Dictionary Learning of Binary Atoms by using a Smooth Approximation

Edwin Vargas[1], and Henry Arguello[2],
[1]Department of Electrical Engineering, Universidad Industrial de Santander, Bucaramanga, Colombia
[2]Department of Computer Science, Universidad Industrial de Santander, Bucaramanga, Colombia

***Abstract*— The decomposition of a signal as a linear combination of few atoms of a learned dictionary has been widely studied for low and high-level tasks in signal and image processing applications. The atoms of the dictionary are typically assumed to be normalized, real-valued, and stored as floating-point numbers, which leads to high costs in storage and transmission time for large scale applications. In this work, we propose to learn binary atoms in order to represent an image sparsely. To solve this problem, we include a smoothing function for binarization and present an algorithm that iteratively alternates between a sparse coding update and a dictionary update. The binary structure allows to reduce the storage size of the dictionary as well as efficiently synthesize the underlying image using only addition and subtraction operations. Experiments on sparse representation of natural images show that the proposed binary dictionary gains up to 2 dB compared to binary dictionaries obtained using traditional binarization techniques.**

Sparsity, smooth approximation, dictionary learning, binary atoms.

I. Introduction

Unsupervised learning algorithms aim to discover the structure hidden in the data and to learn representations that are more suitable for image analysis systems [1], [2], [3]. Several unsupervised methods are based on reconstructing the input from the representation while constraining it to have specific desirable properties. Representations subject to be sparse and overcomplete [4], [5] have become one of the most widely used and successful models for inverse problems in signal processing, image processing, and computational imaging. The sparse-overcomplete representation model assumes that the signal of interest $\mathbf{x} \in \mathbb{R}^n$ can be decomposed as a linear combination of few atoms of a given redundant dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$ with $m > n$. Thus, the signal can be expressed as $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ contains the representation coefficients with only a few non-zero components.

The redundant dictionary can either be a predefined set of functions such as multiscale Gabor functions [6], [7], multiscale windowed ridgelets [8], wavelets or be learned to adapt to a set of training signals [5], [9]. Learned dictionaries have shown to improve low-level signal processing tasks such as image denoising [5], audio synthesis, as well as higher-level tasks such as image classification [10], showing that sparse learned models are properly adapted to natural signals. In [11], a reconstructive and discriminative dictionary is introduced for classification tasks. In [12], a discriminative method is proposed for various classification tasks, learning one dictionary per class; the classification process itself is based on the corresponding reconstruction error and does not exploit the actual decomposition coefficients.

Commonly, the atoms of the dictionary are assumed to be normalized, real values, and stored as floating-point numbers. For large scale applications, it leads to high costs in terms of storage and high transmission times if the dictionary has to be transmitted. For instance, a learned dictionary of 1024 atoms with size 16×16 requires 8M bytes (MB) for storage [9]. However, for higher dimensional signals as in light field photography [13], a learned overcomplete dictionary with 5000 atoms can have a memory footprint of 111 MB. In general, there are two major approaches to solve this problem. The first is to reduce the number of atoms [14], [15]. The second is to quantize the atoms [16] with the extreme case of the entries of the atoms being binary.

This work studies the extreme case of the second approach when the elements of the dictionary ($\mathbf{D}_{ij}$) are constrained to have binary values. The binary dictionary brings two main advantages to the dictionary learning problem. First, it reduces the memory usage and model size 32 times compared to the single-precision version. Second, the target signal can be efficiently synthesized computationally by only adding and subtracting the components of the sparse representation. To the best of our knowledge, there are no previous studies in the dictionary learning literature for the sparse representation of a real signal using binary atoms.

The binarization implies severe degradation precision in the recovery of the image. Therefore, this paper proposes a scheme for binarizing dictionaries, which aims to alleviate or even eliminate the accuracy degradation while still reducing the synthesis time, resource requirement, and power consumption. We introduce a smooth approximation function of the Sign function that allows employing gradient methods for the training of the dictionary that are restricted due to the non-differentiable nature of the Sign function. Specifically, this paper proposes a binary dictionary learning smoothing stochastic gradient method (BinDic) based on a special smooth function. BinDic is based on a smooth projected gradient method (SPG) [17], [18] which is useful for large scale non-smooth non-convex optimization problems on a closed convex set [19], [20]. The SPG algorithm solves the non-smoothness of the optimization problem by introducing a smoothing function, which approximates the original optimization function.

II. Dictionary Learning Problem

In this section, we formulate the dictionary learning problem based on a set of N examples. Consider the training set

$\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$. The dictionary learning process can be formulated as the following joint optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{D} \in \mathcal{D}, \mathbf{A}} & \|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \sum\limits_{i=1}^{N} r(\boldsymbol{\alpha}_i) \\ \text{subject to (s.t.)} & \mathcal{D} \subseteq \mathbb{R}^{n \times m}, \end{array} \tag{1}$$

where $\mathbf{D}$ is the dictionary and $\mathbf{A} = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_N] \in \mathbb{R}^{m \times N}$ is the sparse coefficient set for all the examples. Therein, $r(\cdot) : \mathbb{R}^m \to \mathbb{R}$ is a regularization function that promotes the sparse constraint for the coefficient vectors. For example, the l_1 norm $\|\cdot\|_1$ or the l_0 pseudonorm $\|\cdot\|_0$ [21], [22]. This problem seeks the best possible dictionary for the sparse representation of the example set $\mathbf{X}$. In this work, we propose to obtain a binary dictionary. One approach to obtain these values is to constrain the set $\mathcal{D}$ to have elements with only binary entries, e.g., $\{-1, 1\}$, which is known to be an expensive and combinatorial problem [23]. An alternative approach used in this work is to constrain $\mathcal{D}$ to be a convex real set and employ a function that promotes binary values. In the following section, we present a binarization function applied to the elements of the dictionaries and give some insights into how the dictionary update becomes more challenging.

Binarization function

In order to transform the real-valued variables into two values, one common approach is to use a binarization function. The most known approach is to use the sign function defined as:

$$\phi(z) = \text{Sign}(z) = \begin{cases} 1, & \text{if } z \geq 0, \\ -1, & \text{otherwise}, \end{cases} \tag{2}$$

where $\phi(z)$ is the binarized variable and z the real-valued variable. To simplify the notation we set ϕ to be the Sign function. Using this convention, the binary dictionary learning problem can be rewritten as

$$\begin{array}{ll} \min\limits_{\mathbf{D} \in \mathcal{D}, \mathbf{A}} & \|\mathbf{X} - \phi(\mathbf{D})\mathbf{A}\|_F^2 + \sum\limits_{i=1}^{N} r(\boldsymbol{\alpha}_i) \\ \text{subject to (s.t.)} & \mathcal{D} \subseteq \mathbb{R}^{n \times m}, \end{array} \tag{3}$$

where the function ϕ is applied component-wise.

The optimization problem (3) with respect to $\mathbf{D}$ (dictionary update) becomes more challenging due to the non-linear and non-smooth function $\phi(\cdot)$. The derivative of $\phi(\cdot)$ is zero in all the domain except in zero, making it incompatible with gradient descent methods for the optimization of the atoms of the dictionary. To overcome this limitation, this paper proposes an algorithm based on the Smoothing Projected Gradient (SPG) method [17] proposed by the minimization problem on a closed convex set, assuming that the objective function is locally Lipschitz continuous but non-convex, non-smooth. The SPG method introduces an auxiliary smoothing function $\varphi_\mu(\cdot)$ with parameter μ to approximate the original objective function, in order to solve the non-smooth and non-convex optimization problem.

III. Binary Dictionary Learning

An ideal objective function in dictionary learning would be to select the optimal dictionary for the underlying distribution of training samples

$$\mathbf{D}^\star \in \underset{\mathbf{D} \in \mathcal{D}}{\text{argmin}} \quad \mathbb{E}_{\mathbf{x} \sim \mathbb{P}} f_{\mathbf{x}}(\mathbf{D}), \tag{4}$$

where $\mathbb{P}$ is the distribution of the signal of interest and $f_{\mathbf{x}}(\mathbf{D})$ defines a loss function that should be small if $\mathbf{D}$ is good at representing the signal $\mathbf{x}$ in a sparse fashion. The loss function for the binary dictionary learning problem is defined as :

$$f_{\mathbf{x}}(\mathbf{D}) = \min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \mathcal{L}_{\mathbf{x}}(\mathbf{D}, \boldsymbol{\alpha}), \tag{5}$$

with

$$\mathcal{L}_{\mathbf{x}}(\mathbf{D}, \boldsymbol{\alpha}) = \|\mathbf{x} - \phi(\mathbf{D})\boldsymbol{\alpha}\|_2^2 + r(\boldsymbol{\alpha}). \tag{6}$$

However, in practical dictionary learning problems, given a finite set of samples $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_N]$ of the distribution $\mathbb{P}$, we optimize the empirical average cost function as follows

$$\min_{\mathbf{D} \in \mathcal{D}} f_N(\mathbf{D}) = \tfrac{1}{N} \sum_{i=1}^{N} f_{\mathbf{x}_i}(\mathbf{D}). \tag{7}$$

It should be noted that for our purposes, we assume that a minimizer $\boldsymbol{\alpha}^\star$ exists. There are several works providing algorithms that minimize the empirical cost function given by (7) with ϕ equal to the identity function as well as theoretical results and sample complexity for the dictionary learning problem [24], [25], [5], [26]. Recall, that this function is supposed to be applied component wise.

Since the objective function in (7) is non-smooth and non-convex, we employ an algorithm based on the smoothing projected gradient method (SPG) [18] to solve this problem. Combining smoothing techniques and the classical projected gradient method, the SPG method was proposed to solve problems where the objective is locally Lipschitz continuous but not necessarily convex and differentiable [18].

A. Smooth binarization function

The SPG is a smoothing function method that generalizes the PG method and its convergence for continuously differentiable optimization to non-smooth, non-convex optimization. As its name indicates, the key idea of the SPG method is to use a parametric smoothing approximation function with parameter μ in the PG method. More formally, the concept of smooth function is defined as follows [18]

Definition 1 (Smothing function:): Let $f : \mathcal{A} \to \mathbb{R}$ be a locally Lipschitz continuos function with $\mathcal{A} \in \mathbb{R}$ being an open set. Then $g : \mathcal{A} \times \mathbb{R}_{++} \to \mathbb{R}$ is a smoothing function of $f(\cdot)$, if $g(\cdot, \mu)$ is smooth in $\mathcal{A}$ for any fixed $\mu \in \mathbb{R}_{++}$ and

$$\lim_{\mu \downarrow 0} \; g(\mathbf{w}, \mu) = f(\mathbf{w}), \tag{8}$$

for any fixed $\mathbf{w} \in \mathbb{R}^n$.

Based on this definition we can construct a smoothing gradient method that exploits the rich theory and solution methods of optimization problems with continuously differentiable functions [27]. For instance, in [18], it is shown that any accumulation point generated by the SPG method globally converges to a stationary point associated with the smoothing function used in the method, which is also a stationary point of the non-smooth problem. Developing a smoothing method to solve (7) involves three main parts: 1) define a smoothing function, 2) choose an algorithm to solve the smooth problem and 3) update the smoothing parameter μ [18], [27]. Thus, in order to obtain a smooth approximation of the cost function f_N, we propose first to consider the function $\varphi_\mu(\cdot)$ with parameter μ defined below to be a smooth approximation of the function $\phi(\cdot)$ in (2)

$$\varphi_\mu(w) = \frac{w}{\sqrt{w^2+\mu^2}}, \tag{9}$$

where $\mu \in \mathbb{R}_{++}$. Fig. 1 shows the function (9) for different values of μ in the interval $[1, 10^{-6}]$.

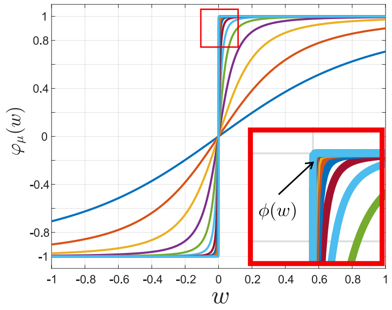

Fig. 1. Smooth approximation function φ_μ. Note that when $\mu \to 0$, $\varphi_\mu(w) \approx \phi(w)$

Based on the smoothing version of the Sign function we can construct a smooth approximation $g(\cdot,\mu)$ with parameter μ of the cost function f_N in (7) which leads to the following binary smooth optimization problem

$$\min_{\mathbf{D}\in\mathcal{D}} \quad g(\mathbf{D},\mu) = \tfrac{1}{N}\sum_{i=1}^{N} g_{\mathbf{x}_i}(\mathbf{D},\mu), \tag{10}$$

with

$$g_{\mathbf{x}_i}(\mathbf{D},\mu) = \inf_{\boldsymbol{\alpha}\in\mathbb{R}^m} \|\mathbf{x}_i - \varphi_\mu(\mathbf{D})\boldsymbol{\alpha}_i\|_2^2 + r(\boldsymbol{\alpha}_i). \tag{11}$$

The cost function g can be rewritten in short form as $g(\mathbf{D},\mu) = \inf_{\mathbf{A}\in\mathbb{R}^{m\times N}} g_{\mathbf{X}}(\mathbf{D},\mu)$ with

$$g_{\mathbf{X}}(\mathbf{D},\mu) = \frac{1}{2N}\|\mathbf{X} - \varphi_\mu(\mathbf{D})\mathbf{A}\|_F^2 + R(\mathbf{A}) \tag{12}$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_N]$ and $R(\mathbf{A}) = \sum_{i=1}^{N} r(\boldsymbol{\alpha}_i)$.

The following Theorem shows that the function $g(\cdot)$ is a uniformly smooth approximation of the function $f_N(\cdot)$.

Theorem 1: Let f_N and $\varphi_\mu(\cdot)$ be as defined in (7) and (9), respectively. Then $g(\cdot,\mu)$ in (10) is smooth for any fixed $\mu > 0$, and there exists a constant $k_1 > 0$ satisfying

$$|g(\mathbf{x},\mu) - f_N(\mathbf{x})| \leq \mu k_1. \tag{13}$$

IV. Binary Dictionary Learning Method

This section presents the optimization algorithm termed BDL-SSG that is proposed to solve (10). Note that this non-convex problem is a joint optimization problem with respect to $\mathbf{D}$ and $\mathbf{A}$. A natural strategy to solve this problem is to alternate between the two variables, minimizing over one while keeping the other one fixed. The optimization problem with respect to $\mathbf{A}$ is convex, while the problem with respect to $\mathbf{D}$ is not convex due to the binarization function φ_μ. Thus, for the dictionary update, we use a smoothing stochastic gradient method. A sketch of the proposed strategy is detailed in Algorithm 1. The initialization of the algorithm and the optimization steps with respect to $\mathbf{A}$ and $\mathbf{D}$ are detailed in the following sections.

Algorithm 1 Proposed Binary Dictionary Learning

```
for $t = 1$ to stopping rule do
    $\mathbf{A}^{(t)} = g(\mathbf{D}^{(t-1)},\mu) + R(\mathbf{A})$          ▷ Algorithm 2
    $\mathbf{D}^{(t)} = \operatorname{argmin}_{\mathbf{D}\in\mathcal{D}} \; g(\mathbf{D},\mu)$          ▷ Algorithm 3
return: $\mathbf{D}, \mathbf{A}$
```

A. Optimization with respect to A

The first step of the minimization problem optimizes the cost function with respect to $\mathbf{A}$ for a fixed dictionary $\mathbf{D}$ using the ADMM algorithm. An auxiliary variable is introduced to split the objective function and the constraints leading to

$$\begin{aligned} \operatorname*{argmin}_{\mathbf{A},\mathbf{V}} \quad & h(\mathbf{A},\mathbf{V}) = \|\mathbf{X} - \varphi_\mu(\mathbf{D})\mathbf{A}\|_F^2 + \sum_{i=1}^{N} R(\mathbf{v}_i) \\ \text{s.t.} \quad & \mathbf{A} = \mathbf{V}. \end{aligned} \tag{14}$$

The augmented Lagrangian associated with (14) is

$$\mathcal{L}(\mathbf{A},\mathbf{V},\mathbf{G}) = h(\mathbf{A},\mathbf{V}) + \tfrac{\rho}{2}\|\mathbf{A} - \mathbf{V} + \mathbf{G}\|_2^2, \tag{15}$$

where $\mathbf{G}$ is the scaled dual variable and $\rho \geq 0$ is weighting the augmented Lagrangian term. In this work we set the regularization function as $r(\cdot) = \lambda\|\cdot\|_1$. The exact procedure used for estimating $\mathbf{A}$ is summarized in Algorithm (2).

Algorithm 2 ADMM algorithm to estimate $\mathbf{A}$

```
$\mathbf{V}^{(0)}, \mathbf{G}^{(0)}$
for $k = 1$ to stopping rule do
    $\mathbf{A}^{(k+1)} = \operatorname{argmin}_{\mathbf{A}} \; \mathcal{L}(\mathbf{A},\mathbf{V}^{(k)},\mathbf{G}^{(k)})$
    $\mathbf{V}^{(k+1)} = \operatorname{argmin}_{\mathbf{V}} \; \mathcal{L}(\mathbf{A}^{(k+1)},\mathbf{V},\mathbf{G}^{(k)})$
    $\mathbf{G}^{(k+1)} = \mathbf{G}^{(k)} + \mathbf{V}^{(k+1)} - \mathbf{A}^{(k+1)}$
return: $\mathbf{A}^{k+1}$
```

B. Optimization with respect to D

In this section we introduce a smoothing gradient method to solve (10). Note that we assume that a minimizer $\hat{\mathbf{A}}$ for the sparse representation exists. Therefore, the cost function can be rewritten as $g(\mathbf{D},\mu) = \frac{1}{2N}\left\|\mathbf{X} - \varphi_\mu(\mathbf{D})\hat{\mathbf{A}}\right\|_F^2$. The standard gradient algorithm is defined as

$$\mathbf{D}^{k+1} = \mathbf{D}^k - \beta\frac{\partial g(\mathbf{D}^k,\mu_k)}{\partial \mathbf{D}}, \tag{16}$$

where β is the step size. Here we propose a block stochastic gradient method in order to alleviate the computational complexity when N is large. Instead of calculating the whole gradient ∂g, we choose only a random subset $\mathbf{H}_{\Gamma_k}$ (stochastic gradient) of the sum of each iteration k.

Algorithm 3 BinDic: Binary Dictionary learning smoothing stochastic gradient

```
Input: Data $\{\mathbf{x}_i\}_{i=1}^N$ and $\epsilon_0 = 10^{-10}$. Choose constants $\gamma_1 = 0.5, \mu_0 = 1, \gamma = 0.01$ and tolerance $\epsilon$.
Initial dictionary $\mathbf{D}_0$
while $\|\mathbf{H}_{\Gamma_k}\|_2 \geq \epsilon$ do Choose $\Gamma_k$ uniformly at random from the subsets of $\{1, \cdots, N\}$
    $\mathbf{D}^{k+1} = \mathbf{D}^k - \beta\mathbf{H}_{\Gamma_k}$
    if $\|\mathbf{H}_{\Gamma_k}\|_2 \geq \gamma\mu_k$ then
        $\mu_{k+1} = \mu_k$
    else
        $\mu_{k+1} = \gamma_1\mu_k$
return: $\mathbf{D}_{k+1} = \hat{\mathbf{D}}$
```

V. Simulation Results

We carried out experiments on natural image data to illustrate the practicality of the proposed algorithm and the general sparse coding framework in a binary dictionary. We should note that our tests come only to prove the concept of using such dictionaries with sparse representations of natural images. However, this algorithm can be applied to other types of signals.

A. Simulation scenario and parameters

The training data were constructed as a set of $12,288$ examples of block patches of size 8×8 pixels of 7 different natural images, taken from an image database (in various no overlapping locations). Additionally, following common practice (see e.g. [28], [29]), the binary sparse coding/dictionary learning algorithms are also applied to test/training images after a high pass filtering preprocessing step. We used the following parameters for algorithm 3: $\gamma_1 = 0.5$, $\gamma = 0.01$, $\beta = 0.5$, $\mu_0 = 1$, and $\epsilon = 1 \times 10^{-6}$. The regularization parameter for the estimation of the sparse representation was fixed to $\lambda = 0.001$.

B. Sparse representation using binary dictionaries

In this section we evaluated the proposed learned dictionary for sparse representation of natural images. We compare the trained dictionary with a full precision dictionary trained using K-SVD [5], and a binarization of the K-SVD dictionary using the Sign function (bin-KSVD). The number of atoms is equal to $m = 4n$. We also compare with the analytical Hadamard transform with $m = n$.

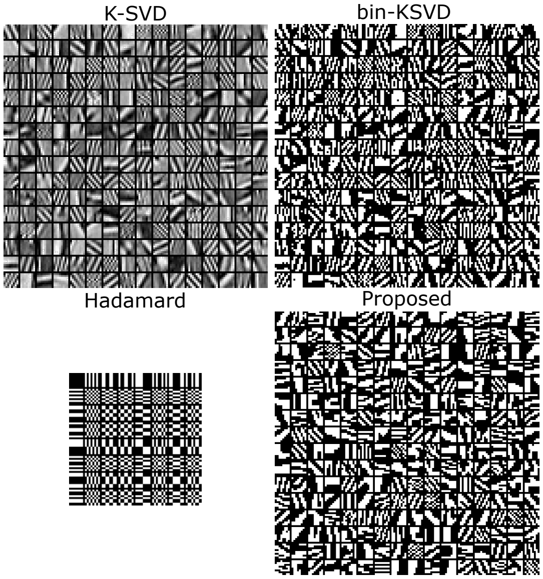

Fig. 2. Learned dictionaries using K-SVD algorithm, the proposed algorithm, binarization of the K-SVD (bin-KSVD), and Hadamard dictionary.

First, the capability of the sparse representation of the proposed dictionary on three different high frequency images is tested. The learned dictionaries and the Hadamard dictionary are presented in Fig. 2. The patches from the test images are not included in the training set. The number of non-zero coefficients of each non-overlapping patch of the images is 6. Quantitative results are presented in Table I. Numerically, we can see that the proposed binary dictionary presents results with lower degradation accuracy compared with bin-KSVD. Furthermore, in some cases, the results using the proposed dictionary are very competitive with the full precision case having the advantage of only using binary atoms.

TABLE I
Performance of DL methods: PSNR (dB) and SSIM

Methods	Metric	HOUSE	BOAT	ELAINE
K-SVD	PSNR	**30.93**	**29.71**	24.26
	SSIM	**0.95**	**0.89**	0.85
bin-KSVD	PSNR	27.62	27.87	23.81
	SSIM	0.91	0.84	0.83
Hadamard	PSNR	26.33	26.78	22.74
	SSIM	0.52	0.54	0.46
Proposed	PSNR	29.63	28.84	**24.51**
	SSIM	0.94	0.87	**0.87**

VI. Conclusion

In this work, we have proposed an algorithm for training a binary dictionary for the overcomplete sparse representation of natural signals. Using a smooth approximation of the Sign function, the proposed algorithm solves a smooth problem associated with the non-smooth non-convex binary dictionary learning problem. Numerical results showed that the proposed binary dictionary has lower degradation accuracy for the reconstruction of natural signals compared with a traditional binarization scheme. Furthermore, in some cases the reconstruction using the binary dictionary is competitive with the full precision dictionary obtained using the K-SVD algorithm.

References

[1] T. D. Sanger, “Optimal unsupervised learning in a single-layer linear feedforward neural network,” *Neural networks*, vol. 2, no. 6, pp. 459–473, 1989.

[2] P. Baldi, “Autoencoders, unsupervised learning, and deep architectures,” in *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 37–49, 2012.

[3] J. Yang, D. Parikh, and D. Batra, “Joint unsupervised learning of deep representations and image clusters,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5147–5156, 2016.

[4] M. S. Lewicki and T. J. Sejnowski, “Learning overcomplete representations,” *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[5] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[6] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, 1993.

[7] S. Qian and D. Chen, “Signal representation using adaptive normalized gaussian functions,” *Signal processing*, vol. 36, no. 1, pp. 1–11, 1994.

[8] J.-L. Starck, E. J. Candès, and D. L. Donoho, “The curvelet transform for image denoising,” *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, 2002.

[9] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online dictionary learning for sparse coding,” in *Proceedings of the 26th annual international conference on machine learning*, pp. 689–696, ACM, 2009.

[10] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach, “Supervised dictionary learning,” in *Advances in neural information processing systems*, pp. 1033–1040, 2009.

[11] Z. Jiang, Z. Lin, and L. S. Davis, “Label consistent k-svd: Learning a discriminative dictionary for recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, 2013.

[12] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Discriminative learned dictionaries for local image analysis,” in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.

[13] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, “Compressive light field photography using overcomplete dictionaries and optimized projections,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, p. 46, 2013.

[14] M. Marsousi, K. Abhari, P. Babyn, and J. Alirezaie, “An adaptive approach to learn overcomplete dictionaries with efficient numbers of elements,” *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3272–3283, 2014.

[15] R. Mazhar and P. D. Gader, “EK-SVD: Optimized dictionary design for sparse representations,” in *2008 19th International Conference on Pattern Recognition*, pp. 1–4, IEEE, 2008.

[16] L. Liu, J. Liang, Y. Zhao, C. Lin, and H. Bai, “Quantized dictionary for sparse representation,” in *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, IEEE, 2015.

[17] C. Zhang and X. Chen, “Smoothing projected gradient method and its application to stochastic linear complementarity problems,” *SIAM Journal on Optimization*, vol. 20, no. 2, pp. 627–649, 2009.

[18] X. Chen and W. Zhou, “Smoothing nonlinear conjugate gradient method for image restoration using nonsmooth nonconvex minimization,” *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 765–790, 2010.

[19] S. Pinilla, J. Bacca, and H. Arguello, “Phase retrieval algorithm via nonconvex minimization using a smoothing function,” *IEEE Trans. Signal Process.*, vol. 66, no. 17, pp. 4574–4584, 2018.

[20] M. Nikolova, M. K. Ng, S. Zhang, and W.-K. Ching, “Efficient reconstruction of piecewise constant images using nonsmooth nonconvex minimization,” *SIAM journal on Imaging Sciences*, vol. 1, no. 1, pp. 2–25, 2008.

[21] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM review*, vol. 43, no. 1, pp. 129–159, 2001.

[22] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Proceedings of 27th Asilomar conference on signals, systems and computers*, pp. 40–44, IEEE, 1993.

[23] K. G. Murty and S. N. Kabadi, “Some np-complete problems in quadratic and nonlinear programming,” 1985.

[24] D. Vainsencher, S. Mannor, and A. M. Bruckstein, “The sample complexity of dictionary learning,” *Journal of Machine Learning Research*, vol. 12, no. Nov, pp. 3259–3281, 2011.

[25] R. Gribonval, R. Jenatton, F. Bach, M. Kleinsteuber, and M. Seibert, “Sample complexity of dictionary learning and other matrix factorizations,” *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3469–3486, 2015.

[26] K. Engan, S. O. Aase, and J. H. Husøy, “Multi-frame compression: Theory and design,” *Signal Processing*, vol. 80, no. 10, pp. 2121–2140, 2000.

[27] X. Chen, “Smoothing methods for nonsmooth, nonconvex minimization,” *Mathematical programming*, vol. 134, no. 1, pp. 71–99, 2012.

[28] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun, “Learning convolutional feature hierarchies for visual recognition,” in *Advances in neural information processing systems*, pp. 1090–1098, 2010.

[29] M. D. Zeiler, G. W. Taylor, R. Fergus, *et al.*, “Adaptive deconvolutional networks for mid and high level feature learning.,” in *ICCV*, vol. 1, p. 6, 2011.