Learning Overcomplete and Sparsifying Transform with Approximate and Exact Closed Form Solutions

Dimche Kostadinov
Department of Computer Science,
University of Geneva
Carouge 1227, Switzerland
dimche.kostadinov@unige.ch

Sviatoslav Voloshynovskiy
Department of Computer Science,
University of Geneva
Carouge 1227, Switzerland
svolos@unige.ch

Sohrab Ferdowsi
Department of Computer Science,
University of Geneva
Carouge 1227, Switzerland
sohrab.ferdowsi@unige.ch

***Abstract*—This paper addresses the learning problem for data-adaptive transform that provides sparse representation in a space with dimensions larger than (or equal to) the dimensions of the original space. We present an iterative, alternating algorithm that has two steps: (i) transform update and (ii) sparse coding. In the transform update step, we focus on novel problem formulation based on a lower bound of the objective that addresses a trade-off between (a) how much are aligned the gradients of the approximative objective and the original objective, and (b) how much the lower bound is close to the original objective. This allows us not only to propose approximate closed form solution but also gives the possibility to find an update that can lead to accelerated local convergence and enables us to estimate an update that can lead to a satisfactory solution under a small amount of data. Since in the transform update, the approximate closed form solution preserves the gradient and in the sparse coding step, we use exact closed form solution, the resulting algorithm is convergent. On the practical side, we evaluate on image denoising application and demonstrate promising denoising performance together with advantages in training data requirements, accelerated local convergence and the resulting computational complexity.**

Index terms— Sparse representation, sparsifying transform learning, image denoising.

I. Introduction

Nowadays, in many areas, a common practice is to process, analyze, recognize, classify data, etc. in a *transform domain*. Due to the presence of noise (or data variability) specific data properties or prior knowledge in the form of assumption has to be taken into account.

The fundamental concept that was widely exploited in the past decade, addressing *data adaptive* processing and data analysis is a *sparse representation*. Given a data sample $\mathbf{x} \in \Re^N$ and a set of vectors $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_M] \in \Re^{N \times M}$ (formally known as a frame[1]), a sparse representation $\mathbf{y} \in \Re^M$ for $\mathbf{x}$ over $\mathbf{B}$ is one that uses a sparse (small) number of vectors $\mathbf{b}_i \in \Re^N$ from $\mathbf{B}$ to represent $\mathbf{x}$. Three main models were proposed for sparse signal representations: the *synthesis model* [1], the *analysis model* (*noisy signal analysis model* [2]) and the *sparsifying transform model* [3].

[1]A set of M orthonormal vectors with vector dimensionality N equal to M is said to form a basis set for that vector space. A frame of an inner product space is a generalization of a basis of a vector space to sets that may be linearly dependent.

Learning any one of the previous model is challenging when the model matrix is overcomplete[2]. Several algorithms [4], [5] and [6] were proposed for learning analysis and sparsifying models with well conditioned, non-structured and overcomplete matrix. To find a solution, these algorithms typically alternate between update on the transform matrix and estimate for the sparse representations. Usually, the transform update step is based on the gradient of the objective where a solution is obtained by iteratively taking one or several gradient steps. Depending on the used algorithm for an update this might add computational complexity. On the other hand, the existence of a closed form solution (unique or not) w.r.t. the optimization objective or its approximation in the transform update step and the algorithm convergence for that case is not fully explored.

A. Contributions

This paper addresses the sparsifying transform model with overcomplete matrix and presents the following contributions:

(i) we propose iterative, alternating algorithm for learning overcomplete sparsfing transform with two steps: transform update step and sparse coding step,

(ii) we introduce a constrained problem formulation for the transform update step with objective that represents a lower bound approximation to the original objective of the related transform estimation problem,

(iii) we propose approximate closed form solution that addresses a trade-off between (a) how much are aligned the gradients of the approximative objective and the original objective and (b) how much the lower bound is close to the original objective,

(iv) we give a convergence result for the iterating sequence of the objective function values generated by the iterating steps of the proposed algorithm with exact and approximate closed form solutions and

(v) we present an evaluation by computer simulation in the image denosing application, showing competitive performance while using small amount of the noisy data used for learning.

[2]A matrix $\mathbf{A} \in \Re^{M \times N}$ is said to be overcomplete if $M > N$. Equivalently, if the number M of columns $\mathbf{a}_m \in \Re^N$ in $\mathbf{A}^T$ is bigger than the dimensionality N of $\mathbf{a}_m$, *i.e.*, $M > N$, we might also say that the set of vectors $\{\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_M\}$ is linearly dependent and that this set forms a frame.

II. Prior Work

In the following we introduce the common sparse signal models.

A. Synthesis Model

As the name suggests, the *synthesis model synthesizes* a data sample $\mathbf{x} \in \Re^N$ as an approximation by a linear combination $\mathbf{y} \in \Re^M$ (referred to as a sparse data representation) of a few words (frame vectors) $\|\mathbf{y}\|_0 << M$, from a dictionary (frame) $\mathbf{D} \in \Re^{N \times M}$, *i.e.*, $\mathbf{x} = \mathbf{D}\mathbf{y} + \mathbf{v}$, where $\mathbf{v} \in \Re^N$ denotes the approximation error. With the *synthesis model* approach the *data reconstruction* is addressed explicitly. This model assumes that the data $\mathbf{x}$ lies in the column space of the dictionary $\mathbf{D}$, with the error vector $\mathbf{v}$ defined in the original data domain. The two main open issues with this model are the high computational complexity for the learning of the dictionary $\mathbf{D}$ and the estimation of the sparse representation $\mathbf{y}$. The problem of learning a synthesis dictionary is NP-hard. Nonetheless, in the recent years many algorithms have been developed for the solution of the dictionary learning problem [1], [7], [8], [9] and [10].

B. Analysis Model

It uses a dictionary $\mathbf{\Omega} \in \Re^{M \times N}$ with $M > N$ to *analyze* the signal $\mathbf{x} \in \Re^N$. This model assumes that the product of $\mathbf{\Omega}$ and $\mathbf{x}$ is sparse, *i.e.*, $\mathbf{y} = \mathbf{\Omega}\mathbf{x}$ with $\|\mathbf{y}\|_0 = M - s$, where $0 \leq s \leq M$ is the number of zeros in $\mathbf{y} \in \Re^M$ [11] and [4]. The vector $\mathbf{y}$ is the analysis sparse representation of the data $\mathbf{x}$ w.r.t. $\mathbf{\Omega}$. If the data sample $\mathbf{x}$ is known, its analysis representation w.r.t. a given $\mathbf{\Omega}$ can be obtained via multiplying $\mathbf{x}$ by $\mathbf{\Omega}$. However, when the observed signal is contaminated by noise, the clean signal $\mathbf{x}$ has to be estimated first in order to get its analysis representation, which leads to the *analysis pursuit problem* [11]. Several algorithms have been proposed for analysis dictionary learning [11], [4], [12], [5] and [13]. The authors in [14] give a comprehensive overview of different learning methods for the analysis model. Again for this class of algorithms the computational complexity is an open issues. Moreover, it is ever higher compared to the previous model if the analysis pursuit problem [11] is considered, coupled with the estimate of its dictionary.

C. Transform Model

In contrast to the synthesis model and similarity to the analysis model, the *sparsifying transform model* does not targeted explicitly the data reconstruction. This model assumes that the data sample $\mathbf{x}$ is approximately sparsifiable under a linear transform $\mathbf{A} \in \Re^{M \times N}$, *i.e.*, $\mathbf{A}\mathbf{x} = \mathbf{y} + \mathbf{z}$, $\mathbf{z} \in \Re^M$, where $\mathbf{y}$ is sparse $\|\mathbf{y}\|_0 << M$. The error vector $\mathbf{z}$ is defined in the transform domain, which is different compared to the two previous models. Note that the first advantage of the sparsifiying transform model is that it extends and represents a generalization of the analysis model [15] since there is no explicit assumption on the sparse representation $\mathbf{y}$ or on the data sample $\mathbf{x}$. The sparse encoding in this model is a *direct problem* which is a converse to the *inverse problem* in the synthesis model. The *sparsifying transform model* was introduced in [16]. The sparsifying transform having a square matrix was studied in [15], the sparsifying transform having a structured set of square matrices and non-structured overcomplete matrix $\mathbf{A} \in \Re^{M \times N}, M \geq N$ were studied in [6], [17] and [18].

D. Paper Organization

The rest of the paper is organized as follows. Section 3, presents the problem formulation. Subsection $3.A.$ presents the learning algorithm. Subsection $3.B.$ presents the algorithm convergence result and subsection $3.C.$ gives a block level image denoising formulation. Section 4 is devoted to computer simulation and Section 5 concludes the paper.

III. Problem Formulation

Assume a data matrix $\mathbf{X} \in \Re^{N \times L}$ is given that has as columns data samples $\mathbf{x}_l \in \Re^N$, where $l \in \mathcal{L} = \{1, ..., L\}$ and L is the number of data samples. We address the learning of approximately sparsifying transform having overcomplete transform matrix $\mathbf{A} \in \Re^{M \times N}, M > N$ by the following problem formulation:

$$\underset{\mathbf{A},\mathbf{Y}}{min} \|\mathbf{AX} - \mathbf{Y}\|_F^2 + \Omega_1(\mathbf{A}), \text{ subject to } \|\mathbf{y}_l\|_0 \leq s, \forall l \in \mathcal{L}, \quad (1)$$

where $\|.\|_F$ and $\|.\|_0$ denotes the Frobenius and ℓ_0-"norm", respectively and $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_L] \in \Re^{M \times L}$ has as columns the transform representations $\mathbf{y}_l$. The first term in (1) is the *sparsification error* [3], it represents the deviation of the linearly transform data $\mathbf{AX}$ from the exact sparse representation $\mathbf{Y}$ in the transform domain. The penalty $\Omega_1(\mathbf{A})$ on the transform matrix $\mathbf{A}$ is defined as $\Omega_1(\mathbf{A}) = \frac{\lambda_1}{2}\|\mathbf{A}\|_F^2 + \frac{\lambda_2}{2}\|\mathbf{A}\mathbf{A}^T - \mathbf{I}\|_F^2 - \lambda_3 \log|\det \mathbf{A}^T\mathbf{A}|$, where λ_k are Lagrangian multipliers $\forall k \in \{1, 2, 3\}$. The second term $\Omega_1(\mathbf{A})$ and the penalty $\|\mathbf{y}_i\|_0, \forall i \in \mathcal{I}$ induce constraints on the properties of the matrix $\mathbf{A}$ and the transform representations $\mathbf{Y}$, respectively. The $\|\mathbf{A}\|_F^2$ penalty helps regularize the scale ambiguity in the solution of (1), that occurs when the data samples have representations with zero valued components. The $\log|\det(\mathbf{A}^T\mathbf{A})|$ and $\|\mathbf{A}\|_F^2$ are functions of the singular values of $\mathbf{A}$ and together help regularize the conditioning of $\mathbf{A}$ [15], [6], [18], [19]. Assuming that the expected coherence $\mu^2(\mathbf{A})$ between the rows $\mathbf{a}_m$ of $\mathbf{A}$, *i.e.*, $\mathbf{A}^T = [\mathbf{a}_1, ..., \mathbf{a}_M]$ is defined as $\mu^2(\mathbf{A}) = \frac{2}{M(M-1)} \sum_{m_1 \neq m_2} |\mathbf{a}_{m_1}\mathbf{a}_{m_2}^T|^2$, $\forall m_1, m_2 \in \mathcal{M} = \{1, .., M\}$, then the penalty $\|\mathbf{A}\mathbf{A}^T - \mathbf{I}\|_F^2$ helps enforce a minimum expected coherence $\mu^2(\mathbf{A})$ and unit ℓ_2-norm for the rows of $\mathbf{A}$.

The transform data $\mathbf{y}_l$ are constrained to have s non-zero elements by the sparsity inducing ℓ_0-"norm" $\|\mathbf{y}_l\|_0 \leq s, l \in \mathcal{L}$.

A. Learning Algorithm

Problem (1) is non-convex in the variables $\{\mathbf{A}, \mathbf{Y}\}$. If the variable $\mathbf{A}$ is fixed, (1) is convex, however if $\mathbf{Y}$ is fixed (1) remains non-convex because the matrix $\mathbf{A}\mathbf{A}^T$ in the penalty function Ω_1 has the term $\mathbf{A}\mathbf{A}^T$ to the power of 2 and the penalty $-\log|\det \mathbf{A}^T\mathbf{A}|$.

To solve (1) we use an iterative, alternating algorithm that has two steps: *transform estimate* and *sparse coding*. In the transform estimate step, given $\mathbf{Y}^t$ that is estimated at iteration t, we use *approximate closed form* solution to estimate the transform matrix $\mathbf{A}^{t+1}$ at iteration $t+1$. In the sparse coding step, given $\mathbf{A}^{t+1}$, the sparse codes $\mathbf{y}_l^{t+1}$ are estimated by a closed form solution.

Transform estimate Let the transform data $\mathbf{Y}^t$ at iteration t be known, then problem (1) reduces to a problem for estimation of the transform matrix $\mathbf{A}^{t+1}$ that is defined as follows:

$$(P1): \min_{\mathbf{A}^{t+1}} \overbrace{\|\mathbf{A}^{t+1}\mathbf{X}-\mathbf{Y}^t\|_F^2+\Omega_1(\mathbf{A}^{t+1})}^{g(\mathbf{A}^{t+1},\mathbf{Y}^t)}. \tag{2}$$

– *Alternative Problem Formulation for Transform Estimate* Instead of addressing problem $(P1)$, in this work, we introduce a constrained problem and focus on a objective that is a lower bound on the objective in problem $(P1)$, *i.e.*,

$$\min_{\mathbf{A}^{t+1}} \overbrace{Tr\{\mathbf{A}^{t+1}\mathbf{X}\mathbf{X}^T(\mathbf{A}^{t+1})^T - 2\mathbf{A}^{t+1}\mathbf{G} + (\mathbf{Y}^{t+1})^T\mathbf{Y}^t\} + \Omega_1(\mathbf{A}^{t+1})}^{g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)\le g(\mathbf{A}^{t+1},\mathbf{Y}^t)}$$

$$\textit{subject to} \quad \mathbf{A}^{t+1}=\mathbf{V}\mathbf{S}^T\mathbf{\Sigma}_A\mathbf{\Sigma}^{-1}\mathbf{U}^T \tag{3}$$
$$Tr\{\mathbf{\Sigma}_A\} \ge \frac{1}{\beta\lambda_{min}}Tr\{\mathbf{\Sigma}_C\},$$

where $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t) = Tr\{\mathbf{A}^{t+1}\mathbf{X}\mathbf{X}^T(\mathbf{A}^{t+1})^T - 2\mathbf{A}^{t+1}\mathbf{G} + (\mathbf{Y}^t)^T\mathbf{Y}^t\} + \Omega_1(\mathbf{A}^{t+1})$ is the lower bound for $g(\mathbf{A}^{t+1},\mathbf{Y}^t) = Tr\{\mathbf{A}^{t+1}\mathbf{X}\mathbf{X}^T(\mathbf{A}^{t+1})^T - 2\mathbf{A}^{t+1}\mathbf{X}(\mathbf{Y}^t)^T + (\mathbf{Y}^t)^T\mathbf{Y}^t\} + \Omega_1(\mathbf{A}^{t+1})$, with $\mathbf{G} \in \Re^{N\times M}$ as approximation for $\mathbf{X}(\mathbf{Y}^t)^T$. We assume that $\mathbf{A}^{t+1}$ decomposes as $\mathbf{A}^{t+1} = \mathbf{V}\mathbf{S}^T\mathbf{\Sigma}_A\mathbf{\Sigma}^{-1}\mathbf{U}^T$, where $\mathbf{V} \in \Re^{M\times N}$, $\mathbf{S} \in \Re^{N\times N}$ and $\mathbf{U} \in \Re^{N\times N}$ are tall column-wise orthonormal, and square orthonormal matrices, respectively, $\mathbf{\Sigma}_A \in \Re^{N\times N}$, $\mathbf{\Sigma} \in \Re^{N\times N}$, $\mathbf{\Sigma}_C \in \Re^{N\times N}$ are diagonal matrices, $\lambda_{min} \in \Re$ and $\beta \ge 0$.

We derive the matrix $\mathbf{\Sigma}_C$ and λ_{min} based on the matrices $\mathbf{V}$, $\mathbf{S}$, $\mathbf{\Sigma}$ and $\mathbf{U}$ and the first order derivative of $g(\mathbf{A}^t,\mathbf{Y}^t)$ w.r.t. $\mathbf{A}^t$ estimated at iteration t. We use the constraint $Tr\{\mathbf{\Sigma}_A\} \ge \frac{1}{\beta\lambda_{min}}Tr\{\mathbf{\Sigma}_C\}$ to ensure that the gradient of the approximated objective $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)$ is preserved w.r.t. gradient of the original objective $g(\mathbf{A}^{t+1},\mathbf{Y}^t)$.

In the following we present the approximate closed form solution for $(P1)$ which in fact represents the solution to (3). In addition, we explain how $\mathbf{V}$, $\mathbf{S}$, $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_A$ are estimated, and how $\mathbf{G}$, $\mathbf{\Sigma}_C$ and λ_{min} are constructed.

– *Approximate Closed Form Solution* Given $\mathbf{Y}^t \in \Re^{M\times L}$, $\mathbf{X} \in \Re^{N\times L}$, $M \ge N$, $\lambda_1 \ge 0, \lambda_2 \ge 0, \lambda_3 \ge 0$, let the eigen value decomposition $\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$ of $\mathbf{X}\mathbf{X}^T + \lambda_1\mathbf{I}$ and the singular value decomposition $\mathbf{S}\mathbf{D}\mathbf{V}^T$ of $\mathbf{U}^T\mathbf{X}(\mathbf{Y}^t)^T$ exist, then if and only if $\Sigma(n,n) = \sigma(n) > 0$, $\forall n \in \mathcal{N} = \{1,...,N\}$, the original problem $(P1)$ has approximate closed form solution as:

$$\mathbf{A}^{t+1} = \mathbf{V}\mathbf{S}^T\mathbf{\Sigma}_A\mathbf{\Sigma}^{-1}\mathbf{U}^T, \tag{4}$$

where $\mathbf{\Sigma}_A$ is a diagonal matrix, $\Sigma_A(n,n) = \sigma_A(n) \ge 0$, and $\sigma_A(n)$ are solutions to:

$$\min_{\sigma_A(n)} c_4(n)\sigma_A^4(n) + c_2(n)\sigma_A^2(n) - \frac{\sigma_\Gamma(n)}{\sigma(n)}\sigma_A(n) - f(\sigma_A(n)),$$
$$\text{subject to} \qquad \sigma_A(n) \ge \frac{1}{\beta\lambda_{min}}\sigma_C(n), \tag{5}$$

with $c_4(n) = \frac{\lambda_2}{\sigma^4(n)}, c_2(n) = \frac{\sigma^2(n)-2\lambda_2}{\sigma^2(n)}$, $f(\sigma_A(n)) = 2\lambda_3 \log\frac{\sigma(n)}{\sigma_A(n)}$, $\sigma_\Gamma(n) = T(n,n)$, $\mathbf{T} = \mathbf{S}\mathbf{D}$, $\sigma_C(n) = \Sigma_C(n,n)$, $\forall n \in \mathcal{N}$.

The diagonal matrix $\mathbf{\Sigma}_C$ contains the singular values of $\mathbf{C} = \left(\frac{\partial g(\mathbf{A}^t,\mathbf{Y}^t)}{\partial \mathbf{A}^t}\right)^T\mathbf{A}^t$, $\frac{\partial g(\mathbf{A}^t,\mathbf{Y}^t)}{\partial \mathbf{A}^t} = (\mathbf{A}^t\mathbf{X}-\mathbf{Y}^t)\mathbf{X}^T + \lambda_1\mathbf{A}^t + \lambda_2(\mathbf{A}^t(\mathbf{A}^t)^T-\mathbf{I})\mathbf{A}^t - \lambda_3((\mathbf{A}^t)^T)^{-1}$ is the first order derivative of $g(\mathbf{A}^t,\mathbf{Y}^t)$ w.r.t. to the estimate $\mathbf{A}^t$ at iteration t, λ_{min} is the smallest singular value of $\mathbf{F} = \left(\mathbf{\Sigma}^{-1}\mathbf{U}^T\right)\left(\frac{\partial g(\mathbf{A}^t,\mathbf{Y}^t)}{\partial \mathbf{A}^t}\right)^T\left(\mathbf{V}\mathbf{S}^T\right)$ and $\beta \ge 0$.

Proof: The matrix $\mathbf{G} = \mathbf{\Sigma}_\Gamma\mathbf{V}^T \simeq \mathbf{X}(\mathbf{Y}^t)^T$. In the trace form, this approximation results in the lower bound inequality $-Tr\{\mathbf{\Sigma}_A\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_\Gamma\} \le -Tr\{\mathbf{\Sigma}_A\mathbf{\Sigma}^{-1}\mathbf{T}\}$ and ensures that $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t) \le g(\mathbf{A}^{t+1},\mathbf{Y}^t)$. The proof is given in [20], *Appendix A* together with the proof about the existence of a closed form solution in the form of (4) with a solution for $\mathbf{\Sigma}_A$ without the constraints in (5).

The constraint (5) is important in order to guarantee the preservation of the gradient $\frac{\partial g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)}{\partial \mathbf{A}^{t+1}}$ of the approximative objective $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)$ w.r.t. gradient $\frac{\partial g(\mathbf{A}^{t+1},\mathbf{Y}^t)}{\partial \mathbf{A}^{t+1}}$ of the original objective $g(\mathbf{A}^{t+1},\mathbf{Y}^t)$.

We assume that the solutions $\mathbf{A}^t$ and $\mathbf{Y}^t$, and the gradient $\frac{\partial g_\epsilon(\mathbf{A}^t,\mathbf{Y}^t)}{\partial \mathbf{A}}$ at iterations t are known. In order to preserve the gradient of the approximative objective $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)$ the solution for $\mathbf{A}^{t+1}$ should be estimated such that it holds:

$$Tr\{\left(\frac{\partial g(\mathbf{A}^t,\mathbf{Y}^t)}{\partial \mathbf{A}^t}\right)^T(\mathbf{A}^t - \beta\mathbf{A}^{t+1})\} \le 0, \tag{6}$$

where $\mathbf{A}^t - \beta\mathbf{A}^{t+1}$ is a descend direction only if (6) holds true and $\beta \ge 0$. We denote $\mathbf{C} = \left(\frac{\partial g(\mathbf{A}^t,\mathbf{Y}^t)}{\partial \mathbf{A}^t}\right)^T\mathbf{A}^t$ and by using (4), we denote $\mathbf{F} = \mathbf{\Sigma}^{-1}\mathbf{U}^T\left(\frac{\partial g(\mathbf{A}^t,\mathbf{Y}^t)}{\partial \mathbf{A}^t}\right)^T\mathbf{V}\mathbf{S}^T$. In order to simplify, we use $\mathbf{C}$ and $\mathbf{F}$ and express the left hand side of (6) as $Tr\{\mathbf{C} - \beta\mathbf{F}\mathbf{\Sigma}_A\}$. By using the smallest singular value λ_{min} of the matrix $\mathbf{F}$ we have the following bound:

$$Tr\{\mathbf{C} - \beta\lambda_{min}\mathbf{\Sigma}_A\} \ge Tr\{\mathbf{C} - \beta\mathbf{F}\mathbf{\Sigma}_A\}, \tag{7}$$

which represents an upper bound on the condition for the preservation of the gradient (6). When we reorder the left hand side of (7), impose that the upper bound is nonnegative from above, *i.e.*, $0 \ge Tr\{\mathbf{C} - \beta\lambda_{min}\mathbf{\Sigma}_A\}$, and consider a more strict, element-wise condition then we arrive at:

$$\sigma_A(n) \ge \frac{1}{\beta\lambda_{min}}\sigma_C(n), \tag{8}$$

where $\sigma_C(n) = \Sigma_C(n,n)$ and $\mathbf{\Sigma}_C$ is diagonal matrix with diagonal elements equal to the singular values of $\mathbf{C}$ □

– *Trade-Off Gradient Alignment and Lower Bound Tightness* We use bounds in the form of:

$$\begin{aligned} -Tr\{\mathbf{\Sigma}_A\mathbf{\Sigma}^{-1}\mathbf{\Sigma}_\Gamma\} &\le -Tr\{\mathbf{\Sigma}_A\mathbf{\Sigma}^{-1}\mathbf{T}\} \\ 0 \ge Tr\{\mathbf{C} - \beta\mathbf{\Sigma}_F\mathbf{\Sigma}_A\} &\ge Tr\{\mathbf{C} - \beta\mathbf{F}\mathbf{\Sigma}_A\} \end{aligned} \tag{9}$$

where $\Sigma_F(n,n) = \lambda_{min}, \forall n \in \mathcal{N}$. The first bound is related to the approximated objective $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)$, that is, its left

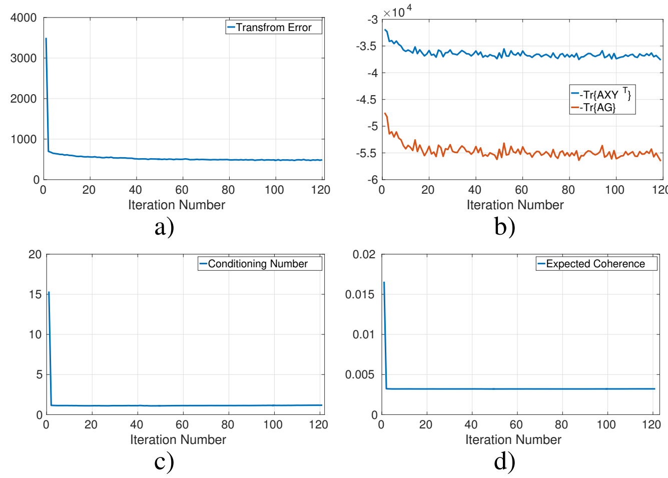

Fig. 1. The evolution of a) the transform error $\|\mathbf{AX}-\mathbf{Y}\|_F^2$, b) the $-Tr\{\mathbf{AXY}^T\}$ and its lower bound approximation $-Tr\{\mathbf{AG}\}$ c) the conditioning number and d) the expected mutual coherence $\mu(\mathbf{A})$ while learning the transform matrix $\mathbf{A}$ on overlapping 8×8 noisy image blocks (equivalently $N=64$) from the Cameraman image, where M was set to 80 and the sparsity level was set to 36.

hand side is related to $\mathbf{G}$ that appears in $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)$ and the second bound is related to the constraint about the the preservation of the gradient. The bounds (9) address a trade-off between how much are aligned the gradients $\frac{\partial g(\mathbf{A}^{t+1},\mathbf{Y}^t)}{\partial \mathbf{A}^{t+1}}$ and $\frac{\partial g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)}{\partial \mathbf{A}^{t+1}}$ of the approximative objective and the original objective, and how much the lower bound $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)$ is close to the objective $g(\mathbf{A}^{t+1},\mathbf{Y}^t)$.

The bounds (9) offer three advantages in the proposed solution. First, their use results in approximate closed form solution expressed by (4) with (5). Second, they allow to estimate $\mathbf{A}^{t+1}$ that can lead to accelerated convergence, that is they allow to estimate a descend direction $\mathbf{A}^t-\beta\mathbf{A}^{t+1}$ such that the original objective $g(\mathbf{A}^{t+1},\mathbf{Y}^t)$ is rapidly decreased. Third, the bounds (9) enables us to estimate $\mathbf{A}^{t+1}$ that can lead to satisfactory solution under small amount of data, where the key is again the trade-off between the lower bound approximation and the alignment of the gradient that is addressed with (9).

While (9) only describes the trade-off, its limits and its optimal characterization w.r.t. the acceleration and the required minimum amount of data for acceptable solution are out of the scope of this paper. Nonetheless, using the proposed bounds (9), we empirically demonstrate that indeed the proposed solution for an image denoising application exhibits the aforementioned advantages.

Sparse coding Given $\mathbf{A}^{t+1}$, for $\forall \mathbf{x}_l, l\in\mathcal{L}$, the sparse coding problem is formulated as follows:

$$\mathbf{y}_l^{t+1}=\underset{\mathbf{y}_l}{argmin}\,\|\mathbf{A}^{t+1}\mathbf{x}_l-\mathbf{y}_l\|_2^2,\ \text{subject to}\ \|\mathbf{y}_l\|_0\le s, \quad (10)$$

where we use the global optimal solution as proposed in [6].

B. Local Convergence of the Algorithm

Since in the transform update, the approximate closed form solution preserves the gradient and in the sparse coding step,

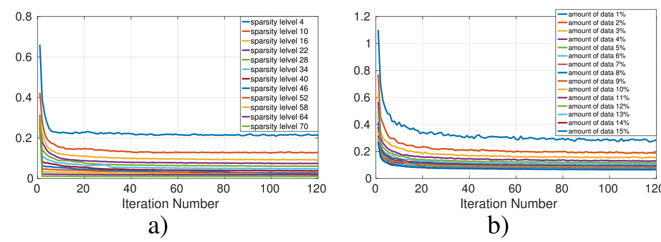

Fig. 2. The evolution of the normalized transform error $\frac{\|\mathbf{AX}-\mathbf{Y}\|_F^2}{L}$, where L is the total number of samples $\mathbf{x}_l\in\{1,...,L\}$ under a) sparsity levels $s\in\{4,10,16,22,28,24,40,46,52,58,64,70\}$ and b) amounts of data expressed in percentage from the total amount of data while learning the transform matrix $\mathbf{A}$ on overlapping 8×8 noisy image blocks (equivalently $N=64$) from the Cameraman image, where M was set to 80 and the sparsity level was set to 36.

we use exact closed form solution, the following allows us to state and prove a local convergence result.

Theorem 1 *Given data* $\mathbf{X}$ *and a pair of initial transform and sparse data* $\{\mathbf{A}^0,\mathbf{Y}^0\}$, *let* $\{\mathbf{A}^t,\mathbf{Y}^t\}$ *denote the iterative sequence generated by the solution* (4) *with* (5) *and the closed form solution of* (10). *Then, the sequence of the objective function values* $g(\mathbf{A}^t,\mathbf{Y}^t)$ *is a monotone decreasing sequence, satisfying* $g(\mathbf{A}^{t+1},\mathbf{Y}^{t+1})\le g(\mathbf{A}^{t+1},\mathbf{Y}^t)\le g(\mathbf{A}^t,\mathbf{Y}^t)$, *and converges to a finite value denoted as* g^*.

Proof: In the transform update step, $\mathbf{Y}^t$ is fixed and approximative minimizer is obtained w.r.t. $\mathbf{A}^{t+1}$, with $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)\le g(\mathbf{A}^{t+1},\mathbf{Y}^t)$. Therefore $g_\epsilon(\mathbf{A}^{t+1},\mathbf{Y}^t)\le g(\mathbf{A}^{t+1},\mathbf{Y}^t)\le g(\mathbf{A}^t,\mathbf{Y}^t)$. In the sparse coding step an exact solution is obtained for $\mathbf{Y}^{t+1}$ with fixed $\mathbf{A}^{t+1}$. Therefore, $g(\mathbf{A}^{t+1},\mathbf{Y}^{t+1})\le g(\mathbf{A}^{t+1},\mathbf{Y}^t)$, holds trivially. Combining the results for the two steps, we have $g(\mathbf{A}^{t+1},\mathbf{Y}^{t+1})\le g(\mathbf{A}^t,\mathbf{Y}^t)$ for any t. Since the function $g(\mathbf{A}^t,\mathbf{Y}^t)$ is lower bounded [3], the sequence of objective function values $\{g(\mathbf{A}^t,\mathbf{Y}^t)\}$ is a monotone decreasing and lower bounded, therefore it converges □

Since we use ϵ-**C**lose **A**pproximative solution in the **T**ransform estimate step we named our algorithm as ϵCAT.

C. Image Denoising With ϵCAT

A $S_1\times S_2$ noisy image, represented as a vector is denoted as $\mathbf{q}=\mathbf{x}+\mathbf{g}\in\Re^{S_1S_2}$, where $\mathbf{g}$ and $\mathbf{x}$ are the noise and the original image, respectively. The noisy image block is denoted as $\mathbf{q}_i=\mathbf{E}_i\mathbf{q}\in\Re^N$, $\forall i\in\mathcal{I}$, where the matrix $\mathbf{E}_i\in\{0,1\}^{N\times S_1S_2}$ is used to extract noisy image block at location i and $\mathcal{I}$ is the index set of all image block locations.

The extension of (1) for block level image denoising is formulated as:

$$\min_{\mathbf{x}_i,\boldsymbol{\alpha}_i,\mathbf{A}}\sum_{i=1}^{L}\|\mathbf{Ax}_i-\mathbf{y}_i\|_2^2+\tau\|\mathbf{x}_i-\mathbf{q}_i\|_2^2+\Omega_1(\mathbf{A}) \quad (11)$$
$$\text{subject to}\ \|\mathbf{y}_i\|_0\le s_i, i\in\mathcal{I},$$

where $\mathbf{A}\in\Re^{M\times N}$ is the transform matrix, $\mathbf{x}_i\in\Re^N$ is the estimated original image block, $\mathbf{y}_i\in\Re^M$ is the sparse transform representation with sparsity level s_i and τ is a parameter inversely propositional to the noise variance σ^2.

Note that by using (4), the pseudo-inverse of $\mathbf{A}$ exists as $\mathbf{A}^\dagger=\mathbf{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}_A^{-1}\mathbf{S}\mathbf{V}^T$. Furthermore, given $\mathbf{A}^\dagger$ and $\mathbf{y}_i$,

	TL-S	TL-O	K-SVD	ϵCAT	FRIST	OCTOBOS	BM3D
$l_{data}[\%]$	$25-100$	$25-100$	100	$\mathbf{3-15}$	100	100	100
$t_e[min]$	4.6	2.9	9.8	**1.24**	3.1	3.3	/

TABLE I
THE EXECUTION TIME IN MINUTES AND THE PERCENTAGE OF THE USED IMAGE DATA.

	σ	TL-S	TL-O	K-SVD	ϵCAT	FRIST	OCTOBOS	BM3D
Peppers	10	34.45	**34.49**	34.2	34.44	**34.68**	34.57	34.66
	20	29.98	30.60	**30.82**	30.63	31.02	30.97	**31.29**
Cameramen	10	33.93	33.83	33.72	**33.93**	34.16	34.15	**34.18**
	20	29.93	29.95	29.82	**30.12**	30.33	30.24	**30.48**
Barbara	10	34.45	34.55	34.42	**34.60**	34.57	34.64	**34.98**
	20	30.53	30.90	30.82	**30.91**	/	31.05	**31.78**
Lena	10	34.48	34.68	**35.28**	34.96	35.67	35.64	**35.93**
	20	32.12	32.00	32.43	**32.51**	/	32.59	**33.05**
Man	10	**32.96**	32.16	32.71	32.75	33.06	32.98	**33.98**
	20	**29.57**	28.63	29.40	29.41	29.76	29.74	**30.03**

TABLE II
DENOSING PERFORMANCE IN PSNR, WHERE σ IS THE NOISE STANDARD DEVIATION.

(11) approximately reduces to *constrained projection problem* $(P_D): \min_{\mathbf{x}_i} \|\mathbf{x}_i - \mathbf{A}^{\dagger}\boldsymbol{\alpha}_i\|_2^2 + \tau\|\mathbf{x}_i - \mathbf{q}_i\|_2^2$ for the variable $\mathbf{x}_i$, and its closed form solution for individual image block $\mathbf{x}_i$ can be computed as $\mathbf{x}_i = \begin{bmatrix}\sqrt{\tau}\mathbf{I}\\ \mathbf{I}\end{bmatrix}^{\dagger}\begin{bmatrix}\sqrt{\tau}\mathbf{q}_i\\ \mathbf{A}^{\dagger}\mathbf{y}_i\end{bmatrix} = e_1\sqrt{\tau}\mathbf{q}_i + e_2\mathbf{A}^{\dagger}\mathbf{y}_i$, where $\begin{bmatrix}\sqrt{\tau}\mathbf{I}\\ \mathbf{I}\end{bmatrix}^{\dagger}$ is the pseudo-inverse of $\begin{bmatrix}\sqrt{\tau}\mathbf{I}\\ \mathbf{I}\end{bmatrix}$ (two concatenated diagonal matrices $\sqrt{\tau}\mathbf{I}$ and $\mathbf{I}$) and the solution is easily computed as $[e_1, e_2] = [\sqrt{\tau}, 1]^{\dagger}$[3]. The denoising problem (11) is non-convex in the variables $\mathbf{x}_i, \mathbf{y}_i$ and $\mathbf{A}$ together. Similarly to [6] and [10], we use an iterative procedure that has two steps. In the first step (*Transform estimate* update), $\mathbf{x}_i = \mathbf{E}_i\mathbf{q}$ is fixed, the initial sparsity is set to $s_i = s$ and the overcomplete transform matrix $\mathbf{A}$ is estimated using the proposed approximate closed from solution. In the second step (*Sparse coding* update), given $\mathbf{A}$, the remaining variables $\mathbf{y}_i$ and $\mathbf{x}_i$ are updated similarly as proposed in [6]. Commonly, a sparsity level s_i for the sparse code $\mathbf{y}_i$ is chosen such that the *denosing error* term $\|\mathbf{q}_i - \mathbf{x}_i\|_2^2$ is bounded from above by a constant. The usual bound is $\|\mathbf{q}_i - \mathbf{x}_i\|_2^2 \le CN\sigma^2$, where C is a constant, σ^2 is the noise variance, $\mathbf{x}_i = e_1\sqrt{\tau}\mathbf{q}_i + e_2\mathbf{A}^{\dagger}\mathbf{y}_i$ and $\mathbf{y}_i$ is estimated as a solution to $(P2)$. Here, instead, we upper bound just the inner product of the estimate $\mathbf{x}_i$, *i.e.*, $\mathbf{x}_i^T\mathbf{x}_i \le C_0CN\sigma^2$, where C_0 is an additional constant. The new estimates for the sparsity levels $s_i, \forall i$, are used in the next *Transform estimate* update, and the procedure is iterated between *Transform estimate* and *Sparse coding* updates until the predefined number of iterations is reached. In the final iteration, only these $\mathbf{x}_i$ that satisfy $\|\mathbf{q}_i - \mathbf{x}_i\|_2^2 \le CN\sigma^2$ are considered as the actual denoised image patches. Given the final estimates the denoised image $\mathbf{x}$ is obtained in the same fashion as in [10], [16] and [6].

IV. COMPUTER SIMULATION

This section validates the proposed approach by numerical experiments and demonstrates its advantages.

[3]Note that the coefficients $[e_1, e_2]$ have to be computed only once, stored and then reused in the later computations.

	$l_{data}[\%]$	1	2	3	5	7	10
Peppers	$t_e[min]$	0.17	0.22	0.25	0.37	0.45	0.70
	$PSNR$	31.5	33.8	34.1	34.3	**34.4**	**34.4**
Cameramen	$t_e[min]$	0.17	0.22	0.26	0.5	0.8	1.01
	$PSNR$	31.5	33.3	33.4	33.7	33.8	**33.9**
Barbara	$t_e[min]$	0.22	0.34	0.49	0.66	0.83	1.13
	$PSNR$	31.5	33.3	33.4	33.7	34.1	**34.3**
Lena	$t_e[min]$	0.25	0.40	0.61	0.70	0.86	1.14
	$PSNR$	31.5	33.3	33.4	34.4	34.4	**34.7**
Man	$t_e[min]$	0.27	0.41	0.61	0.73	0.87	1.17
	$PSNR$	30.0	31.8	31.9	32.5	32.6	**32.7**

TABLE III
THE PSNR FOR THE ϵCAT ALGORITHM LEARNED ON PERCENTAGE OF THE AVAILABLE NOISY IMAGE DATA WITH NOISE LEVEL $\sigma = 10$.

A. Data and Algorithm Inicalization

To evaluate the potential of the proposed approach we used the Peppers, Cameramen, Barbara, Lena and Man images at image resolution 256×256, 256×256, 512×512, 512×512 and 512×512, respectively. The following algorithm parameters are used $N = 64, M = 80, \lambda_1 = \lambda_2 = \lambda_3 = 10\times10^7$, $C = 1.08, C_0 = 1/2$ and $\tau = 0.01/\sigma$. The algorithm is initialized with a random matrix having i.i.d. Gaussian (zero mean, variance one) entries.

B. Denoising Setup

The denoising recovery performance is evaluated at noise levels $\sigma = 10$ and $\sigma = 20$ and the sparsity is set to 25 and 19, respectively. The transform is learned by executing 300 iterations. The results are obtained as average of 3 runs. We use not optimized Matlab implementation running on PC having Intel Xeon(R) 3.60GHz CPU and 32G RAM memory. For each of the noisy images a sparsifying transform matrix $\mathbf{A}$ is learned using only $1\% - 15\%$ of the total amount of its noisy patches. The result of ϵCAT is compared with the results of the algorithms proposed in [16] (TL-S), [6] (TL-O), [10] (K-SVD), [17] (OCTOBOS), [21] (FRIST) and [22] (BM3D).

C. Results

The results are shown in Figures 1 and 2, and Tables I, II and III. Our empirical validation suggested that in our algorithm the solution for $\mathbf{A}$ expressed by (4) with (5) when using the bounds (9), is equivalent to the solution (4) with (5), but without the constraint in (5). We noticed that the resulting $\sigma_A(n)$ are higher then $\frac{1}{\beta\lambda_{min}}\sigma_C(n)$, $\forall n \in \mathcal{N}$ which implies that the constraint is implicitly satisfied and the proposed solution without the same constraint in (5) preserves the gradient. Therefore, we present results using the solution for $\mathbf{A}$ that is without the explicit inequality constraint in (5).

In Figure 1 are shown the evolution of the transform error, the term $Tr\{\mathbf{AXY}^T\}$, its lower bound $Tr\{\mathbf{AG}\}$, the conditioning number and the expected mutual coherence $\mu(\mathbf{A})$ while learning the transform matrix $\mathbf{A}$ on Cameraman image. The transform error rapidly decreases, the lower bound $Tr\{\mathbf{AG}\}$ approximation is well below $Tr\{\mathbf{AXY}^T\}$ while the conditioning number and the expected mutual coherence $\mu(\mathbf{A})$ are decreased from initial values and remain low. This suggest that the proposed solution efficiently reduces the transform error while satisfying the regularization constraints on $\mathbf{A}$.

In Figure 2 a) are shown the evolution of the transform error across varying number of sparsity wile the dictionary size, the amount of data and the parameters $\{\lambda_2, \lambda_3, \lambda_4\}$ are fixed. The transform error is decreasing for all sparsity levels and for higher sparsity levels the rate of decrease on the transform error is faster. In the same figure under b) we show the evolution of the transform error across varying amount of data wile the sparsity level, the dictionary size, the amount of data and the parameters $\{\lambda_2, \lambda_3, \lambda_4\}$ are fixed. We see that the actual error is decreasing and the rate of decrease is increasing as we increase the amount of data and it saturates around 14% and 15%. This confirms that the proposed algorithm with the introduced update for $\mathbf{A}$ can attain satisfactory solution with low transform error while using a small amount of data.

Considering the results that are shown in Tables I and II only 15% of the total amount of available patches were used for learning the transform matrix of the ϵCAT algorithm, whereas the rest of the algorithms use $25\% - 100\%$. This reflects the resulting execution time that (as shown on Table I) is around $4\times$, $2\times$, $9\times$, $3\times$ and $3\times$ faster then TL-S, TL-O, K-SVD, FRIST and OCTOBOS, respectively.

Table II shows evaluation across different images and compares with several algorithm. The proposed algorithm has slightly better overall denoising results for the used noise levels $\sigma \in \{10, 20\}$ compared to the TL-S, TL-O and K-SVD algorithm. On the other hand w.r.t. the rest of the algorithms the results are competitive, but, have slightly lower PSNR. We explain this by the fact that in the FRIST, OCTOBOS and BM3D algorithms a flipping and rotation invariance, grouping and block similarity priors were used, respectively, but, in the current version of our algorithm these priors were not considered. Nonetheless, some of the benefits when using ϵCAT are notable in Table III. Even when using only $3\% - 7\%$ of the noisy image patches during learning there is no big degradation in the final results and they remain competitive compared to the TL-S, TL-O and K-SVD algorithm.

In summary, the results confirm that the main advantage of the current version of the proposed algorithm when using the bounds (9) for updating $\mathbf{A}$ by (4) with (5), but without the constraint in (5) is the implicit preservation of the gradient and the ability to rapidly decrease the transform error and thereby the objective per iteration. This results in fast convergence and small amount of data required to learn the model parameters.

V. Conclusion

This paper considered the transform model, an alternating algorithm and presented analysis around the transform update step. We proposed iterative algorithm for transform learning with two alternating steps having exact and approximate closed form solutions. The preliminary results demonstrated promising performance on the image examples provided in this paper. Performance evaluation on more extensive image collection, together with extensions considering other image priors is left for future work.

References

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

[2] R. Rubinstein, T. Peleg, and M. Elad, "Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model," *Signal Processing, IEEE Transactions*, vol. 61, no. 3, pp. 661–677, January 2013.

[3] S. Ravishankar and Y. Bresler, "ℓ_0 sparsifying transform learning with efficient optimal updates and convergence guarantees," *CoRR*, vol. abs/1501.02859, 2015.

[4] S. Hawe, M. Kleinsteuber, and K. Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Trans. Image Processing*, vol. 22, no. 6, pp. 2138–2150, 2013.

[5] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms," *CoRR*, vol. abs/1106.4987, 2011.

[6] S. Ravishankar and Y. Bresler, "Learning overcomplete sparsifying transforms for signal processing," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 2013, pp. 3088–3092.

[7] S.G. Mallat and Zhifeng Z., "Matching pursuits with time-frequency dictionaries," *Trans. Sig. Proc.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[8] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing: Closing the gap between performance and complexity," *CoRR*, vol. abs/0803.0811, 2008.

[9] I.F. Gorodnitsky and B.D. Rao, "Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm," *Trans. Sig. Proc.*, vol. 45, no. 3, pp. 600–616, Mar. 1997.

[10] M. Aharon, M. Elad, and A. Bruckstein, "Svdd: An algorithm for designing overcomplete dictionaries for sparse representation," *Trans. Sig. Proc.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

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

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

[13] K. Schnass, "Convergence radius and sample complexity of ITKM algorithms for dictionary learning," *CoRR*, vol. abs/1503.07027, 2015.

[14] J. Dong, W. Wang, W. Dai, M. D. Plumbley, Zi-Fa Han, and J. A. Chambers, "Analysis simco algorithms for sparse analysis model based dictionary learning," *IEEE Trans. Signal Processing*, vol. 64, no. 2, pp. 417–431, 2016.

[15] S. Ravishankar and Y. Bresler, "Closed-form solutions within sparsifying transform learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, 2013, pp. 5378–5382.

[16] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms for image processing," in *19th IEEE International Conference on Image Processing, ICIP 2012, Lake Buena Vista, Orlando, FL, USA, September 30 - October 3, 2012*, 2012, pp. 681–684.

[17] B. Wen, S. Ravishankar, and Y. Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *International Journal of Computer Vision*, vol. 114, no. 2-3, pp. 137–167, 2015.

[18] B. Wen, S. Ravishankar, and Y. Bresler, "Learning overcomplete sparsifying transforms with block cosparsity," in *2014 IEEE International Conference on Image Processing, ICIP 2014, Paris, France, October 27-30, 2014*, 2014, pp. 803–807.

[19] S. Ravishankar and Y. Bresler, "Doubly sparse transform learning with convergence guarantees," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, 2014, pp. 5262–5266.

[20] D. Kostadinov and S. Voloshynovskiy, "Learning non-linear transform with discriminative and minimum information loss priors," 2018, openreview.net, open acess at https://openreview.net/pdf?id=SJzmJEq6W.

[21] Y. Bresler B. Wen and S. Ravishankar, "FRIST - flipping and rotation invariant sparsifying transform learning and applications," *CoRR*, vol. abs/1511.06359, 2015.

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