\|_2 \leq c_0$, *and a constant* $c' > 0$, *the function* $u(B) = \|By - H_s(By)\|_2^2$ *is uniformly Lipschitz with respect to* B *on the bounded set* $S \triangleq \left\{B \in \mathbb{R}^{n\times n} : \|B\|_2 \leq c'\right\}$.

Proof The proof is identical to that for Lemma 4 in (Ravishankar et al. 2014), except that the conditions $\|y\|_2 = 1$ and $\|B\|_2 \leq 1$ in (Ravishankar et al. 2014) are replaced by the conditions $\|y\|_2 \leq c_0$ and $\|B\|_2 \leq c'$ for the proof here. □

References

Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P., & Tandon, R. (2013). Learning sparsely used overcomplete dictionaries via alternating minimization, arXiv:1310.7991, Preprint.

Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P., & Tandon, R. (2014). Learning sparsely used overcomplete dictionaries. *Journal of Machine Learning Research*, *35*, 1–15.

Aharon, M., & Elad, M. (2008). Sparse and redundant modeling of image content using an image-signature-dictionary. *SIAM Journal on Imaging Sciences*, *1*(3), 228–247.

Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, *54*(11), 4311–4322.

Arora, S., Ge, R., & Moitra, A. (2013). New algorithms for learning incoherent and overcomplete dictionaries. arXiv:1308.6273v5.pdf, Preprint

Bao, C., Ji, H., Quan, Y., & Shen, Z. (2014). ℓ_0 Norm based dictionary learning by proximal methods with global convergence. In *IEEE Conference on Computer Vision and Pattern Recognition*. Online: http://www.math.nus.edu.sg/~matzuows/BJQS.pdf, to appear

Brodatz, P. (1966). *Textures: A photographic album for artists and designers*. New York: Dover.

Bruckstein, A. M., Donoho, D. L., & Elad, M. (2009). From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, *51*(1), 34–81.

Candès, E. J., Donoho, D. L. (1999). Curvelets: A surprisingly effective nonadaptive representation for objects with edges. In *Curves and surfaces* (pp. 105–120). Nashville: Vanderbilt University Press.

Candès, E. J., & Donoho, D. L. (1999). Ridgelets: A key to higher-dimensional intermittency? *Philosophical Transactions of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences*, *357*(1760), 2495–2509.

Candès, E. J., Eldar, Y. C., Needell, D., & Randall, P. (2011). Compressed sensing with coherent and redundant dictionaries. *Applied and Computational Harmonic Analysis*, *31*(1), 59–73.

Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, *20*(1–2), 89–97.

Chen, S. S., Donoho, D. L., & Saunders, M. A. (1998). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, *20*(1), 33–61.

Chen, Y., Pock, T., & Bischof, H. (2012a). Learning ℓ_1-based analysis and synthesis sparsity priors using bi-level optimization. In *Proceedings of the Workshop on Analysis Operator Learning vs. Dictionary Learning*, NIPS. arXiv:1401.4105

Chen, Y. C., Sastry, C. S., Patel, V. M., Phillips, P. J., & Chellappa, R. (2012b). Rotation invariant simultaneous clustering and dictionary learning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1053–1056).

Chi, Y. T., Ali, M., Rajwade, A., & Ho, J. (2013). Block and group regularized sparse modeling for dictionary learning. In *CVPR* (pp. 377–382).

Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2007). Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, *16*(8), 2080–2095.

Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. (2011). BM3D web page. Retrieved 2014, from http://www.cs.tut.fi/~foi/GCF-BM3D/

Dai, W., & Milenkovic, O. (2009). Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, *55*(5), 2230–2249.

Davis, G., Mallat, S., & Avellaneda, M. (1997). Adaptive greedy approximations. *Journal of Constructive Approximation*, *13*(1), 57–98.

Do, M. N., & Vetterli, M. (2005). The contourlet transform: An efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, *14*(12), 2091–2106.

Donoho, D. L., & Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization. *Proceedings of the National Academy of Sciences*, *100*(5), 2197–2202.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, *32*, 407–499.

Elad, M. (2009). Michael Elad personal page. http://www.cs.technion.ac.il/~elad/Various/KSVD_Matlab_ToolBox.zip. Accessed 2014.

Elad, M., & Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, *15*(12), 3736–3745.

Elad, M., Milanfar, P., & Rubinstein, R. (2007). Analysis versus synthesis in signal priors. *Inverse Problems*, *23*(3), 947–968.

Engan, K., Aase, S., & Hakon-Husoy, J. (1999). Method of optimal directions for frame design. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (pp. 2443–2446).

Giryes, R., Nam, S., Elad, M., Gribonval, R., & Davies, M. (2014). Greedy-like algorithms for the cosparse analysis model. *Linear Algebra and its Applications*, *441*:22–60, Special Issue on Sparse Approximate Solution of Linear Systems.

Gorodnitsky, I. F., George, J., & Rao, B. D. (1995). Neuromagnetic source imaging with FOCUSS: A recursive weighted minimum norm algorithm. *Electrocephalography and Clinical Neurophysiology*, *95*, 231–251.

Harikumar, G., & Bresler, Y. (1996). A new algorithm for computing sparse solutions to linear inverse problems. In *ICASSP* (pp. 1331–1334).

Hawe, S., Kleinsteuber, M., & Diepold, K. (2013). Analysis operator learning and its application to image reconstruction. *IEEE Transactions on Image Processing*, *22*(6), 2138–2150.

He, D.-C., & Safia, A. (2013). Multiband Texture Database. http://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html. Accessed 2014.

Kong, S., & Wang, D. (2012). A dictionary learning approach for classification: Separating the particularity and the commonality. In *Proceedings of the 12th European Conference on Computer Vision* (pp 186–199).

Liao, H. Y., & Sapiro, G. (2008). Sparse representations for limited data tomography. In *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)* (pp 1375–1378).

Liu, Y., Tiebin, M., & Li, S. (2012). Compressed sensing with general frames via optimal-dual-based ℓ_1-analysis. *IEEE Transactions on Information Theory*, *58*(7), 4201–4214.

Mairal, J., Elad, M., & Sapiro, G. (2008a). Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, *17*(1), 53–69.

Mairal, J., Sapiro, G., & Elad, M. (2008b). Learning multiscale sparse representations for image and video restoration. *SIAM, Multiscale Modeling and Simulation*, *7*(1), 214–241.

Mairal, J., Bach, F., Ponce, J., Sapiro, G, & Zisserman, A. (2009). Non-local sparse models for image restoration. In *IEEE International Conference on Computer Vision* (pp. 2272–2279).

Mairal, J., Bach, F., Ponce, J., & Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, *11*, 19–60.

Mallat, S. (1999). *A wavelet tour of signal processing*. Boston: Academic Press.

Mallat, S. G., & Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, *41*(12), 3397–3415.

Marcellin, M. W., Gormish, M. J., Bilgin, A., & Boliek, M. P. (2000). An overview of JPEG-2000. In Proceedings of the Data Compression Conference (pp. 523–541).

Nam, S., Davies, M. E., Elad, M., & Gribonval, R. (2011). Cosparse analysis modeling: Uniqueness and algorithms. In *ICASSP* (pp. 5804–5807).

Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, *24*(2), 227–234.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607–609.

Ophir, B., Elad, M., Bertin, N., & Plumbley, M. (2011). Sequential minimal eigenvalues: An approach to analysis dictionary learning. In *Proceedings of the European Signal Processing Conference (EUSIPCO)* (pp. 1465–1469).

Pati, Y., Rezaiifar, R., & Krishnaprasad, P. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Asilomar Conference on Signals, Systems and Computers* (Vol. 1, pp. 40–44).

Peleg, T., & Elad, M. (2014). A statistical prediction model based on sparse representations for single image super-resolution. *IEEE Transactions on Image Processing*, *23*(6), 2569–2582.

Peyré, G., & Fadili, J. (2011). Learning analysis sparsity priors. In *Proceedings of the International Conference on Sampling Theory and Applications (SampTA)*, Singapore. http://hal.archives-ouvertes.fr/hal-00542016/. Accessed 2014.

Pfister, L. (2013). Tomographic reconstruction with adaptive sparsifying transforms. Master's Thesis, University of Illinois at Urbana-Champaign.

Pfister, L., & Bresler, Y. (2014). Model-based iterative tomographic reconstruction with adaptive sparsifying transforms. In S. P. I. E. International (Ed.), *Symposium on Electronic Imaging: Computational Imaging XII*, to appear.

Pratt, W. K., Kane, J., & Andrews, H. C. (1969). Hadamard transform image coding. *Proceedings of the IEEE*, *57*(1), 58–68.

Ramirez, I., Sprechmann, P., & Sapiro, G. (2010). Classification and clustering via dictionary learning with structured incoherence and shared features. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3501–3508).

Ravishankar, S., & Bresler, Y. (2011a). MR image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE Transactions on Medical Imaging*, *30*(5), 1028–1041.

Ravishankar, S., & Bresler, Y. (2011b). Multiscale dictionary learning for MRI. In *Proceedings of ISMRM* (p. 2830).

Ravishankar, S., & Bresler, Y. (2012a). Learning doubly sparse transforms for image representation. In *IEEE International Conference on Image Processing* (pp 685–688).

Ravishankar, S., & Bresler, Y. (2012b). Learning sparsifying transforms for signal and image processing. In *SIAM Conference on Imaging Science* (p. 51).

Ravishankar, S., & Bresler, Y. (2013a). Closed-form solutions within sparsifying transform learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE (pp. 5378–5382).

Ravishankar, S., & Bresler, Y. (2013b). Learning doubly sparse transforms for images. *IEEE Transactions on Image Processing*, *22*(12), 4598–4612.

Ravishankar, S., & Bresler, Y. (2013c). Learning sparsifying transforms. *IEEE Transactions on Signal Processing*, *61*(5), 1072–1086.

Ravishankar, S., & Bresler, Y. (2013d). Sparsifying transform learning for compressed sensing MRI. In *Proceedings of the IEEE International Symposium on Biomedical Imaging* (pp. 17–20).

Ravishankar, S., & Bresler, Y. (2014). ℓ_0 Sparsifying transform learning with efficient optimal updates and convergence guarantees. In *IEEE Transactions on Signal Processing*. (submitted). https://uofi.box.com/s/vrw0i13jbkj6n8xh9u9h. Accessed 2014.

Ravishankar, S., & Bresler, Y. (2014). Online sparsifying transform learning: Part II: Convergence analysis. *IEEE Journal of Selected Topics in Signal Process*. (accepted). https://uofi.box.com/s/cmqme2avnz5pygobxj3u. Accessed 2014.

Rubinstein, R., & Elad, M. (2011). K-SVD dictionary-learning for analysis sparse models. In *Proceedings of SPARS11* (p. 73).

Rubinstein, R., Bruckstein, A. M., & Elad, M. (2010). Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, *98*(6), 1045–1057.

Rubinstein, R., Faktor, T., & Elad, M. (2012). K-SVD dictionary-learning for the analysis sparse model. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5405–5408).

Sadeghi, M., Babaie-Zadeh, M., & Jutten, C. (2013). Dictionary learning for sparse representation: A novel approach. *IEEE Signal Processing Letters*, *20*(12), 1195–1198.

Sahoo, S. K., & Makur, A. (2013). Dictionary training for sparse representation as generalization of k-means clustering. *IEEE Signal Processing Letters*, *20*(6), 587–590.

Skretting, K., & Engan, K. (2010). Recursive least squares dictionary learning algorithm. *IEEE Transactions on Signal Processing*, *58*(4), 2121–2130.

Smith, L. N., & Elad, M. (2013). Improving dictionary learning: Multiple dictionary updates and coefficient reuse. *IEEE Signal Processing Letters*, *20*(1), 79–82.

Spielman, D. A., Wang, H., & Wright, J. (2012). Exact recovery of sparsely-used dictionaries. In *Proceedings of the 25th Annual Conference on Learning Theory* (pp. 37.1–37.18).

Sprechmann, P., Bronstein, A., & Sapiro, G. (2012a). Learning efficient structured sparse models. In *Proceedings of the 29th International Conference on Machine Learning* (Vol. 1, pp. 615–622).

Sprechmann, P., Bronstein, A. M., Sapiro, G. (2012b). Learning efficient sparse and low rank models. arXiv:1212.3631, Preprint.

Wang, S., Zhang, L., Liang, Y., Pan, Q. (2012). Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2216–2223).

Weiss, Y. (2011). Yair Weiss home page. Retrieved 2014, from http://www.cs.huji.ac.il/~daniez/epllcode.zip.

Xu, Y., Yin, W. (2013). A fast patch-dictionary method for whole-image recovery ftp://ftp.math.ucla.edu/pub/camreport/cam13-38.pdf, UCLA CAM report 13–38.

Yaghoobi, M., Blumensath, T., & Davies, M. (2009). Dictionary learning for sparse approximations with the majorization method. *IEEE Transaction on Signal Processing*, *57*(6), 2178–2191.

Yaghoobi, M., Nam, S., Gribonval, R., & Davies, M. (2011). Analysis operator learning for overcomplete cosparse representations. In *European Signal Processing Conference (EUSIPCO)* (pp. 1470–1474).

Yaghoobi, M., Nam, S., Gribonval, R., & Davies, M. E. (2012). Noise aware analysis operator learning for approximately cosparse signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5409–5412).

Yaghoobi, M., Nam, S., Gribonval, R., & Davies, M. E. (2013). Constrained overcomplete analysis operator learning for cosparse signal modelling. *IEEE Transactions on Signal Processing*, *61*(9), 2341–2355.

Yu, G., Sapiro, G., & Mallat, S. (2012). Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *IEEE Transactions on Image Processing*, *21*(5), 2481–2499.

Zelnik-Manor, L., Rosenblum, K., & Eldar, Y. C. (2012). Dictionary optimization for block-sparse representations. *IEEE Transactions on Signal Processing*, *60*(5), 2386–2395.

Zoran, D., & Weiss, Y. (2011). From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Computer Vision* (pp. 479–486).