# Convolutional Dictionary Learning for Multi-Channel Signals

Cristina Garcia-Cardona
CCS-3, CCS Division
Los Alamos National Laboratory
Los Alamos, NM 87544, USA
cgarciac@lanl.gov

Brendt Wohlberg
T-5, Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545, USA
brendt@ieee.org

***Abstract*—There has recently been a rapid growth in interest in the design of efficient algorithms for convolutional sparse coding, and in the application of these methods to signal and image processing inverse problems. Thus far, however, the design of algorithms and methods for multi-channel signals has received very little attention. In this work we extend our initial results in convolutional sparse coding and dictionary learning for this type of data, proposing new algorithms that scale well to signals with large numbers of channels, and demonstrate their performance in an application involving hyperspectral imagery.**

## I. Introduction

Convolutional sparse representations [1], also known as translation-invariant sparse representations [2], are a form of sparse representation with a structured dictionary that can be applied to an entire signal or image, providing a convenient alternative to the usual approach of independently applying sparse representations to relatively small signal or image regions [3]. Despite the rapidly growing research literature on this technique, their application to multi-channel signals and images has received very little attention. The primary goal of the present paper is to extend previous work on applying these methods to color imagery [4], [5] to multi- and hyper-spectral imagery, involving many more than three channels.

### A. Convolutional Sparse Coding

The convolutional sparse coding (CSC) problem is usually posed as

$$\arg\min_{\{\mathbf{x}_m\}} \frac{1}{2}\Big\|\sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s}\Big\|_2^2 + \lambda \sum_m \|\mathbf{x}_m\|_1 \ , \tag{1}$$

where $\{\mathbf{d}_m\}$ is a set of $M$ linear filters that constitute the dictionary, $\{\mathbf{x}_m\}$ is a corresponding set of $M$ coefficient maps that constitute the convolutional sparse representation, and $\mathbf{s}$ is the single-channel image to be sparse coded, represented as an $N$-vector, where $N$ is the number of pixels in the image. If we define matrices $D_m$ such that $D_m\mathbf{x}_m = \mathbf{d}_m * \mathbf{x}_m$, and block matrices

$$D = \begin{pmatrix} D_0 & D_1 & \dots \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \end{pmatrix} \ , \tag{2}$$

we can rewrite (1) in the more convenient matrix-vector product form

$$\arg\min_{\mathbf{x}} \ (1/2)\, \|D\mathbf{x} - \mathbf{s}\|_2^2 + \lambda \|\mathbf{x}\|_1 \ . \tag{3}$$

Algorithms for solving this problem are discussed in detail in [3]. The two leading approaches are based on the the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) [6], [7], [3] and on the Alternating Direction Method of Multipliers (ADMM) [8], [9], [10]. In the case of FISTA, the primary computational cost is in calculating the gradient

$$\nabla_{\mathbf{x}} \ (1/2)\, \|D\mathbf{x} - \mathbf{s}\|_2^2 = D^T(D\mathbf{x} - \mathbf{s}) \ , \tag{4}$$

which is often more efficient to compute in the frequency domain [3, Sec. IV.B]. The main computational cost of the ADMM method is in solving a subproblem of the form

$$\arg\min_{\mathbf{x}} (1/2)\big\|D\mathbf{x} - \mathbf{s}\big\|_2^2 + (\rho/2)\, \|\mathbf{x} - \mathbf{z}\|_2^2 \ . \tag{5}$$

This subproblem has a closed-form solution given by

$$(D^T D + \rho I)\mathbf{x} = D^T\mathbf{s} + \rho\mathbf{z} \ . \tag{6}$$

The matrix $D^T D$ is too large for solution via conventional methods, but it can be solved efficiently by exploiting the diagonalization of the $D_m$ matrices in the frequency domain [9], [10].

### B. Convolutional Dictionary Learning

The convolutional dictionary learning (CDL) problem is usually posed as

$$\arg\min_{\{\mathbf{d}_m\},\{\mathbf{x}_{m,k}\}} \frac{1}{2}\sum_k \Big\|\sum_m \mathbf{d}_m * \mathbf{x}_{m,k} - \mathbf{s}_k\Big\|_2^2 + \lambda \sum_{m,k} \|\mathbf{x}_{m,k}\|_1 \quad \text{such that } \ \|\mathbf{d}_m\|_2 = 1 \ \forall m \ , \tag{7}$$

$\mathbf{s}_k$ is the $k^{\text{th}}$ of $K$ training images and $\mathbf{x}_{m,k}$ are the corresponding coefficient maps.

Algorithms for solving this problem are discussed in detail in [5]. These algorithms consist of an alternation between a sparse coding stage, solving for $\mathbf{x}_{m,k}$ for fixed $\mathbf{d}_m$, and a dictionary update stage that solves for $\mathbf{d}_m$ with fixed $\mathbf{x}_{m,k}$. As in the case of the sparse coding problem, the most efficient solutions for the dictionary update problem are again based on the FISTA and ADMM frameworks, but due to the different algebraic structure of the dictionary update, the linear solve required by the ADMM approach is inherently more computationally expensive than the corresponding linear solve in the ADMM algorithm for sparse coding [5, Sec. II.B].

## II. Multi-Channel CSC

Thus far, the convolutional sparse coding of multi-channel signals has received only limited attention [11], [4], [5], [12], and we are not aware of any existing work that considers the application of convolutional sparse representations to multi-channel signals with many more than three channels, such as multi- and hyper-spectral imagery. There is a wide variety of different approaches to constructing a convolutional sparse representation of a multi-channel signal:

1) Multi-channel dictionary and single channel representation
2) Single channel dictionary and multi-channel representation
   a) Dictionary:
      i) Same dictionary filters for each channel
      ii) Different dictionary filters for each channel
      iii) Product of convolutional and channel dictionaries
   b) Representation:

i) Completely independent representation for each channel
ii) Distinct representation for each channel, but with a coupling via an $\ell_{2,1}$ norm regularization term

3) A combination of the above approaches applied to disctinct sets of channels

Options 1, 2(a)i+2(b)i, and 2(a)i+2(b)ii were considered in [4], [5], and methods for options 2(a)iii+2(b)i and 2(a)iii+2(b)ii will be presented here. We do not consider any of the combinations of options involving 2(a)ii or 3, but note that they do not involve solving any fundamentally different optimization problems, and can be solved via the same methods that are applicable to the other options.

To simplify notation, we will only consider the sparse coding of a single image to avoid the need to introduce an initial index over distinct images. While the extension to multiple images complicates the notation, it is computationally straightforward since the sparse coding problems are independent across the distinct input images. Multi-channel input images will be denoted by a set of C individual channels, $\{\mathbf{s}_c\}$, or by block matrix $S = \begin{pmatrix} \mathbf{s}_0 & \mathbf{s}_1 & \dots \end{pmatrix}$.

### A. Multi-Channel Dictionary

In this case the dictionary consists of a set of multi-channel filters $\{\mathbf{d}_{c,m}\}$ where $c$ indexes the $C$ distinct channels and $m$ indexes the $M$ distinct atoms in the dictionary, and the representation $\{\mathbf{x}_m\}$ consists of a single channel of $M$ coefficient maps. The CSC problem can be posed as

$$\arg\min_{\{\mathbf{x}_m\}} \frac{1}{2} \sum_c \left\| \sum_m \mathbf{d}_{c,m} * \mathbf{x}_m - \mathbf{s}_c \right\|_2^2 + \lambda \sum_m \|\mathbf{x}_m\|_1 \ , \quad (8)$$

or in in matrix-vector form

$$\arg\min_X (1/2) \|DX - S\|_F^2 + \lambda \|X\|_1 \ , \quad (9)$$

where $D_{c,m}$ is defined such that $D_{c,m}\mathbf{x}_m = \mathbf{d}_{c,m} * \mathbf{x}_m$, and

$$D = \begin{pmatrix} D_{0,0} & D_{0,1} & \dots \\ D_{1,0} & D_{1,1} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad X = \begin{pmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \\ \vdots \end{pmatrix} \ . \quad (10)$$

*1) Algorithms:* Algorithms for solving this problem are discussed in [4][5, Sec. VI].

### B. Single Channel Dictionary

In this case the dictionary consists of a set of $M$ single channel filters $\{\mathbf{d}_m\}$, and the representation $\{\mathbf{x}_{m,c}\}$ consists of a set of $C$ distinct channel coefficient maps for each of the $M$ dictionary atoms. The CSC problem can be posed as

$$\arg\min_{\{\mathbf{x}_{m,c}\}} \frac{1}{2} \sum_c \left\| \sum_m \mathbf{d}_m * \mathbf{x}_{m,c} - \mathbf{s}_c \right\|_2^2 + \lambda \sum_{m,c} \|\mathbf{x}_{m,c}\|_1 \ , \quad (11)$$

or in matrix-vector form

$$\arg\min_X (1/2) \|DX - S\|_F^2 + \lambda \|X\|_1 \ , \quad (12)$$

where $D_m$ is defined such that $D_m\mathbf{x}_{m,c} = \mathbf{d}_m * \mathbf{x}_{m,c}$, and

$$D = \begin{pmatrix} D_0 & D_1 & \dots \end{pmatrix} \quad X = \begin{pmatrix} \mathbf{x}_{0,0} & \mathbf{x}_{0,1} & \\ \mathbf{x}_{1,0} & \mathbf{x}_{1,1} & \\ \vdots & \vdots & \ddots \end{pmatrix} \ . \quad (13)$$

The distinct representations for the channels can be coupled by replacing the $\ell_1$ norm with an $\ell_{2,1}$ norm, with the $\ell_2$ component taken along the channel index $c$.

*1) Algorithms:* Since the channels are independent, or coupled only via an $\ell_{2,1}$ regularization term, this problem can be solved via straightforward extension of algorithms for the single-channel case.

### C. Product Dictionary

In this case the dictionary consists of a set of $M_D$ single channel filters $\{\mathbf{d}_m\}$, as well as a $C \times M_B$ dictionary matrix $B$ that represents the cross-channel properties of the signal, and the representation consists of a set of coefficient maps, $\{\mathbf{x}_{m,c}\}$, where $m \in \{0, 1, \dots, M_D - 1\}$ indicates the relevant filter in dictionary $\{\mathbf{d}_m\}$ and $c \in \{0, 1, \dots, M_B - 1\}$ indicates the relevant column of dictionary $B$. The CSC problem can be posed as

$$\arg\min_{\{\mathbf{x}_{m,c}\}} \frac{1}{2} \left\| B \begin{pmatrix} (\sum_m \mathbf{d}_m * \mathbf{x}_{m,0})^T \\ (\sum_m \mathbf{d}_m * \mathbf{x}_{m,1})^T \\ \vdots \end{pmatrix} - \begin{pmatrix} \mathbf{s}_0^T \\ \mathbf{s}_1^T \\ \vdots \end{pmatrix} \right\|_2^2 + \lambda \sum_{m,c} \|\mathbf{x}_{m,c}\|_1 \ , \quad (14)$$

or in in matrix-vector form

$$\arg\min_X (1/2) \left\| DXB^T - S \right\|_F^2 + \lambda \|X\|_1 \ , \quad (15)$$

where $D_m$ is defined such that $D_m\mathbf{x}_{m,c} = \mathbf{d}_m * \mathbf{x}_{m,c}$, and

$$D = \begin{pmatrix} D_0 & D_1 & \dots \end{pmatrix} \quad X = \begin{pmatrix} \mathbf{x}_{0,0} & \mathbf{x}_{0,1} & \\ \mathbf{x}_{1,0} & \mathbf{x}_{1,1} & \\ \vdots & \vdots & \ddots \end{pmatrix} \ . \quad (16)$$

While this type of composite dictionary has not previously been considered for convolutional sparse representations, it should be noted that it is closely related to the *t-product* tensor decomposition [13], [14], [15].

*1) FISTA Algorithm:* Solution of this problem via FISTA requires calculating the gradient

$$\nabla_X (1/2) \left\| DXB^T - S \right\|_F^2 = D^T (DXB^T - S)B \ . \quad (17)$$

Denoting the DFT transform by $F$, and defining $\hat{D} = FDF^{-1}$, $\hat{X} = FX$ and $\hat{S} = FS$, this gradient can be computed in the frequency domain as

$$\nabla_X \frac{1}{2} \left\| DXB^T - S \right\|_F^2 = F^{-1} (\hat{D}^H \hat{D} \hat{X} B^T B - \hat{D}^H \hat{S} B) \ . \quad (18)$$

*2) ADMM Algorithm:* Solution of this problem via ADMM involves solving a subproblem of the form

$$\arg\min_X (1/2) \left\| DXB^T - S \right\|_F^2 + (\rho/2) \|X - Z\|_F^2 \ . \quad (19)$$

By reuse of the notation of Sec. II-C1, and defining $\hat{Z} = FZ$, we can write this as

$$\arg\min_{\hat{X}} (1/2) \left\| \hat{D} \hat{X} B^T - \hat{S} \right\|_2^2 + (\rho/2) \left\| \hat{X} - \hat{Z} \right\|_2^2 , \quad (20)$$

the solution of which is given by the linear equation

$$\hat{D}^H \hat{D} \hat{X} B^T B + \rho \hat{X} = \hat{D}^H \hat{S} B + \rho \hat{Z} \ . \quad (21)$$

Since $B^T B$ is a normal matrix it has an eigenvector decomposition $B^T B = Q \Omega Q^T$ where $Q$ is an orthogonal matrix (i.e. $Q^T Q = Q Q^T = I$) and $\Omega$ is diagonal. We therefore have

$$\begin{aligned} \hat{D}^H \hat{D} \hat{X} Q \Omega Q^T + \rho \hat{X} &= \hat{D}^H \hat{S} B + \rho \hat{Z} \\ \hat{D}^H \hat{D} \hat{X} Q \Omega Q^T Q + \rho \hat{X} Q &= \hat{D}^H \hat{S} B Q + \rho \hat{Z} Q \\ \hat{D}^H \hat{D} \hat{X} Q \Omega + \rho \hat{X} Q &= \hat{D}^H \hat{S} B Q + \rho \hat{Z} Q \ , \end{aligned} \quad (22)$$

and by defining $\tilde{X} = \hat{X}Q$, we have

$$\hat{D}^H \hat{D} \tilde{X} \Omega + \rho \tilde{X} = \hat{D}^H \hat{S} B Q + \rho \hat{Z} Q \;. \tag{23}$$

Although this is a variant of a Sylvester equation [16, Sec. 13.3], we cannot apply standard techniques for that class of equations due to the very large size of $\hat{D}^H \hat{D}$. However, since [17, Sec. 2.3][16, Ch. 13]

$$\mathrm{vec}(AXB) = (B^T \otimes A)\,\mathrm{vec}(X) = \mathrm{vec}(C) \;, \tag{24}$$

we can express this problem in the form

$$(\Omega \otimes \hat{D}^H \hat{D} + \rho I_{BD})\,\mathrm{vec}(\tilde{X}) = ((Q^T B^T) \otimes \hat{D}^H)\,\mathrm{vec}(\hat{S}) + \rho\,\mathrm{vec}(\hat{Z}Q) \;, \tag{25}$$

where $I_{BD}$ is an $M_B M_D N \times M_B M_D N$ identity matrix. Since $\Omega$ is diagonal, we have

$$\Omega \otimes \hat{D}^H \hat{D} + \rho I_{BD} = \begin{pmatrix} \omega_0 \hat{D}^H \hat{D} + \rho I_D & 0 & \cdots \\ 0 & \omega_1 \hat{D}^H \hat{D} + \rho I_D & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \;,$$

where

$$\Omega = \begin{pmatrix} \omega_0 & 0 & \cdots \\ 0 & \omega_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad \Omega \in \mathbb{R}^{M_B \times M_B} \;, \tag{26}$$

and $I_D$ is an $M_D N \times M_D N$ identity matrix. Since $\Omega \otimes \hat{D}^H \hat{D} + \rho I_{BD}$ is block diagonal, we can independently solve each block of (25) using the efficient approach that exploits the Sherman-Morrison formula [10].

## III. Multi-Channel CDL

As in the case of the single channel CDL discussed in Sec. I-B, multi-channel CDL algorithms iteratively alternate between sparse coding and dictionary update stages. Simultaneous sparse coding of an entire set of training images is required, but since the sparse coding problem for multiple images is entirely decoupled across the individual images, it is straightforward to extend the single-image sparse coding algorithms discussed in Sec. II as required. We therefore focus on the dictionary update stage in this section.

### A. Multi-Channel Dictionary

The reader is referred to [5, Sec. VI] for a detailed description of the multi-channel dictionary update problem.

### B. Single Channel Dictionary

If a distinct single channel dictionary is to be learned for each channel of the signal (option 2(a)ii+2(b)i in Sec. II), then the dictionary learning problem decomposes into a set of independent single channel dictionary learning problems. If the same single channel dictionary is to be learned for all channels of the signal (option 2(a)i+2(b)i in Sec. II), then the dictionary learning problem decomposes into a single channel dictionary learning problem with all channels of the training data treated as distinct single channel training signals (i.e. a training data set consisting of $K$ images each with $C$ channels is treated as $CK$ single channel training images).

### C. Product Dictionary

In this case, re-using the notation of Sec. II-C, we can write the dictionary learning problem as

$$\begin{aligned} \underset{\{X_k\}, D, B}{\arg\min} \; & \frac{1}{2} \sum_k \left\| D X_k B^T - S_k \right\|_F^2 + \lambda \sum_k \|X_k\|_1 \\ & \text{s.t. } D \in \mathcal{C}_D \;, B \in \mathcal{C}_B \;, \end{aligned} \tag{27}$$

where $S_k$ is the $k^{\text{th}}$ of $K$ training images, $\mathcal{C}_D$ is the constraint set expressing the required normalization and support projection for $D$ (see [5, Sec. II.B]), and $\mathcal{C}_B$ is the set of matrices with unit-norm columns. Since the sparse coding problem for the entire training set decomposes into $K$ independent single-image sparse coding problems, we can solve it using the approach described in Sec. II-C. The update for each of the two dictionaries is computed by applying standard methods to the problems that remain after taking the product of the sparse representation and the other dictionary. Algorithms for the convolutional dictionary update are described in detail in [5, Sec. III], and the non-convolutional dictionary update is easily solved via simplified variants of the same algorithms[1]. The full dictionary learning algorithm is summarized in Algorithm 1. An alternative version with better convergence behavior but higher computational cost can be constructed by inserting an additional CSC update between the $D$ and $B$ dictionary updates.

**Data:** Training image set $\{S_k\}$
Initialize dictionaries $D^{(0)}$ and $B^{(0)}$;
Initialize iteration counter $i = 0$;
**while** *termination criteria not satisfied* **do**
  Apply product dictionary CSC (see in Sec. II-C) to compute $\{X_k\}^{(i+1)}$ for dictionaries $D^{(i)}$ and $B^{(i)}$, and training images $\{S_k\}$;
  Set $\mathring{X}_k = X_k^{(i+1)} (B^{(i)})^T$;
  Apply a convolutional dictionary update (see [5, Sec. III]) to compute $D^{(i+1)}$ for representation $\{\mathring{X}_k\}$ and training images $\{S_k\}$;
  Set $\mathring{X}_k = D^{(i+1)} X_k^{(i+1)}$;
  Apply a non-convolutional dictionary update to compute $B^{(i+1)}$ for representation $\{\mathring{X}_k\}$ and training images $\{S_k\}$;
  Increment iteration counter $i$;
**end**

**Algorithm 1:** Multi-channel CDL with product dictionary.

### D. Scaling with Number of Channels

In [5, Sec. VII.G] we estimated the scaling of CDL algorithm parameters with a change in the number of training images, $K$, by considering the simplified case in which this scaling is achieved by replicating the same training image. Here we estimate the corresponding algorithm parameter scaling behavior with the number of channels, $C$, in the simplified case in which the scaling is achieved by replicating channels. In Table I these properties are reported separately for parameters of the sparse coding (CSC) and for the dictionary updates (CDU) based on ADMM with an equality constraint and on FISTA.

## IV. Extended CSC Problem

We will compare the performance of the different approaches to convolutional sparse representation of multi-channel signals in the

[1] Such an algorithm is implemented as the `CnstrMOD` solver class included in the SPORCO software library [18]

TABLE I
SCALING PROPERTIES OF THE ALGORITHM PARAMETERS WITH RESPECT TO NUMBER OF CHANNELS $C$.

| Step | Dict. Type | Method | Parameter | |
|---|---|---|---|---|
| | | | | $\lambda$ |
| **CSC** | Multi-C | ADMM | $\rho : \mathcal{O}(C)$ | $\mathcal{O}(C)$ |
| | | FISTA | $L_{\text{csc}} : \mathcal{O}(C)$ | |
| | Single | ADMM | $\rho : \mathcal{O}(1)$ | $\mathcal{O}(1)$ |
| | | FISTA | $L_{\text{csc}} : \mathcal{O}(1)$ | |
| | Product | ADMM | $\rho : \mathcal{O}(C)$ | $\mathcal{O}(M_B)$ |
| | | FISTA | $L_{\text{csc}} : \mathcal{O}(C)$ | |
| **CDU** | Multi-C | ADMM | $\sigma : \mathcal{O}(C)$ | — |
| | | FISTA | $L_{\text{cdl}} : \mathcal{O}(1)$ | |
| | Single | ADMM | $\sigma : \mathcal{O}(C)$ | — |
| | | FISTA | $L_{\text{cdl}} : \mathcal{O}(C)$ | |
| | Product | ADMM | $\sigma_D : \mathcal{O}(C)$<br>$\sigma_B : \mathcal{O}(1)$ | — |
| | | FISTA | $L_D : \mathcal{O}(C)$<br>$L_B : \mathcal{O}(1)$ | |

context of a salt & pepper noise restoration problem [19]. The most effective approach to solving this problem via convolutional sparse representations is via a CSC variant with an $\ell_1$ data fidelity term [19, Sec. 6] and an additional $\ell_2$ penalty on the gradient of one of the coefficient maps [19, Sec. 3–4].

### A. Single Channel

The single channel variant of this CSC problem can be written as

$$\begin{aligned}\arg\min_{\{\mathbf{x}_m\}} \left\| \sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s} \right\|_1 + \lambda \sum_m \alpha_m \left\| \mathbf{x}_m \right\|_1 + \\ \frac{\mu}{2} \sum_m \beta_m \left\| \sqrt{(\mathbf{g}_0 * \mathbf{x}_m)^2 + (\mathbf{g}_1 * \mathbf{x}_m)^2} \right\|_2^2 \;,\end{aligned} \tag{28}$$

where $\mathbf{g}_0$ and $\mathbf{g}_1$ are filters that compute the gradients along image rows and columns respectively. Defining linear operators $G_0$ and $G_1$ such that $G_l \mathbf{x}_m = \mathbf{g}_l * \mathbf{x}_m$, and

$$\Gamma_l = \begin{pmatrix} \sqrt{\beta_0} G_l & 0 & \cdots \\ 0 & \sqrt{\beta_1} G_l & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \;, \tag{29}$$

this problem can be expressed in block matrix notation as

$$\begin{aligned}\arg\min_{\mathbf{x}} \left\| D\mathbf{x} - \mathbf{s} \right\|_1 + \lambda \left\| \boldsymbol{\alpha} \odot \mathbf{x} \right\|_1 + \\ (\mu/2) \left\| \Gamma_0 \mathbf{x} \right\|_2^2 + (\mu/2) \left\| \Gamma_1 \mathbf{x} \right\|_2^2 \;.\end{aligned} \tag{30}$$

This problem can be written in a form suitable for solution via ADMM as

$$\begin{aligned}\arg\min_{\mathbf{x}, \mathbf{y}_0, \mathbf{y}_1} \left\| \mathbf{y}_0 \right\|_1 + \lambda \left\| \boldsymbol{\alpha} \odot \mathbf{y}_1 \right\|_1 + (\mu/2) \left\| \Gamma_0 \mathbf{x} \right\|_2^2 + (\mu/2) \left\| \Gamma_1 \mathbf{x} \right\|_2^2 \\ \text{s.t.} \quad \mathbf{y}_0 = D\mathbf{x} - \mathbf{s} \;, \quad \mathbf{y}_1 = \mathbf{x} \;.\end{aligned} \tag{31}$$

The main computational cost of the resulting ADMM algorithm is in solving an equation of the form

$$\frac{\rho}{2} \left\| D\mathbf{x} - \mathbf{z}_0 \right\|_2^2 + \frac{\rho}{2} \left\| \mathbf{x} - \mathbf{z}_1 \right\|_2^2 + \frac{\mu}{2} \left\| \Gamma_0 \mathbf{x} \right\|_2^2 + \frac{\mu}{2} \left\| \Gamma_1 \mathbf{x} \right\|_2^2 \;. \tag{32}$$

Full details of the ADMM algorithm, which are omitted here due to space considerations, can be found in [19, Sec. 6]. A spatial mask to support careful boundary handling [20] can easily be included in the data fidelity term, replacing $\left\| \mathbf{y}_0 \right\|_1$ with $\left\| W \mathbf{y}_0 \right\|_1$ for a diagonal weighting matrix $W$, with only minor changes to the algorithm.

### B. Multi-Channel

Since the extensions to both the multi-channel dictionary/single channel representation and single channel dictionary/multi-channel representation forms are straightforward, here we will focus on the extension to the product dictionary form of multi-channel convolutional sparse representation. Using the block-matrix notation from Sec. II-C, we can write the problem as

$$\begin{aligned}\arg\min_{X} \left\| DXB^T - S \right\|_1 + \lambda \left\| A \odot X \right\|_1 + \\ (\mu/2) \left\| \Gamma_0 X \right\|_2^2 + (\mu/2) \left\| \Gamma_1 X \right\|_2^2 \;,\end{aligned} \tag{33}$$

or in ADMM form

$$\begin{aligned}\arg\min_{X, Y_0, Y_1} \left\| Y_0 \right\|_1 + \lambda \left\| A \odot Y_1 \right\|_1 + (\mu/2) \left\| \Gamma_0 X \right\|_2^2 + (\mu/2) \left\| \Gamma_1 X \right\|_2^2 \\ \text{s.t.} \quad Y_0 = DXB^T - S \;, \quad Y_1 = X \;.\end{aligned} \tag{34}$$

The only computationally expensive subproblem of the corresponding ADMM algorithms consists of minimizing an equation of the form

$$\frac{\rho}{2} \left\| DXB^T - Z_0 \right\|_2^2 + \frac{\rho}{2} \left\| X - Z_1 \right\|_2^2 + \frac{\mu}{2} \left\| \Gamma_0 Z \right\|_2^2 + \frac{\mu}{2} \left\| \Gamma_1 X \right\|_2^2 \;. \tag{35}$$

Transforming into the DFT domain as in Sec. II-C, we have

$$\frac{\rho}{2} \left\| \hat{D} \hat{X} B^T - \hat{Z}_0 \right\|_2^2 + \frac{\rho}{2} \left\| \hat{X} - \hat{Z}_1 \right\|_2^2 + \frac{\mu}{2} \left\| \hat{\Gamma_0} \hat{X} \right\|_2^2 + \frac{\mu}{2} \left\| \hat{\Gamma}_1 \hat{X} \right\|_2^2 \;.$$

where $\hat{Z}_0 = F Z_0$, $\hat{Z}_1 = F Z_1$, and $\hat{\Gamma} = F \Gamma F^{-1}$. The minimizer of this functional is given by the linear equation

$$\begin{aligned}\rho \hat{D}^H \hat{D} \hat{X} B^T B + (\rho I + \mu \hat{\Gamma_0}^H \hat{\Gamma_0} + \mu \hat{\Gamma_1}^H \hat{\Gamma_1}) \hat{X} \\ = \rho \hat{D}^H \hat{Z}_0 B + \rho \hat{Z}_1 \;.\end{aligned} \tag{36}$$

Following the same procedure as in Sec. II-C, this equation can be transformed into a block-diagonal form in which each block can be efficiently solved via the Sherman-Morrison formula.

## V. RESULTS

The results reported here were computed using the Python implementation of the SPORCO library [18], [21] on a Linux workstation equipped with two Xeon E5-2690V4 CPUs. Experiments involving color images used a set of images derived from the MIRFLICKR-1M dataset [23] by cropping and rescaling, and those involving hyperspectral images were selected from a dataset hosted at the University of Manchester [26], [27].

### A. Dictionary Learning

We compare the performance of a number of different dictionary learning algorithms on color and hyperspectral imagery. While we have previously published such a comparison for color imagery [5], that work did not include any variants in which the CSC subproblem is solved via FISTA. Since the structure of the CSC problem with a multi-channel dictionary suggest that FISTA would enjoy an advantage for large number of channels [5, Sec. VI.A], we include this option in the comparisons presented here. In addition, due to the sensitivity of the FISTA algorithms to the step size parameter that was noted in our previous work [5, Sec. VII.G], we also include CSC and dictionary update algorithms based on a form of FISTA that is more robust to the choice of this parameter [22].

*1) Color Images:* We used training sets of 40 color images of size $512 \times 512$ pixels, derived from the MIRFLICKR-1M dataset [23] by cropping and rescaling. Each of the color components were divided by 255 so that values were within the interval [0,1], and were highpass filtered [24], [25], [3][19, Sec. 3] by subtracting a lowpass component computed by Tikhonov regularization with a gradient term [21, pg. 3], with regularization parameter $\lambda = 5.0$.

We compare the performance of the methods in learning a color dictionary of 100 filters of size $8 \times 8$, setting the sparsity parameter $\lambda = 0.1$. We used fixed penalty parameters $\rho$ and $\sigma$ for ADMM consensus, and three different parameter selection methods for FISTA: fixed inverse of gradient step parameters $L_{\mathrm{csc}}$ and $L_{\mathrm{cdl}}$, parameters adapted via standard backtracking (FISTA ST) [6], and parameters adapted via robust backtracking (FISTA Robust) [22]. The fixed parameters selected for ADMM and FISTA, listed in Table II, were selected by a grid search for the parameters giving the lowest functional value after 100 iterations for ADMM and 200 iterations for FISTA. The initial parameters for FISTA ST and FISTA Robust were set to values lower than the fixed FISTA setting: $L_{\mathrm{csc}} = 100$ and $L_{\mathrm{cdl}} = 1000$.

TABLE II
PARAMETERS FOUND BY GRID SEARCH.

| | | | Parameter | |
|---|---|---|---|---|
| **Method** | **Data** | $C$ | $\rho$ | $\sigma$ |
| **Cns** | color | 3 | 4.64 | 3.16 |
| | hyperspectral | 3 | 4.64 | 0.06 |
| | hyperspectral | 10 | 16.68 | 12.91 |
| | hyperspectral | 33 | 16.68 | 12.91 |
| | | | $L_{\mathrm{csc}}$ | $L_{\mathrm{cdl}}$ |
| **FISTA** | color | 3 | 251.19 | 4641.59 |
| | hyperspectral | 3 | 359.38 | 100.00 |
| | hyperspectral | 10 | 359.38 | 100.00 |
| | hyperspectral | 33 | 681.29 | 189.57 |

The convergence rates of the multi-image multi-channel sparse coding functional with respect to both computation time and iterations are displayed in Fig. 1. Note that all the methods achieve a similar functional value at the end of 500 iterations, with FISTA Robust yielding the best value and FISTA ST the worst. The ADMM parallel consensus method (Pll Cns) [5, Sec. VII.A] has the best performance with respect to time. While the FISTA algorithms could also be parallelized over different training images, our implementations do not do so. FISTA with fixed $L_{\mathrm{csc}}$ and $L_{\mathrm{cdl}}$ is the fastest of the FISTA variants, because no additional operations are needed to adapt the algorithm parameters. However, if these parameters are not set in the appropriate range, the method may become unstable and diverge. Due to the interleaved CSC and CDU updates [5, Sec. II.C], this tuning becomes a more critical issue. FISTA Robust provides an effective tracking mechanism for adapting the algorithm parameters, allowing both adaptive increments and decrements in the gradient step size, at higher computational cost than the other FISTA variants. In contrast, FISTA ST only allows for decrements in the gradient step size, which can yield very slow convergence rates.

*2) Hyperspectral Images:* We used training sets constructed from a single $1021 \times 1338$ hyperspectral image (Scene 6 from [27]) by cropping and taking a number of different channels per set. These sets had resolution of $712 \times 712$ and included 1, 3, 10 and 33 channels, respectively. Similarly to the color case, the different channels were highpass filtered with regularization parameter $\lambda = 5.0$.

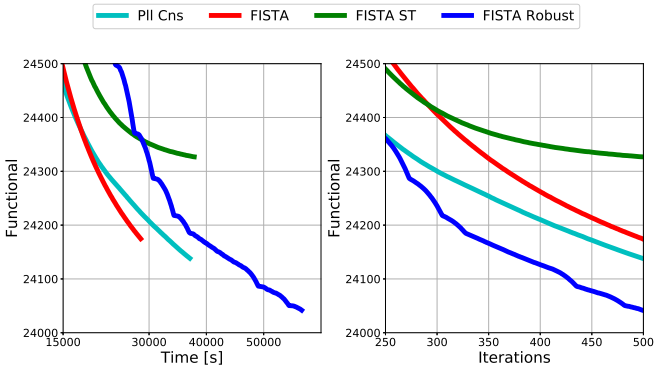



Fig. 1. Multi-Channel Dictionary Learning: A comparison on a set of $K = 40$, $512 \times 512$ color images (i.e. $C = 3$ channels) of the decay of the value of the multi-image multi-channel sparse coding functional with respect to run time and iterations.

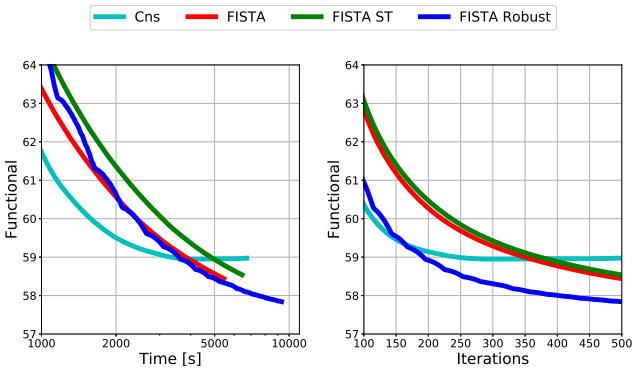



Fig. 2. Multi-Channel Dictionary Learning ($C = 3$): A comparison on a set of $C = 3$ channels of the decay of the value of the multi-channel sparse coding functional (8) with respect to run time and iterations.

We compare the performance of the methods in learning a multi-channel dictionary of 72 filters of size $12 \times 12$ for 1, 3, 10 and 33 channels, setting the sparsity parameter $\lambda = 0.03$. As in the color case, selected the best choices of fixed penalty parameters for ADMM consensus and the inverse of the gradient step parameter for FISTA, listed in Table II, by a parameter grid search to determine the parameters giving the lowest functional value after 100 iterations for ADMM and 200 iterations for FISTA. The initial parameters for FISTA ST and FISTA Robust were set to $L_{\mathrm{csc}} = 100$ and $L_{\mathrm{cdl}} = 100$. We used consensus (Cns) rather than parallel consensus because these data sets include only one training image.

Performance in terms of the convergence rate of the multi-channel sparse coding functional (8), with respect to both computation time and iterations, is compared in Figs. 2 – 4 for different number of channels $C$. It can be seen that in all the cases the methods achieve similar functional values at 500 iterations, with FISTA Robust showing the best results. In addition, Fig. 5(a) summarizes the time scaling with respect to $C$ for all the methods. Note that among all these batch methods, consensus is very competitive for a small number of channels but it does not scale as well as FISTA variants for large number of channels.

Even though FISTA Robust [22] requires more computation time and has slightly worse scalability than other FISTA variants, the very good convergence with respect to iterations and the insensitivity to initial parameter settings make it a very attractive method for the

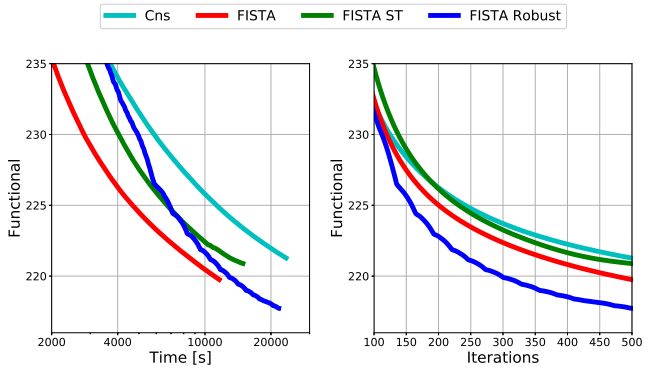



Fig. 3. Multi-Channel Dictionary Learning ($C = 10$): A comparison on a set of $C = 10$ channels of the decay of the value of the multi-channel sparse coding functional (8) with respect to run time and iterations.

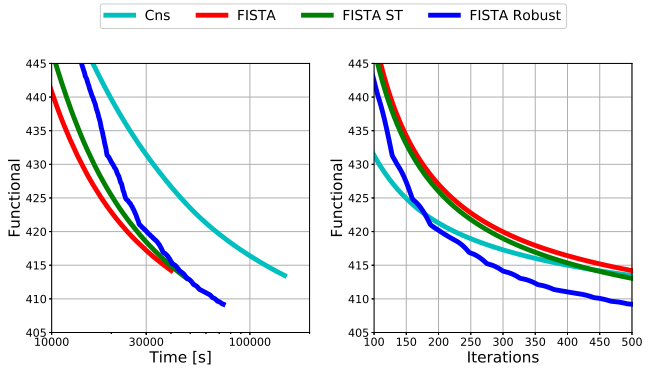
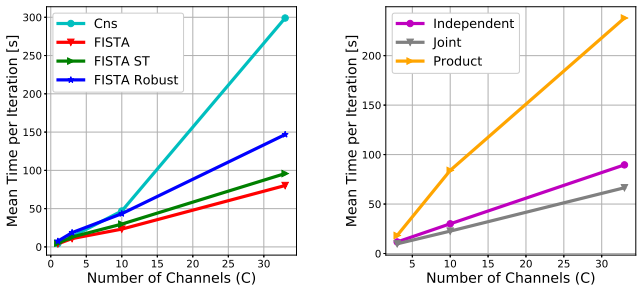



Fig. 4. Multi-Channel Dictionary Learning ($C = 33$): A comparison on a set of $C = 33$ channels of the decay of the value of the multi-channel sparse coding functional (8) with respect to run time and iterations.




(a) Multi-Channel Dictionary (b) Single Channel Dictionary

Fig. 5. Comparison of time per iteration for the dictionary learning methods for 1, 3, 10 and 33 channels.

CDL problem.

We also learned two different single channel dictionaries of 72 filters of size $12 \times 12$. The one was learned with an independent multi-channel representation ($\ell_1$ regularization term), using FISTA Robust for both the CSC and dictionary update steps, and the other was learned with a joint-sparsity coupled representation ($\ell_{2,1}$ regularization term), using ADMM for the CSC step and FISTA Robust for the CDL step. In addition we used Algorithm 1 with ADMM algorithms for the sparse coding and dictionary update stages to learn a multi-channel representation with a single channel dictionary constructed as the product of convolutional (72 filters of size $12 \times 12$) and channel dictionaries ($C \times \min(C, 15)$). Scaling of these single channel dictionary methods with respect to $C$ is summarized in Fig. 5(b). The labels used are 'Independent', 'Joint' and 'Product', respectively.

Note that methods using FISTA Robust, i.e. independent and joint, have good scaling performance, while the product method has worse scaling performance for large $C$. Also, note that the joint method has slightly better scaling than the independent method since the ADMM CSC algorithm used for the joint method has better time performance than the FISTA Robust CSC algorithm used for the independent method in the single-channel dictionary case.

Finally, it is worth noting that while online CDL [28], [29] is not considered here, we expect that it will prove to be more effective than batch methods for CDL of hyperspectral images since the considerable memory requirements of batch CDL methods will usually limit them to training with only a few hyperspectral images.

### *B. Denoising Comparison*

A $512 \times 512$ hyperspectral image with 33 channels obtained after cropping Scene 7 from [27] was used to compare generalization performance in a denoising task. The test image was generated by corrupting the original image with 33% salt-and-pepper noise, i.e. 33% of the pixels, at randomly selected locations were set at random to either 0 or 1.

The test image was restored using the following convolutional dictionary learning methods for multi-channel signals: (i) multi-channel dictionary with single channel representation, (ii) single channel dictionary with independent multi-channel representation, (iii) single channel dictionary with $\ell_{2,1}$ coupled multi-channel representation, (iv) dictionary constructed as the product of convolutional and channel dictionaries, with dictionary $B$ learned from the training scene and independent multi-channel representation, and (v) dictionary constructed as the product of a convolutional dictionary and a partial PCA transform matrix $B$ constructed from a PCA decomposition of the training scene[2], with $\ell_{2,1}$ coupled multi-channel representation. In addition, (vi) a multi-channel $\ell_1$-TV denoising (with an $\ell_1$ data fidelity term [31] and a vector-TV regularization term [32]) was applied to provide a comparison with a dictionary-free method. All the methods used a $\ell_1$ norm fidelity term.

The dictionaries were learned from Scene 6 from [27], as described in the experiments of the previous section. The corresponding dictionary learning algorithm was applied to learn the dictionary for each method, except for methods (iv) and (v) that used for $D$ the same convolutional dictionary learned from method (iii). The channel dictionary $B$, a dictionary of size $33 \times 15$, was learned from the training scene (Scene 6) in method (iv). In method (v), $B$ was constructed from the first 15 principal components of the PCA decomposition of the training scene.

Parameters for all of the methods were optimized via a grid search. Due to the computational cost of performing such a search on the full 33 channel test image, the search was performed for a subset of only 3 channels and the resulting parameters, reported in Table III, were extrapolated to the 33 channel problem. All optimization problems were solved over 200 iterations. The denoising performances of the different methods are compared in Table IV. Small differences in performance should not be considered to be significant due to the

[2]The use of this representation across the channels is suggested by the efficacy of PCA in spectral decorrelation of hyperspectral imagery [30]

TABLE III
PARAMETERS FOR DENOISING EXPERIMENT ($\lambda$, $\mu$, AND $\nu$ ARE THE $\ell_1$, $\ell_2$ OF GRADIENT, AND $\ell_{2,1}$ REGULARIZATION PARAMETERS RESPECTIVELY, AND $\rho$ IS THE ADMM PENALTY PARAMETER).

| | Parameter | | | |
|---|---|---|---|---|
| **Method** | $\lambda$ | $\mu$ | $\nu$ | $\rho$ |
| **(i)** | 4.360 | 33.11 | — | 4.36 |
| **(ii)** | 2.750 | 20.09 | — | 52.48 |
| **(iii)** | 0.003 | 316.23 | 6.81 | 46.41 |
| **(iv)** | 0.575 | 4.36 | — | 10.00 |
| **(v)** | 5.000 | — | 12.00 | 10.00 |
| **(vi)** | 0.820 | — | — | — |

compromise made in parameter selection, but it can be seen that the worst performance is obtained by method (ii), which makes no attempt to model the inter-channel correlations in the signal. A very large difference in performance is observed between the $\ell_1$-TV baseline method and the CSC methods. The CSC methods are, however, all more than an order of magnitude slower than the $\ell_1$-TV baseline.

TABLE IV
SALT & PEPPER DENOISING PERFORMANCE COMPARISON.

| | Method | | PSNR |
|---|---|---|---|
| | **Dictionary** | **Representation** | **[dB]** |
| **(i)** | Multi-Channel | Single Channel | 44.92 |
| **(ii)** | Single Channel | Independent Multi-Channel | 42.53 |
| **(iii)** | Single Channel | $\ell_{2,1}$ Coupled Multi-Channel | 45.08 |
| **(iv)** | Product (Dict. $B$) | Multi-Channel | 42.92 |
| **(v)** | Product (PCA $B$) | $\ell_{2,1}$ Coupled Multi-Channel | 43.24 |
| **(vi)** | $\ell_1$-TV | | 34.21 |

## VI. CONCLUSION

Our recent comparison of CDL algorithms found the most effective method for single-channel images to consist of an ADMM algorithm for the CSC stage and an ADMM Consensus algorithm for the dictionary update stage, both implemented in parallel [5]. Since the ADMM algorithm for CSC with a multi-channel dictionary [4] scales poorly with the number of channels, $C$, in this work we have considered CDL algorithms with FISTA methods for both the CSC and dictionary update stages[3]. The results presented in Sec. V-A confirm that this approach has significantly better scaling with $C$ than the ADMM-based approach, giving similar time performance for $C = 3$, and much better time performance when $C = 33$. While the standard FISTA methods have the disadvantage of having parameters that are more difficult to set for optimal performance than those of the ADMM methods, we have found that a recent robust FISTA variant [22] provides a useful compromise, being somewhat more computationally expensive, but having performance that is largely insensitive to the initial parameter selection.

We have also evaluated the effectiveness of a number of different methods for convolutional sparse representation of multi-channel imagery by comparing their performance in the denoising of hyperspectral imagery subject to salt & pepper noise. The results of Sec. V-B indicate that all of these different methods give very substantially better performance than the $\ell_1$-TV baseline method. As expected, the CSC-based methods that model inter-channel correlations provide the best performance. Although the product dictionary methods, (iv) and (v), give slightly inferior performance to the single-channel dictionary/$\ell_{2,1}$ term coupled method, (iii), it is worth noting that the use of the spectral dictionary $B$ makes it possible to reduce the number of channels in the sparse representation (i.e. by taking $M_B < C$) when memory requirements are a concern.

[3]An ADMM Consensus CSC algorithm with consensus across different channels would scale well with the number of channels, but we have not empirically evaluated this option.

## VII. ACKNOWLEDGMENTS

This research was supported by the U.S. Department of Energy via the LANL/LDRD Program.

## REFERENCES

[1] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Conf. Comp. Vis. Pat. Recog. (CVPR)*, Jun. 2010, pp. 2528–2535. doi:10.1109/cvpr.2010.5539957

[2] M. S. Lewicki and T. J. Sejnowski, "Coding time-varying signals using sparse, shift-invariant representations," in *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 11, 1999, pp. 730–736.

[3] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 301–315, Jan. 2016. doi:10.1109/TIP.2015.2495260

[4] ——, "Convolutional sparse representation of color images," in *Proc. IEEE Southwest Symp. Image Anal. Interp. (SSIAI)*, Santa Fe, NM, USA, Mar. 2016, pp. 57–60. doi:10.1109/SSIAI.2016.7459174

[5] C. Garcia-Cardona and B. Wohlberg, "Convolutional dictionary learning: A comparative review and new algorithms," *IEEE Trans. Computational Imaging*, vol. 4, no. 3, pp. 366–381, Sep. 2018. doi:10.1109/TCI.2018.2840334

[6] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009. doi:10.1137/080716542

[7] R. Chalasani, J. C. Principe, and N. Ramakrishnan, "A fast proximal method for convolutional sparse coding," in *Proc. Int. Joint Conf. Neural Net. (IJCNN)*, Aug. 2013. doi:10.1109/IJCNN.2013.6706854

[8] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010. doi:10.1561/2200000016

[9] H. Bristow, A. Eriksson, and S. Lucey, "Fast convolutional sparse coding," in *Proc. IEEE Conf. Comp. Vis. Pat. Recog. (CVPR)*, Jun. 2013, pp. 391–398. doi:10.1109/CVPR.2013.57

[10] B. Wohlberg, "Efficient convolutional sparse coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2014, pp. 7173–7177. doi:10.1109/ICASSP.2014.6854992

[11] Q. Barthélemy, A. Larue, A. Mayoue, D. Mercier, and J. I. Mars, "Shift & 2d rotation invariant sparse coding for multivariate signals," *IEEE Trans. Signal Process.*, vol. 60, no. 4, pp. 1597–1611, Apr. 2012. doi:10.1109/tsp.2012.2183129

[12] T. Dupré la Tour, T. Moreau, M. Jas, and A. Gramfort, "Multivariate convolutional sparse coding for electromagnetic brain signals," in *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 31, 2018, pp. 3295–3305.

[13] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011. doi:10.1016/j.laa.2010.09.020

[14] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, 2013. doi:10.1137/110837711

[15] S. Soltani, M. E. Kilmer, and P. C. Hansen, "A tensor-based dictionary learning approach to tomographic image reconstruction," *BIT Numerical Mathematics*, vol. 56, no. 4, pp. 1425–1454, 2016. doi:10.1007/s10543-016-0607-z

[16] A. J. Laub, *Matrix Analysis For Scientists And Engineers*. Society for Industrial and Applied Mathematics, 2004.

[17] H. V. Henderson and S. R. Searle, "The vec-permutation matrix, the vec operator and Kronecker products: a review," *Linear and Multilinear Algebra*, vol. 9, no. 4, pp. 271–288, 1981. doi:10.1080/03081088108817379

[18] B. Wohlberg, "SParse Optimization Research COde (SPORCO)," Software library available from http://purl.org/brendt/software/sporco, 2016.

[19] ——, "Convolutional sparse representations as an image model for impulse noise restoration," in *Proc. IEEE Image, Video Multidim. Signal Process. Workshop (IVMSP)*, Bordeaux, France, Jul. 2016. doi:10.1109/IVMSPW.2016.7528229

[20] ——, "Boundary handling for convolutional sparse representations," in *Proc. IEEE Conf. Image Process. (ICIP)*, Phoenix, AZ, USA, Sep. 2016, pp. 1833–1837. doi:10.1109/ICIP.2016.7532675

[21] ——, "SPORCO: A Python package for standard and convolutional sparse representations," in *Proc. 15th Python in Science Conference*, Austin, TX, USA, Jul. 2017, pp. 1–8. doi:10.25080/shinma-7f4c6e7-001

[22] M. I. Florea and S. A. Vorobyov, "A robust FISTA-like algorithm," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2017, pp. 4521–4525. doi:10.1109/ICASSP.2017.7953012

[23] M. J. Huiskes, B. Thomee, and M. S. Lew, "New trends and ideas in visual concept detection: The MIR Flickr retrieval evaluation initiative," in *Proc. Int. Conf. Multimedia Information Retrieval (MIR '10)*, 2010, pp. 527–536. doi:10.1145/1743384.1743475

[24] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, "Learning convolutional feature hierarchies for visual recognition," in *Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 23, 2010, pp. 1090–1098.

[25] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 2018–2025. doi:10.1109/iccv.2011.6126474

[26] D. H. Foster, K. Amano, S. M. C. Nascimento, and M. J. Foster, "Frequency of metamerism in natural scenes," *J. Opt. Soc. Am. A*, vol. 23, no. 10, pp. 2359–2372, 2006.

[27] D. H. Foster, S. M. C. Nascimento, and K. Amano, "Hyperspectral images of natural scenes." [Online]. Available: https://personalpages.manchester.ac.uk/staff/d.h.foster/Hyperspectral_images_of_natural_scenes_04.html

[28] J. Liu, C. Garcia-Cardona, B. Wohlberg, and W. Yin, "Online convolutional dictionary learning," in *Proc. IEEE Conf. Image Process. (ICIP)*, Beijing, China, Sep. 2017, pp. 1707–1711. doi:10.1109/ICIP.2017.8296573

[29] ——, "First and second order methods for online convolutional dictionary learning," *SIAM J. Imaging Sci.*, vol. 11, no. 2, pp. 1589–1628, 2018. doi:10.1137/17M1145689

[30] Q. Du and J. E. Fowler, "Hyperspectral image compression using JPEG2000 and principal component analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 2, pp. 201–205, Apr. 2007. doi:10.1109/LGRS.2006.888109

[31] S. Alliney, "Digital filters as absolute norm regularizers," *IEEE Trans. Signal Process.*, vol. 40, no. 6, pp. 1548–1562, Jun. 1992. doi:10.1109/78.139258

[32] P. Blomgren and T. F. Chan, "Color TV: total variation methods for restoration of vector-valued images," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 304–309, Mar. 1998. doi:10.1109/83.661180