# EFFICIENT CONVOLUTIONAL DICTIONARY LEARNING USING PARTIAL UPDATE FAST ITERATIVE SHRINKAGE-THRESHOLDING ALGORITHM

*Gustavo Silva* *Paul Rodriguez*

Electrical Engineering Department, Pontificia Universidad Católica del Perú, Lima, Peru
Email: {gustavo.silva, prodrig}@pucp.edu.pe

**ABSTRACT**

Convolutional sparse representations allow modeling an entire image as an alternative to the more common independent patch-based formulations. Although many approaches have been proposed to efficiently solve the convolutional dictionary learning (CDL) problem, their computational performance is constrained by the dictionary update stage. In this work, we include two improvements to existing methods (i) a dictionary update based on Accelerated Proximal Gradient (APG) approach computed in the frequency domain and (ii) a new update model reminiscent of the Block Gauss Seidel (BGS) method. Our experimental results show that both improvements provide a significant speedup with respect to the state-of-the-art methods. In addition, dictionaries learned by our proposed method yield matching performance in terms of reconstruction and sparsity metrics in a denoising task.

***Index Terms*—** Convolutional Dictionary Learning, Convolutional Sparse Representation, FISTA, APG

## 1. INTRODUCTION

Sparse representations (SR) [1],[2],[3] are well-known techniques that have led to outstanding results for a broad range of signal and image processing tasks, and computer vision applications. In the past 5 years, their convolutional form [4],[5] has received increased attention, since it overcomes the patch-based drawbacks of redundancy by modeling the entire signal as a sum over a set of convolutions between coefficient maps and their corresponding dictionary filters. In particular, the standard convolutional formulation of dictionary learning is an extension of Convolutional Basis Pursuit Denoising (CBPDN) defined as

$$\underset{\{x_{k,m}\},\{d_m\}}{\arg\min} \frac{1}{2}\sum_k \Big\| \sum_m d_m * x_{k,m} - s_k \Big\|_2^2 + \lambda \sum_k \sum_m \|x_{k,m}\|_1 \quad \text{s.t.} \quad \|d_m\|_2 = 1 \quad \forall m \ , \tag{1}$$

where $\{x_{k,m}\}$ represents the sets of coefficient maps, $\{d_m\}$ a set of dictionary filters, $\{s_k\}$ the training images and $\lambda$ is the regularization parameter of sparsity. The constraint on the norms of the filters is required to avoid the scaling ambiguity between filters and coefficient maps.

Recently, several effective algorithms [5],[6],[7] based on the Alternating Direction Method of Multipliers (ADMM) framework have been proposed to deal with the most computationally demanding linear system of the dictionary update in the frequency domain. While the aforementioned system has closed-form solution, which can be directly solved via either matrix inversion techniques or conjugate gradient method in each outer-loop, it can also be computationally expensive.

In this paper, we incorporate two particular contributions on the standard CDL approaches that substantially improve runtime performance. First, we extend the use of an efficient APG-based solution partially computed in the frequency domain, previously introduced in [5] only for the sparse coding (SC) sub-problem (2) to both sub-problems of CDL. Secondly, we describe a new update model inspired on the Block Gauss Seidel (BGS) method [8], which enables the computation of partial sets of coefficient maps during each sparse coding stage.

Our experimental results (see Section 4) show that the proposed APG-based algorithm for CDL problem, when using 10 to 40 training images, is about 2.2 to 5.3 times faster than the Iterated Sherman Morrison solution [5], about 2.5 times faster than the Conjugate Gradient solution [5]; and about 1.5 times faster than the Consensus solution [7]. The proposed BGS-inspired update model applied in our algorithm increases its speed up by a factor of 1.6 and 2.5 times using 2 and 5 partitions respectively. Furthermore, the dictionaries learned by our method provide equivalent performance as those learned by other leading methods when applied to the denoising task.

## 2. PREVIOUS RELATED WORK

The convolutional dictionary learning (1) is a non-convex problem when being simultaneously evaluated in both variables $\{x_{k,m}\}$ and $\{d_m\}$, but it becomes convex in either variable when keeping the other one constant. This latter form is usually handled by an alternating approach between both sub-problems: sparse coding (coefficient update) and dictionary learning (dictionary update). This section will explain in more details these sub-problems and the existing methods used to deal with them.

### 2.1. Convolutional sparse coding

Considering a single observed image $\{s\}$, the most common formulation of the CSC problem is posed as

$$\underset{\{x_m\}}{\arg\min} \frac{1}{2}\| \sum_m d_m * x_m - s\|_2^2 + \lambda \sum_m \|x_m\|_1 \ . \tag{2}$$

Early approaches such as Iterative Shrinkage-Thresholding algorithm (ISTA) [9] and its accelerated version (FISTA) [10] addressed (2) in the spatial domain. Due to the high complexity associated to the convolution operations, [11] proposed a direct derivation of the CDL problem (1) based on Augmented Lagrangian framework, in which the most computationally expensive linear system of each sub-problems were carried out in the frequency domain. [6] and [12] later improved the SC stage via particular arrangements of the linear system derived from an ADMM model, along with matrix inversion methods.

### 2.2. Dictionary learning

The convolutional variant of the Method of Optimal Direction [9],[13] is one of the most widely used formulations for the dictionary learning sub-problem, which can be partially solved in the frequency by rewriting it in a constrained form

$$\underset{\{d_m\}}{\arg\min}\ \frac{1}{2}\sum_k \Big\|\sum_m x_{k,m} * d_m - s_k\Big\|_2^2 \quad \text{s.t} \quad d_m \in C_{PN} \quad , \tag{3}$$

where $C_{PN}$ is the constraint set for an adequate spatial support and normalized dictionary. Denoting $i_{C_{PN}}$ as the indicator function of the set $C_{PN}$, its unconstrained derivation is given by

$$\underset{\{d_m\}}{\arg\min}\ \frac{1}{2}\sum_k \Big\|\sum_m x_{k,m} * d_m - s_k\Big\|_2^2 + \sum_m i_{C_{PN}}(d_m) \ . \tag{4}$$

Unlike [11] (presented in Section 2.1), [5] proposed an alternative approach to solved the expensive linear system of the dictionary stage (4) using either Iterated Sherman Morisson (ISM) or Gradient Conjugate (GC). [6] redefined this sub-problem (4) in consensus-ADMM (CSS) and 3D forms to decouple the solution across the number of images, later the convergence rate were improved by [7]. Furthermore, [14],[15] presented online learning methods to reduce the use of resources. During the development of this work, [16] independently extended an APG-based solution (called in that work as FISTA-based) for the DL problem, in which only the gradient is computed in the frequency domain.

## 3. PROPOSED METHODS

### 3.1. Frequency domain APG

As we mentioned in Section 1 we approach the standard CDL problem (1) by using the APG framework. We have heuristically observed that the most suitable way to handle both updates of this problem is to use APG in both cases, since this combination showed to deliver better runtime performance with similar convergence rate in compassion to ADMM-APG combinations. Due to space constraints, we do not report this comparison in the present work [1] and focus mainly on explaining the proposed dictionary update formulation. However, the coefficient update (I) of Algorithm 1 can be derived by following an analogous chain of derivation as described here for the dictionary update (II).

The standard APG-based solution presented in the Algorithm 1 is composed by a gradient step, proximal operator, step size calculation and Nesterov's accelerated gradient calculation, most of which are computationally demanding if computed in the spatial domain. Due to this fact, we propose to perform most of these steps in the frequency domain, keeping only the proximal operator in the spatial domain, to avoid unnecessary convolutions or transformations between both domains.

To address the gradient of the fidelity term (4), labeled $\nabla F$, in the frequency domain, we define a linear operator of $X_{k,m}$ such that $X_{k,m}d_m = x_{k,m} * d_m$ and denote $X_{k,m}, d_m$ and $s_k$ in the DFT domain as $\hat{X}_{k,m}$ (diagonal matrix), $\hat{d}_m$ and $\hat{s}_k$ (column vectors) respectively. The fidelity term is arranged as

$$\begin{aligned}\frac{1}{2}\sum_k \Big\|\sum_m \hat{X}_{k,m}\hat{d}_m - \hat{s}_k\Big\|_2^2 &= \frac{1}{2}\sum_k \Big\|\hat{X}_k d_f - \hat{s}_k\Big\|_2^2 \\ &= \frac{1}{2}\Big\|X_f d_f - s_f\Big\|_2^2 \ ,\end{aligned} \tag{5}$$

[1]The set of experiments on this pair of update combinations is available on [17]

where $\hat{X}_k = (\hat{X}_{k,1}\ \hat{X}_{k,2}\ \cdots)$, $X_f = (\hat{X}_1\ \hat{X}_2\ \cdots)^T$,

$$d_f = (\hat{d}_1\ \hat{d}_2\ \cdots)^T \quad \text{and} \quad s_f = (\hat{s}_1\ \hat{s}_2\ \cdots)^T.$$

The inexact line search as back-tracking [18],[19] is a customary option since the exact search [20],[21] line could be computationally prohibitive. However, in the frequency domain, an exact line search can be effectively perform via (6).

$$\underset{\{\rho\}}{\arg\min}\ \frac{1}{2}\Big\|X_f(d_f - \rho\nabla\mathcal{F}(g_f)) - s_f\Big\|_2^2 \ , \tag{6}$$

where its solution yields a single step size for the dictionary update and $K$ values for the coefficient update in case of analogous approach.

A normalization of the auxiliary dictionary is required to avoid the scaling ambiguities when it is passed to the other sub-problem. Using Parseval's theorem, we extend this normalization to the frequency domain ($\|g_f\|_2/\sqrt{N} = 1$, where N is the number of pixels).

**Algorithm 1:** Frequency domain APG

**for** $k = 1 : \text{maxIter}$ **do**
- **• Coefficient update (I):**
  1: Compute $X_f^{k+1}$ via the frequency domain APG approach.
- **• Dictionary update (II) :**
  2: Compute gradient in the frequency domain $\nabla F(g_f^k) = (X_f^{k+1})^H (X_f^{k+1} g_f^k - s_f)$
  3: Compute step size in the frequency domain $\rho = \|\nabla F(g_f^k)\|_2^2 / \|X_f^{k+1}\nabla F(g_f^k)\|_2^2$
  4: Compute dictionary
  $h^{k+1} = IFFT2\{g_f^k - \rho\cdot\nabla F(g_f^k)\}$
  $d^{k+1} = \text{prox}_{i_{C_{PN}}}(h^{k+1})$
  $d_f^{k+1} = FFT2\{d^{k+1}\}$
  5: Compute auxiliary dictionary $g_f^{k+1}$ (Nesterov accelerated method) in the frequency domain
  $\gamma^{k+1} = (1 + \sqrt{1 + 4(\gamma^{k+1})^2})$
  $g_f^{k+1} = d_f^{k+1} + \frac{\gamma^k - 1}{\gamma^{k+1}}(d_f^{k+1} - d_f^k)$
  6: Compute normalization of auxiliary dictionary $g_f^{k+1} = \sqrt{N}\cdot g_f^{k+1}/\|g_f^{k+1}\|_2$

### 3.2. Partial update model

Given an efficient dictionary update implementation (as proposed in Section 3.1), we noted that the coefficient update becomes the dominant part of the whole CDL problem. With this in mind, we explore a new update model inspired by BGS method [8] (a.k.a Alternating Optimization [22]) which raises the minimization problem for a given function $f(x)$ as

$$x_i^{k+1} = \underset{y\in x_i}{\arg\min}\, f(x_1^{k+1},\ \cdots,\ x_{i-1}^{k+1},\ y,\ x_{i+1}^{k+1},\ \cdots,\ x_r^{k+1}) \ , \tag{7}$$

where the optimization is performed for a single partition of the interest variable by keeping the other partitions fixed.

Extrapolating this form in the CDL problem, we split the data set $\{s_k\}$ into R partitions

$$s_k = \{s_k^{(1)},\ s_k^{(2)}, \dots,\ s_k^{(R)}\}$$

and define the variables $r = \{1,\ldots,R\}$ and $k = \{1,\ldots,P_r\}$, where $P_r$ is the partition size. The full SC problem (2) of CDL can be written as

$$x_{k,m}^{(i,r)} = \underset{\{x_{k,m}\}}{\arg\min}\ \frac{1}{2}\sum_{k=1}^{P_r}\Big\|\sum_m d_m * x_{k,m} - s_k^{(r)}\Big\|_2^2 + \lambda\sum_{k=1}^{P_r}\sum_m \|x_{k,m}\|_1 \tag{8}$$

where a single partition of coefficient maps is estimated in each outer-loop of the CDL problem. After computing the current partial set $x_{k,m}^{(i,r)}$, the complete variable of coefficient maps $x_{k,m}$ is composed from the current estimated partition and its previous values of the other partitions.

$$x_{k,m}^{(i)} = [x_{k,m}^{(i-1,1)},\ \ldots\ ,\ \mathbf{x}_{\mathbf{k,m}}^{(\mathbf{i,r})},\ \ldots\ ,\ x_{k,m}^{(i-1,R)}] \tag{9}$$

This complete set of coefficient maps is used to estimate the current dictionary given by (4), reproduced here for convenience.

$$d_m^{(i)} = \underset{\{d_m\}}{\arg\min}\ \frac{1}{2}\sum_k\Big\|\sum_m x_{k,m}^{(i)} * d_m^{(i-1)} - s_k\Big\|_2^2 + \sum_m i_{C_{PN}}(d_m^{(i-1)})$$

At the end of each outer-loop the variable $r$ is reassigned as $(r+1)\ mod\ R$ to periodically switch from the first to the last partition.

As this is a generic structure, it could be applied to any CDL framework to solve each subproblem. However, we choose to merge this model with our APG-based solution, since in the Section 4, we show that our APG algorithm is computationally more efficient than the other approaches.

## 4. RESULTS

### 4.1. Experimental framework

We present two distinct set of experiments that were carried out using MATLAB R2014b running on an Desktop computer with Intel i7-7700K CPU (4.20 GHz, 8MB Cache, 32GB RAM).

We first compare the computational performance of our method with respect to the standard CDL algorithms using different sets of training and validation images. The training sets consist of 10, 20, 30 and 40 gray-scale images of size $256 \times 256$ pixels, cropped and rescaled from a set of images obtained from Flickr. The validation set is 5 images with similar characteristics as the other mentioned sets, that were not used during the training. Furthermore, 32 filters of size $12\times12$ were learned for each set of 10, 20 and 40 training images using the sparsity parameter $\lambda = 0.2$ and 1000 fixed iterations. For the values of the validation graphs, we saved the dictionaries at each iteration while training and used them to solve the CBPDN problem until converge for $\lambda = 0.2$ and stored the obtained functional value.

We then evaluate the performance in terms of PSNR, SSIM and sparsity metrics [2] of the previously learned dictionaries (of size $12 \times 12 \times 32$, learned from 40 training images) solving the SC problem (2) using the ADMM-based MATLAB algorithm of the SPORCO library [23]. The testing set includes 4 standard gray-scale images such as Mandrill, Barbara, Lena and Peppers, that were corrupted with AWGN level $\sigma = 0.2$. Since CBPDN has an adjustable parameter $\lambda$, in order to present fair comparisons, we used a search grid over $\lambda \in [0.05 - 0.95]$ to find the optimal value that provides the best PSNR and its corresponding SSIM and sparsity for each learned dictionary.

[2]Sparsity measure is defined as $100 \cdot \|x\|_0/N$, where $x$ is the coefficient maps and $N$ is the number of pixel in a test image.

The CDL algorithms used in our experiments are the following:

○ **Iterated Sherman-Morrison (ISM)** and **Conjugate Gradient (CG):** The CDL algorithms proposed in [5].

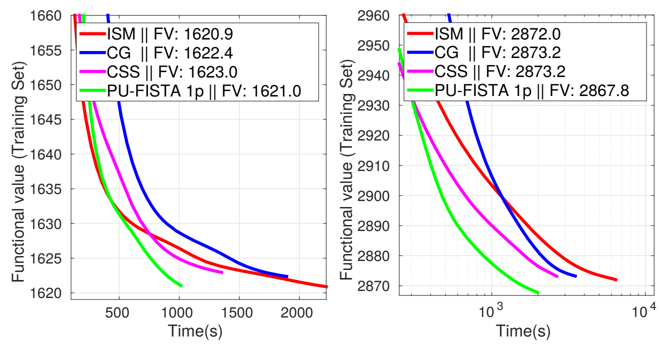
○ **ADMM Consensus (CSS):** The CDL algorithm proposed in [7].

○ **PU-FISTA:** Our proposed APG-based algorithm with partial update structure (code available on [17]). It's worth noting that PU-FISTA solution with a single partition is equivalent to using the APG-based method proposed in Section 3.1 without any partial update model.

Since each outer-loop updates a single partition of the coefficient maps, this affect the $l_1$-term when computing the functional value (FV) of the training set, making this an unsuitable convergence reference. A fair comparison of the functional value would be with the validation set.

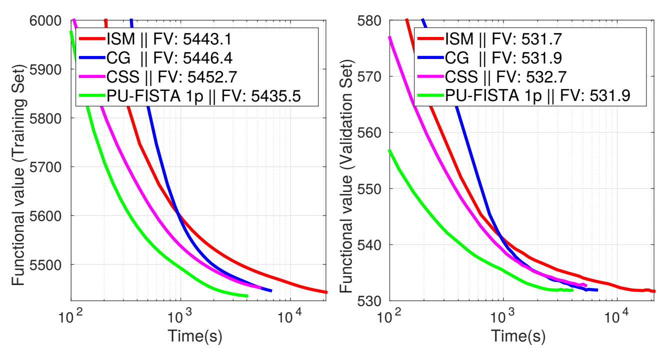
### 4.2. Simulations of the learning

We show in Figure 1, the performance of the existing CDL methods (ISM, CG and CSS) along with our proposed method in terms of functional values of (1) with respect to the learning runtime, using 10 and 20 training images. We observe that in both cases, all methods converge to similar values with distinct runtimes and behaviors. Our proposed method outperforms the rest by achieving the same functional value in less time (at the beginning ISM has a relatively better convergence ratio for 10 training images).




(a) 10 training images (b) 20 training images

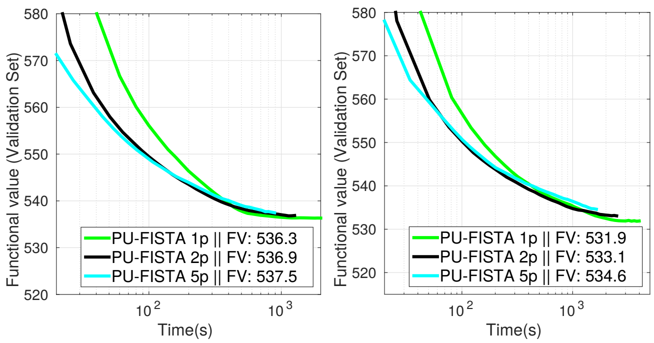
**Fig. 1**: A comparison on the set of 10 and 20 training images of the functional value decay with respect to execution time for the DCL methods.




(a) FV of the training set (b) FV of the validation set

**Fig. 2**: A comparison of the value decay of **the training and validation functional** with respect to execution time for the DCL methods on a set of 40 training images.

We report in Figure 2 the same comparisons as in Figure 1, but for a larger set of training images (40 images). It also includes the progress of the functional value in the validation set in order to observe the generalization of the estimated dictionary with respect to the training runtime. We note that while the number of training images becomes larger our APG algorithm consistently outperforms the other ones in terms of runtime and functional value. Furthermore, we can see in Figure 2.b the same pattern including that our method provides similar generalization.




(a) 20 training images (b) 40 training images

**Fig. 3**: A comparison of the value decay of **validation functional** respect to execution time for the set of 20 and 40 training images on our proposed method with 1, 2 and 5 partitions.

In Figure 3, we present the comparison of our APG-based algorithm with partial update structure for 1, 2 and 5 partitions. As can be observed the PU-FISTA with a single partition achieves a lower point of convergence, which indicates that its learned dictionaries must be better at generalizing testing set. Although PU-FISTA with 2 partitions achieve a slightly higher point of convergence, it is reached in a less execution time. On the other hand, PU-FISTA with 5 partitions is faster, but it needs more iterations to converge while the image set becomes larger.

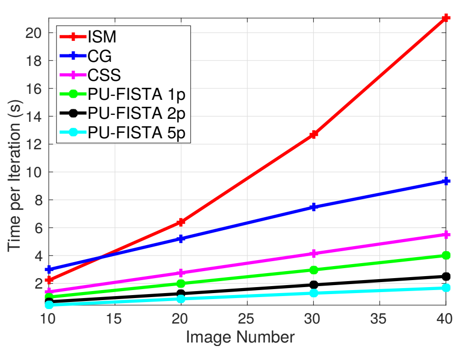



**Fig. 4**: A Comparison of execution time per iteration for the CDL methods for sets of 10, 20, 30 and 40 training images.

In Figure 4, we report mean runtime per iteration of the CDL methods during the learning with respect to the number of training images. Our proposed methods provides better runtime per iteration in comparison to the other ones. As additional information of the Figure 4, we present in Table 1 the values of runtime. In this table, we note that our proposed AGP algorithm (PU-FISTA-1P), when using 10 to 40 training images, is about 2.2 to 5.3 times faster than ISM, about 2.5 times faster than the CG; and about 1.5 times faster than CSS. The proposed update model applied in our algorithm provides an additional improvement of 1.6 and 2.5 times in the computational performance using 2 partitions and 5 partitions.

**Table 1**: Execution time (seconds) of the CDL methods for sets of 10 and 40 training images

| Image Set | ISM | GS | CSS | PU-FISTA 1P | PU-FISTA 2P | PU-FISTA 5P |
|---|---|---|---|---|---|---|
| 10 | 2222 | 2993 | 1475 | 1015 | 678 | 439 |
| 40 | 21051 | 9343 | 5866 | 4003 | 2504 | 1668 |

### 4.3. Simulations of denoising task

The Table 2 and 3 present the performance of the learned dictionary in terms PSNR, SSIM and sparsity metrics. In both tables, we observe that the dictionaries learned by our proposed algorithm yield equivalent results as the existing methods, since their differences in PSNR, SSIM and sparsity are negligible.

**Table 2**: Denoising of Barbara and Mandrill images corrupted with AWGN level $\sigma = 0.2$

| | **Mandrill** | | | **Barbara** | | |
|---|---|---|---|---|---|---|
| Dictionary ($12 \times 12 \times 32$) | PSNR | SSIM | Sparsity (%) | PSNR | SSIM | Sparsity (%) |
| ISM | 21.08 | 0.5286 | 7.46 | 23.15 | 0.6091 | 5.68 |
| GC | 21.08 | 0.5282 | 7.48 | 23.15 | 0.6091 | 5.69 |
| CSS | 21.09 | 0.5293 | 7.63 | 23.14 | 0.6082 | 5.73 |
| PU-FISTA 1p | 21.08 | 0.5293 | 7.43 | 23.15 | 0.6093 | 5.61 |
| PU-FISTA 2p | 21.08 | 0.5293 | 7.38 | 23.11 | 0.6084 | 5.56 |
| PU-FISTA 5p | 21.08 | 0.5280 | 7.38 | 23.08 | 0.6076 | 5.57 |

**Table 3**: Denoising of Lena and Peppers images corrupted with AWGN level $\sigma = 0.2$

| | **Lena** | | | **Peppers** | | |
|---|---|---|---|---|---|---|
| Dictionary ($12 \times 12 \times 32$) | PSNR | SSIM | Sparsity (%) | PSNR | SSIM | Sparsity (%) |
| ISM | 25.36 | 0.6818 | 1.58 | 25.17 | 0.6741 | 1.52 |
| GC | 25.35 | 0.6816 | 1.60 | 25.17 | 0.6739 | 1.51 |
| CSS | 25.34 | 0.6805 | 1.58 | 25.15 | 0.6738 | 1.51 |
| PU-FISTA 1p | 25.36 | 0.6829 | 1.55 | 25.17 | 0.6745 | 1.49 |
| PU-FISTA 2p | 25.37 | 0.6834 | 1.53 | 25.17 | 0.6749 | 1.46 |
| PU-FISTA 5p | 25.34 | 0.6825 | 1.57 | 25.15 | 0.6746 | 1.48 |

## 5. CONCLUSION

We have proposed a computationally efficient algorithm to solve the convolutional dictionary learning problem considering two complementary formulations. Our first contribution has consisted of an APG-based solution for both CDL subproblems that has proved to be significantly faster than state-of-the-art methods. The second contribution has been an update model, which has enabled to reduce the computations in our sparse coding update. The dictionaries learned by our proposed method have showed equivalent performance in terms of PSNR, SSIM and sparsity metrics as the existing methods in a denoising task while the training runtime has been improved substantially.

## 6. REFERENCES


[1] Julien Mairal, Francis Bach, and Jean 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.

[2] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[3] Michael Elad and Michal Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[4] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus, "Deconvolutional networks," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535.

[5] Brendt Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 301–315, Jan. 2016.

[6] Michal Ŝorel and Filip Ŝroubek, "Fast convolutional sparse coding using matrix inversion lemma," *Digital Signal Processing*, vol. 55, pp. 44 – 51, 2016.

[7] Cristina Garcia-Cardona and Brendt Wohlberg, "Subproblem coupling in convolutional dictionary learning," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, Beijing, China, Sept. 2017.

[8] Luigi Grippo and Marco Sciandrone, "On the convergence of the block nonlinear gauss–seidel method under convex constraints," *Operations research letters*, vol. 26, no. 3, pp. 127–136, 2000.

[9] Matthew D Zeiler, Graham W Taylor, and Rob Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2018–2025.

[10] Rakesh Chalasani, Jose C Principe, and Naveen Ramakrishnan, "A fast proximal method for convolutional sparse coding," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*. IEEE, 2013, pp. 1–5.

[11] Hilton Bristow, Anders Eriksson, and Simon Lucey, "Fast convolutional sparse coding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 391–398.

[12] Brendt Wohlberg, "Efficient convolutional sparse coding," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7173–7177.

[13] Kjersti Engan, Sven Ole Aase, and J Hakon Husoy, "Method of optimal directions for frame design," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*. IEEE, 1999, vol. 5, pp. 2443–2446.

[14] J. Liu, C. Garcia-Cardona, B. Wohlberg, and W. Yin, "Online Convolutional Dictionary Learning," *ArXiv e-prints*, June 2017.

[15] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni, "Online convolutional sparse coding," *arXiv preprint arXiv:1706.06972*, 2017.

[16] Cristina Garcia-Cardona and Brendt Wohlberg, "Convolutional dictionary learning," *arXiv preprint arXiv:1709.02893*, 2017.

[17] Gustavo Silva, "Partial update FISTA-based code (PU-FISTA)," *Matlab and Python library available from goo.gl/1rvfDq*, 2018.

[18] Mehiddin Al-Baali, "Descent property and global convergence of the fletcherreeves method with inexact line search," *IMA Journal of Numerical Analysis*, vol. 5, no. 1, pp. 121–124, 1985.

[19] Amir Beck and Marc Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[20] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.

[21] Syazni Shoid, Mohd Rivaie, Mustafa Mamat, and Zabidin Salleh, "A new conjugate gradient method with exact line search," *Applied Mathematical Sciences*, vol. 9, no. 96, pp. 4799–4812, 2015.

[22] James C Bezdek and Richard J Hathaway, "Some notes on alternating optimization," in *Advances in Soft Computing – AFSS 2002*, vol. 2275 of *Lecture Notes in Computer Science*, pp. 288–300. Springer, 2002.

[23] Brendt Wohlberg, "Sparse optimization research code (SPORCO)," *Matlab library available from goo.gl/BjVgH5*, 2017.