

# Convolutional Sparse Representation of Color Images

Brendt Wohlberg

Theoretical Division, Los Alamos National Laboratory

Los Alamos, NM 87545, USA.

Email: brendt@lanl.gov

**Abstract**—Convolutional sparse representations differ from the standard form of sparse representations in their composition from coefficient maps convolved with dictionary filters instead of linear combinations of dictionary vectors. When applied to images, the standard form is usually independently computed over a set of overlapping image patches. The advantage of the convolutional form is that it provides a single-valued representation optimized over an entire signal, but this comes at a substantial computational cost. A recent algorithm for sparse coding via a convolutional form of Basis Pursuit DeNoising, however, has substantially reduced the computational cost of computing these representations. The present paper extends this algorithm to multi-channel signals such as color images, which has not previously been reported in the literature.

**Index Terms**—Sparse Coding, Dictionary Learning, Convolutional Sparse Representation, Multi-channel

## I. INTRODUCTION

Sparse representations [1], [2] are very widely used in image processing problems. When the dictionary is analytically defined and has a fast transform it is feasible to compute the representation for an entire image, but when the dictionary is learned from data, and therefore represented by an explicit matrix, the dictionary learning and sparse coding problems become intractable for images of reasonable size, the standard approach being to independently compute the representations on overlapping image blocks. Convolutional sparse representations [3] or the equivalent translation-invariant sparse representations (see [4, Sec. II.B]), which are far less well known than the standard form of sparse representation, offer an alternative structure that can be used to decompose an entire signal or image, albeit at considerable computational cost.

As is the case for standard sparse coding, there are various formulations of the sparse coding problem, the two main families of approaches consisting of heuristic/greedy algorithms directly targeting sparsity as an optimization criterion, and convex optimization methods for solving the  $\ell^1$  relaxation of the sparsity. The present paper considers the convolutional variant of Basis Pursuit DeNoising (BPDN) [5], referred to here as Convolutional BPDN (CBPDN), an optimization problem associated with the latter family of solution methods, and defined as

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

This research was supported by the U.S. Department of Energy through the LANL/LDRD Program, and by UC Lab Fees Research grant 12-LR-236660.

where  $\{\mathbf{d}_m\}$  is a set of  $M$  dictionary filters,  $*$  denotes convolution, and  $\{\mathbf{x}_m\}$  is a set of coefficient maps<sup>1</sup>. The computational expense of this optimization problem has limited the application of this representation to very small images, but recent Alternating Direction Method of Multipliers (ADMM) [6] algorithms have substantially reduced the computational cost of earlier methods [7], [4], making this a more viable method for signal and image inverse problems.

The present paper extends these algorithms to the decomposition of multi-channel signals, with a particular focus on color images. To the best of the author's knowledge, there is no significant prior work on the use of convolutional sparse representations for color images or on the development of efficient algorithms for solving the multi-channel CBPDN problem<sup>2</sup>. In addition to independent sparse coding of each channel, which can be achieved using existing algorithms, we consider two different methods of exploiting inter-channel statistical dependencies.

## II. SINGLE-CHANNEL DICTIONARY WITH JOINT SPARSITY

If each channel is represented using the same single-channel dictionary, then a reasonable assumption – at least in the case of RGB natural images – is that the same dictionary filters will tend to be active (i.e. non-zero) at the same spatial locations, but with different values. This model can be exploited by including an  $\ell^{2,1}$  norm term to penalize the solution in favor of joint sparsity, as in the problem

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

where  $\mathbf{s}_c$  denotes channel  $c$  of the  $C$  channels of signal  $\mathbf{s}$ , and the  $\ell^{2,1}$  norm takes an  $\ell^2$  norm along the channel index  $c$ , and and  $\ell^1$  norm over the spatial indices and filter index  $m$ .

Introducing auxiliary variables  $\{\mathbf{y}_{c,m}\}$  and dual variables  $\{\mathbf{u}_{c,m}\}$ , this problem can be solved via the ADMM itera-

<sup>1</sup>For notational simplicity  $\mathbf{s}$  and each of the  $\{\mathbf{x}_m\}$  are considered to be  $N$  dimensional vectors, where  $N$  is the number of pixels in image  $\mathbf{s}$ .

<sup>2</sup>Zeiler et al. [3] proposed an image representation that allows for multi-channel signals, without any specific discussion or experimental results addressing this aspect. Barthélemy et al. [8] considered multi-channel signals using algorithms that fall within the family of heuristic/greedy methods, and did not address the representation of images.

tions

$$\{\mathbf{x}_{c,m}\}^{(j+1)} = \arg \min_{\{\mathbf{x}_{c,m}\}} \frac{1}{2} \sum_c \left\| \sum_m \mathbf{d}_m * \mathbf{x}_{c,m} - \mathbf{s}_c \right\|_2^2 + \frac{\rho}{2} \sum_c \sum_m \left\| \mathbf{x}_{c,m} - \mathbf{y}_{c,m}^{(j)} + \mathbf{u}_{c,m}^{(j)} \right\|_2^2 \quad (3)$$

$$\{\mathbf{y}_{c,m}\}^{(j+1)} = \arg \min_{\{\mathbf{y}_{c,m}\}} \lambda \sum_c \sum_m \left\| \mathbf{y}_{c,m} \right\|_1 + \mu \left\| \{\mathbf{x}_{c,m}^{(j+1)}\} \right\|_{2,1} + \frac{\rho}{2} \sum_c \sum_m \left\| \mathbf{x}_{c,m}^{(j+1)} - \mathbf{y}_{c,m} + \mathbf{u}_{c,m}^{(j)} \right\|_2^2 \quad (4)$$

$$\mathbf{u}_{c,m}^{(j+1)} = \mathbf{u}_{c,m}^{(j)} + \mathbf{x}_{c,m}^{(j+1)} - \mathbf{y}_{c,m}^{(j+1)} \quad (5)$$

The primary computational cost is in solving Eq. (3). Since this problem is decoupled along channel index  $c$ , each channel can be solved independently. The problem for a single channel can be expressed in the form

$$\arg \min_{\{\mathbf{x}_m\}} \frac{1}{2} \left\| \sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s} \right\|_2^2 + \frac{\rho}{2} \sum_m \left\| \mathbf{x}_m - \mathbf{z}_m \right\|_2^2, \quad (6)$$

which can be solved efficiently by transforming to the DFT (Discrete Fourier Transform) domain and using the Sherman-Morrison formula to solve the resulting linear system, which has a system matrix consisting of the sum of a rank-one and a diagonal term [7], [4].

The solution to subproblem Eq. (4) is given by the proximal map of the sum of  $\ell^1$  and  $\ell^{2,1}$  norms [9], which can be computed as

$$\mathbf{y}_{c,m}^{(j+1)} = \mathcal{S}_{1,2,\lambda/\rho,\mu/\rho} \left( \mathbf{x}_{c,m}^{(j+1)} + \mathbf{u}_{c,m}^{(j)} \right), \quad (7)$$

where

$$\mathcal{S}_{1,\gamma}(\mathbf{u}) = \text{sign}(\mathbf{u}) \odot \max(0, \|\mathbf{u}\| - \gamma) \quad (8)$$

$$\mathcal{S}_{2,\gamma}(\mathbf{u}) = \frac{\mathbf{u}}{\|\mathbf{u}\|_2} \max(0, \|\mathbf{u}\|_2 - \gamma) \quad (9)$$

$$\mathcal{S}_{1,2,\alpha,\beta}(\mathbf{u}) = \mathcal{S}_{2,\beta}(\mathcal{S}_{1,\alpha}(\mathbf{u})). \quad (10)$$

### III. MULTI-CHANNEL DICTIONARY

An alternative to separately representing each channel together with a joint sparsity term is to introduce a multi-channel dictionary with respect to which the channels share a single representation. In this case we need both a new sparse coding algorithm and a new dictionary learning algorithm.

#### A. Sparse Coding

If we denote channel  $c$  of dictionary filter  $m$  as  $\mathbf{d}_{c,m}$ , then the multi-channel convolutional sparse coding problem can be expressed as

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

which can be solved via the ADMM iterations

$$\{\mathbf{x}_m\}^{(j+1)} = \arg \min_{\{\mathbf{x}_m\}} \frac{1}{2} \sum_c \left\| \sum_m \mathbf{d}_{c,m} * \mathbf{x}_m - \mathbf{s}_c \right\|_2^2 + \frac{\rho}{2} \sum_m \left\| \mathbf{x}_m - \mathbf{y}_m^{(j)} + \mathbf{u}_m^{(j)} \right\|_2^2 \quad (12)$$

$$\{\mathbf{y}_m\}^{(j+1)} = \arg \min_{\{\mathbf{y}_m\}} \lambda \sum_m \left\| \mathbf{y}_m \right\|_1 +$$

$$\frac{\rho}{2} \sum_m \left\| \mathbf{x}_m^{(j+1)} - \mathbf{y}_m + \mathbf{u}_m^{(j)} \right\|_2^2 \quad (13)$$

$$\mathbf{u}_m^{(j+1)} = \mathbf{u}_m^{(j)} + \mathbf{x}_m^{(j+1)} - \mathbf{y}_m^{(j+1)}. \quad (14)$$

Sub-problem Eq. (13) is solved via shrinkage/soft thresholding

$$\mathbf{y}_m^{(j+1)} = \mathcal{S}_{1,\lambda/\rho} \left( \mathbf{x}_m^{(j+1)} + \mathbf{u}_m^{(j)} \right). \quad (15)$$

The only computationally expensive step is Eq. (12), which can be written in slightly simpler form as

$$\arg \min_{\{\mathbf{x}_m\}} \frac{1}{2} \sum_c \left\| \sum_m \mathbf{d}_{c,m} * \mathbf{x}_m - \mathbf{s}_c \right\|_2^2 + \frac{\rho}{2} \sum_m \left\| \mathbf{x}_m - \mathbf{z}_m \right\|_2^2. \quad (16)$$

The DFT convolution theorem implies that Eq. (16) is equivalent to the frequency domain problem

$$\arg \min_{\{\hat{\mathbf{x}}_m\}} \frac{1}{2} \sum_c \left\| \sum_m \hat{D}_{c,m} \hat{\mathbf{x}}_m - \hat{\mathbf{s}}_c \right\|_2^2 + \frac{\rho}{2} \sum_m \left\| \hat{\mathbf{x}}_m - \hat{\mathbf{z}}_m \right\|_2^2, \quad (17)$$

where each  $D_{c,m}$  is a linear operator such that  $D_{c,m} \mathbf{x}_m = \mathbf{d}_{c,m} * \mathbf{x}_m$ , and the DFT domain variables corresponding to  $D_{c,m}$ ,  $\mathbf{x}_m$ ,  $\mathbf{s}_c$ , and  $\mathbf{z}_m$  are denoted by  $\hat{D}_{c,m}$ ,  $\hat{\mathbf{x}}_m$ ,  $\hat{\mathbf{s}}_c$ , and  $\hat{\mathbf{z}}_m$  respectively. Defining

$$\hat{D}_c = \begin{pmatrix} \hat{D}_{c,0} & \hat{D}_{c,1} & \dots \end{pmatrix}, \quad (18)$$

this can be expressed as

$$\arg \min_{\hat{\mathbf{x}}} \frac{1}{2} \sum_c \left\| \hat{D}_c \hat{\mathbf{x}} - \hat{\mathbf{s}}_c \right\|_2^2 + \frac{\rho}{2} \left\| \hat{\mathbf{x}} - \hat{\mathbf{z}} \right\|_2^2 \quad (19)$$

with solution

$$\left( \sum_c \hat{D}_c^H \hat{D}_c + \rho I \right) \hat{\mathbf{x}} = \sum_c \hat{D}_c^H \hat{\mathbf{s}}_c + \rho \hat{\mathbf{z}}. \quad (20)$$

In the single-channel case this involves solving a linear system with system matrix consisting of rank-one and diagonal components, which can be solved very efficiently using the Sherman-Morrison formula [4]. Direct application of the Sherman-Morrison formula in the multi-channel case is not possible since the system matrix consists of a sum of  $C$  rank-one components and a diagonal component. It is possible, however, to apply the Sherman-Morrison formula in sequence to  $\rho I + \hat{D}_0^H \hat{D}_0$ , then  $(\rho I + \hat{D}_0^H \hat{D}_0) + \hat{D}_1^H \hat{D}_1$  and so on. Such an algorithm [4, App. D] was developed for the single-channel dictionary update problem, which has the same structure as Eq. (20). The computational cost is quadratic in  $C$ , but the performance evaluation of this algorithm applied to the single-channel dictionary update [4, Sec. V.C] implies that for a small number of channels, such as in an RGB image, it is far more efficient than applying standard direct (Gaussian Elimination) or iterative (Conjugate Gradient) methods [4].

#### B. Dictionary Learning

The dictionary learning problem corresponding to Eq. (11) is

$$\arg \min_{\{\mathbf{d}_{c,m}\}, \{\mathbf{x}_{k,m}\}} \frac{1}{2} \sum_k \sum_c \left\| \sum_m \mathbf{d}_{c,m} * \mathbf{x}_{k,m} - \mathbf{s}_{k,c} \right\|_2^2 + \lambda \sum_k \sum_m \left\| \mathbf{x}_{k,m} \right\|_1. \quad (21)$$

Taking the same general approach as for the single-channel case [4, Sec. V], the dictionary update step can be posed as

$$\arg \min_{\{\mathbf{d}_{c,m}\}} \frac{1}{2} \sum_k \sum_c \left\| \sum_m \mathbf{x}_{k,m} * \mathbf{d}_{c,m} - \mathbf{s}_{k,c} \right\|_2^2 \text{ s. t. } \mathbf{d}_{c,m} \in C_P \forall c, m, \quad (22)$$

where the  $\mathbf{d}_{c,m}$  have the same spatial support as the  $\mathbf{x}_{k,m}$ ,  $P$  is a zero-padding operator, and

$$C_P = \{\mathbf{x} \in \mathbb{R}^N : (I - PP^T)\mathbf{x} = 0\}. \quad (23)$$

Rewriting with an auxiliary variable in a form suitable for ADMM gives

$$\arg \min_{\{\mathbf{d}_{c,m}\}} \frac{1}{2} \sum_k \sum_c \left\| \sum_m \mathbf{x}_{k,m} * \mathbf{d}_{c,m} - \mathbf{s}_{k,c} \right\|_2^2 + \sum_c \sum_m \iota_{C_{PN}}(\mathbf{g}_{c,m}) \text{ s.t. } \mathbf{d}_{c,m} = \mathbf{g}_{c,m} \quad \forall c, m. \quad (24)$$

This problem can be solved via an ADMM algorithm

$$\{\mathbf{d}_{c,m}\}^{(j+1)} = \arg \min_{\{\mathbf{d}_{c,m}\}} \frac{1}{2} \sum_k \sum_c \left\| \sum_m \mathbf{x}_{k,m} * \mathbf{d}_{c,m} - \mathbf{s}_{k,c} \right\|_2^2 + \frac{\sigma}{2} \sum_c \sum_m \left\| \mathbf{d}_{c,m} - \mathbf{g}_{c,m}^{(j)} + \mathbf{h}_{c,m}^{(j)} \right\|_2^2 \quad (25)$$

$$\{\mathbf{g}_{c,m}\}^{(j+1)} = \arg \min_{\{\mathbf{g}_{c,m}\}} \sum_c \sum_m \iota_{C_{PN}}(\mathbf{g}_{c,m}) + \frac{\sigma}{2} \sum_c \sum_m \left\| \mathbf{d}_{c,m}^{(j+1)} - \mathbf{g}_{c,m} + \mathbf{h}_{c,m}^{(j)} \right\|_2^2 \quad (26)$$

$$\mathbf{h}_{c,m}^{(j+1)} = \mathbf{h}_{c,m}^{(j)} + \mathbf{d}_{c,m}^{(j+1)} - \mathbf{g}_{c,m}^{(j+1)}. \quad (27)$$

The  $\{\mathbf{g}_{c,m}\}$  update is of the form

$$\arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \iota_{C_{PN}}(\mathbf{x}) = \text{prox}_{\iota_{C_{PN}}}(\mathbf{y}) = PP^T \mathbf{y} / \|PP^T \mathbf{y}\|_2. \quad (28)$$

The computationally expensive component is the  $\{\mathbf{d}_{c,m}\}$  update, which has the form

$$\arg \min_{\{\mathbf{d}_{c,m}\}} \frac{1}{2} \sum_k \sum_c \left\| \sum_m \mathbf{x}_{k,m} * \mathbf{d}_{c,m} - \mathbf{s}_{k,c} \right\|_2^2 + \frac{\sigma}{2} \sum_m \sum_c \left\| \mathbf{d}_{c,m} - \mathbf{z}_{c,m} \right\|_2^2, \quad (29)$$

and which can be expressed in the DFT domain as

$$\arg \min_{\{\hat{\mathbf{d}}_c\}} \frac{1}{2} \sum_k \sum_c \left\| \sum_m \hat{X}_{k,m} \hat{\mathbf{d}}_{c,m} - \hat{\mathbf{s}}_{k,c} \right\|_2^2 + \frac{\sigma}{2} \sum_m \sum_c \left\| \hat{\mathbf{d}}_{c,m} - \hat{\mathbf{z}}_{c,m} \right\|_2^2, \quad (30)$$

where  $\hat{X}_{k,m} = \text{diag}(\hat{\mathbf{x}}_{k,m})$ .

Defining  $\hat{X}_k = \begin{pmatrix} \hat{X}_{k,0} & \hat{X}_{k,1} & \dots \end{pmatrix}$  and

$$\hat{\mathbf{d}}_c = \begin{pmatrix} \hat{\mathbf{d}}_{c,0} \\ \hat{\mathbf{d}}_{c,1} \\ \vdots \end{pmatrix} \quad \hat{\mathbf{z}}_c = \begin{pmatrix} \hat{\mathbf{z}}_{c,1} \\ \hat{\mathbf{z}}_{c,1} \\ \vdots \end{pmatrix}, \quad (31)$$

this problem can be expressed as

$$\arg \min_{\{\hat{\mathbf{d}}_c\}} \frac{1}{2} \sum_k \sum_c \left\| \hat{X}_k \hat{\mathbf{d}}_c - \hat{\mathbf{s}}_{k,c} \right\|_2^2 + \frac{\sigma}{2} \sum_c \left\| \hat{\mathbf{d}}_c - \hat{\mathbf{z}}_c \right\|_2^2. \quad (32)$$

Further defining

$$\hat{\Xi}_k = \begin{pmatrix} \hat{X}_k & 0 & \dots \\ 0 & \hat{X}_k & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} \quad \hat{\mathbf{d}} = \begin{pmatrix} \hat{\mathbf{d}}_0 \\ \hat{\mathbf{d}}_1 \\ \vdots \end{pmatrix} \quad \hat{\mathbf{z}} = \begin{pmatrix} \hat{\mathbf{z}}_0 \\ \hat{\mathbf{z}}_1 \\ \vdots \end{pmatrix} \quad \hat{\mathbf{s}}_k = \begin{pmatrix} \hat{\mathbf{s}}_{k,0} \\ \hat{\mathbf{s}}_{k,1} \\ \vdots \end{pmatrix}, \quad (33)$$

allows simplification to the form

$$\arg \min_{\hat{\mathbf{d}}} \frac{1}{2} \sum_k \left\| \hat{\Xi}_k \hat{\mathbf{d}} - \hat{\mathbf{s}}_k \right\|_2^2 + \frac{\sigma}{2} \left\| \hat{\mathbf{d}} - \hat{\mathbf{z}} \right\|_2^2 \quad (34)$$

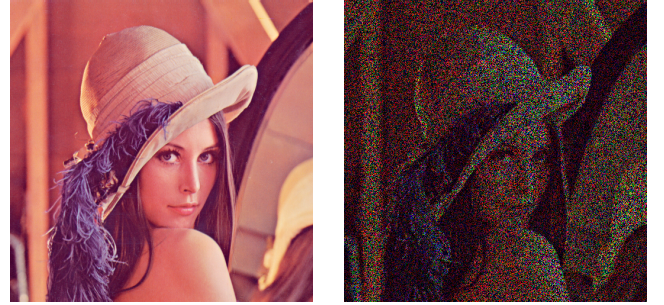
with solution given by

$$\left( \sum_k \hat{\Xi}_k^H \hat{\Xi}_k + \sigma I \right) \hat{\mathbf{d}} = \sum_k \hat{\Xi}_k^H \hat{\mathbf{s}}_k + \sigma \hat{\mathbf{z}}. \quad (35)$$

This problem can again be solved via the iterated Sherman-Morrison algorithm [4, App. D]; unlike sparse coding, the change from the single to the multiple channel dictionary update increases the sizes of the arrays to be processed, but does not change the fundamental problem structure.

#### IV. RESULTS

We compare the performance of three different models on an ‘‘inpainting’’ problem<sup>3</sup> posed by a color image with samples zeroed in randomly selected (but known) channels and spatial locations so that only 1/3 of of the samples remain. The reference and corrupted images are displayed in Fig. 1(a) and Fig. 1(b) respectively.



(a) Reference (b) Corrupted

Fig. 1. Ground truth and corrupted test images.

The natural solution of such an ‘‘inpainting’’ problem via convolutional sparse representations (shown in the single channel case for the sake of clarity) is  $\hat{\mathbf{s}} = \sum_m \mathbf{d}_m * \hat{\mathbf{x}}_m$ , where  $\{\hat{\mathbf{x}}_m\}$  is the minimizer of

$$\arg \min_{\{\mathbf{x}_m\}} \frac{1}{2} \left\| W \left( \sum_m \mathbf{d}_m * \mathbf{x}_m - \mathbf{s} \right) \right\|_2^2 + \lambda \sum_m \|\mathbf{x}_m\|_1, \quad (36)$$

and  $W$  is a mask taking on values zero and one at unknown and known pixel locations respectively. Direct application of the efficient DFT domain solution to this problem is not possible since spatial mask  $W$  is not diagonalized by the DFT. There is, however, a computationally cheap alternative approach [10, Sec. 4] to integrating the mask  $W$  into the optimization problem: introduce an additional impulse filter into the dictionary and apply a spatial weighting to the  $\ell^1$  norm of this filter so that there is no penalty for employing it

<sup>3</sup>The quotes are intended to differentiate this problem from the true inpainting problem of estimating relatively large contiguous regions of corrupted pixels. While this randomly located corruption problem is not as difficult, and arguably not as useful in actual applications, it remains a very convenient problem on which to test and compare image models.

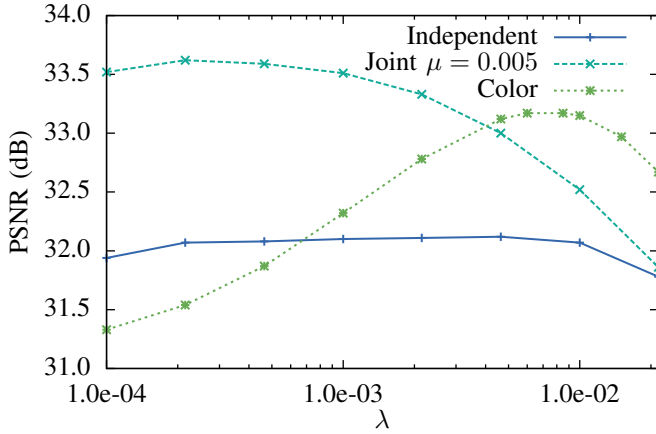


Fig. 2. Inpainting performance comparison. The joint sparse model parameter  $\mu$  was selected by searching for best performance over a range of values.

in spatial locations where the image has been corrupted, and a large penalty in locations where pixel values are known. Since the impulses are completely unpenalized at spatial locations that are desired to be masked out of the data fidelity term, they adapt to zero out any error at these locations, achieving the same effect as the weighting of the data fidelity term. The reconstructed image is obtained by summing over all of the  $\mathbf{d}_m * \hat{\mathbf{x}}_m$  except for those corresponding to an impulse filter.

Three different convolutional sparse models were used to solve the “inpainting” problem: completely independent channel representation, separate channel representation linked by a joint sparsity term, and a combined representation with a multi-channel dictionary. The same basic approach to solving the problem was adopted in each case: augment the dictionary with impulse filter(s) to represent the corruptions, as described above, and weight the  $\ell^1$  (and  $\ell^{2,1}$ , where appropriate) norm so that there is no penalty on these impulse filters corresponding to the locations/channels of the corrupted pixels. A learned dictionary consisting of 144 filters of size  $12 \times 12$  was used for both methods requiring a single-channel dictionary, and the 3-channel dictionary, consisting of 144 filters of size  $12 \times 12 \times 3$ , was learned from a set of color images, using a three channel replication of the single channel dictionary described above for initialization. In the single-channel dictionary cases a single impulse filter was appended to the learned dictionary, and in the multi-channel dictionary case an impulse filter for each channel was appended to the learned dictionary. Since convolutional sparse representations typically can not effectively represent the low-frequency image components, an estimate of this component was computed using Total Variation (TV) inpainting [11], with regularization parameter set to 0.07 in all three cases.

The average times per iteration over 100 iterations of the independent, the joint sparse, and the color dictionary algorithms were 18.2s, 21.4s, and 21.7s respectively. It is particularly noteworthy that the color dictionary algorithm is only marginally slower than the other two methods: while the slower iterated Sherman-Morrison linear solver is used in this

case, this is compensated for by having to perform FFTs on coefficient arrays of one third of the size of those used in the other methods. The performance of the three methods is compared in Fig. 2. Both the joint sparsity and color dictionary methods substantially outperform that based on completely independent modeling of the three channels. It is also interesting that the joint sparsity method outperforms the color dictionary method, but it is worth noting that (i) the color dictionary method has a substantially smaller representation, which may be an advantage in some contexts, and (ii) given the additional complexity of color dictionary learning, some of the performance deficit may reflect deficiencies in the color dictionary, rather than inherent inferiority of the model.

## V. CONCLUSIONS

While not entirely straightforward, the efficient DFT domain solution of the CBPDN problem can be extended to support decomposition of multi-channel signals. Two different methods that model inter-channel dependencies show substantially better performance than independent channel representation in an inpainting-like test problem, at marginal additional computational cost.

Implementations of the algorithms proposed here will be included in a future release of the SPORCO library [12].

## REFERENCES

- [1] A. M. Bruckstein, D. L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009. doi:10.1137/060657704
- [2] J. Mairal, F. Bach, and J. Ponce, “Sparse modeling for image and vision processing,” *Foundations and Trends in Computer Graphics and Vision*, vol. 8, no. 2-3, pp. 85–283, 2014. doi:10.1561/06000000058
- [3] 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
- [4] B. Wohlberg, “Efficient algorithms for convolutional sparse representations,” *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 301–315, Jan. 2016. doi:10.1109/TIP.2015.2495260
- [5] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998. doi:10.1137/S1064827596304010
- [6] 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:10.1561/22000000016
- [7] 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
- [8] 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
- [9] R. Chartrand and B. Wohlberg, “A nonconvex ADMM algorithm for group sparsity with sparse groups,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2013, pp. 6009–6013. doi:10.1109/ICASSP.2013.6638818
- [10] B. Wohlberg, “Endogenous convolutional sparse representations for translation invariant image subspace models,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, Oct. 2014, pp. 2859–2863. doi:10.1109/ICIP.2014.7025578
- [11] J. Shen and T. F. Chan, “Mathematical models for local nontexture inpaintings,” *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2002. doi:10.1137/s0036139900368844
- [12] B. Wohlberg, “SParse Optimization Research COde (SPORCO),” Matlab library available from <http://math.lanl.gov/~brendt/Software/SPORCO/>, 2015, version 0.0.3.