A Robust Dictionary Learning Algorithm for Image Denoising
Subhadip Mukherjee, Rupam Basu, and Chandra Sekhar Seelamantula, Senior Member, IEEE

Abstract—In many image processing applications, we encounter the problem of suppressing noise that obeys a non-Gaussian statistics. It is well known in the signal processing community that, in case of heavy-tailed non-Gaussian noise corruption, $\ell_1$ distortion is a more suitable metric for data fidelity. Dictionary based image denoising algorithms existing in the literature are typically aimed at minimizing the $\ell_2$ distortion metric, and hence not suitable for suppressing non-Gaussian or impulsive noise. In this paper, we develop a dictionary learning algorithm by minimizing the $\ell_1$ error. The proposed algorithm exploits the idea of iterative minimization of suitably weighted $\ell_2$ error. We refer to this algorithm as robust dictionary learning (RDL). We demonstrate that the proposed algorithm results in higher atom detection rate compared to the state-of-the-art $K$-SVD algorithm, both in case of Gaussian and non-Gaussian noise contamination. For real images, we observe clear superiority of the proposed RDL algorithm over its $\ell_2$ based counterpart, namely the $K$-SVD algorithm, in terms of the quality of denoised output.

Index Terms—Noise robustness, heavy-tailed noise, dictionary learning, iteratively re-weighted least squares, $K$-SVD algorithm.

I. INTRODUCTION

IMAGE denoising has been an area of research for several decades and continues to be a challenging task with advances in imaging modalities. Recently, dictionary based algorithms have been attempted to solve the problem of denoising effectively. The principal idea behind such approaches is to train a dictionary that is capable of representing image patches parsimoniously, to be more precise, using a sparse representation. In many applications, the assumption of Gaussian-ness of noise may not hold in reality, thus rendering the $\ell_2$ minimization based algorithms inapplicable. Moreover, these algorithms can lead to over-smoothing of the image, causing serious loss of details in the images. One way to alleviate this problem is to devise dictionary training algorithms tailored to minimizing the $\ell_1$ data error, which is the main contribution of this work. We develop an algorithm that approximates the solution to the $\ell_1$ minimization problem by iteratively solving weighted $\ell_2$ problems. This idea is popularly known as iteratively re-weighted least squares (IRLS) [1] in the compressed sensing literature. We refer to the algorithm as the RDL algorithm. Before elaborating on the proposed technique, we give a brief overview of recent works on dictionary based approaches for image processing applications.

Typically, the problem of dictionary learning is formulated as an optimization problem, where the cost function trades off the data consistency term with the sparsity promoting regularization. Standard metrics used for the data term and the regularization are the $\ell_2$ and $\ell_1$ metrics, respectively. Nevertheless, algorithms for minimizing $\ell_1$ based data terms have been proposed in the context of non-negative matrix factorization (NMF). An online $\ell_1$ based NMF algorithm was developed by Kasiviswanathan et al. [10] for document detection. They imposed additional constraint of non-negativity on the dictionary and the corresponding coefficients, to maintain the interpretability of the result.

Some of the initial contributions to the problem of data-adaptive dictionary learning were made by Aharon et al. [5], [6]. They proposed the $K$-SVD algorithm, which is by far the most popular and successful algorithm for dictionary design. They have successfully deployed the $K$-SVD algorithm to solve the problem of image denoising [6], specifically to eliminate zero-mean additive white Gaussian noise. They proposed two training options: (i) training on a corpus of clean image patches and (ii) training on the noisy input image itself. Since the $K$-SVD algorithm is not suitable for handling large image patches owing to large computational requirements, addition of a global prior was proposed to enforce sparsity. The $K$-SVD algorithm uses the $\ell_2$ distortion metric as a measure of data fidelity.

Dictionary-based approaches find applications in many image processing tasks other than denoising, for example, super-resolution, inpainting, etc. In the context of image super-resolution, Yang et al. [7] proposed an algorithm, which exploits the idea of representing an image patch as a sparse linear combination of elements from an appropriately chosen over-complete dictionary. They proposed a coupled learning strategy, where one tries to learn the common sparse representation of the low-resolution and high-resolution patches jointly and updates the corresponding dictionaries accordingly. Sahoo et al. [8] addressed the problem of image inpainting using sparse representation of local image patches, where an adaptive window is selected for local sparse representation, which affects the global recovery of the image. The problem of image restoration such as color image denoising, demosaicing and inpainting, has been addressed by Mairal et al. [9], who applied the $K$-SVD algorithm for handling non homogeneous noise and missing information.

The authors are with the Department of Electrical Engineering, Indian Institute of Science, Bangalore 560012, India. Emails: subhadip@ee.iisc.ernet.in, rbasu28@ece.iisc.ernet.in, chandra.seelamantula@ieee.org. The authors would like to thank Dr. K. N. Chaudhury for his insightful comments and suggestions in preparing the manuscript.
B. Contributions of this paper

In this paper, we propose an algorithm for dictionary learning aimed at minimizing the \( \ell_1 \) distortion metric on the data term. The proposed RDL algorithm minimizes the \( \ell_1 \) error using the idea of IRLS, which essentially approximates the solution of the \( \ell_1 \) minimization problem by iteratively solving a series of \( \ell_2 \) minimization problems. The use of \( \ell_1 \) metric for data fidelity results in noise robustness in terms of training performance and proves to be efficient for removal of impulsive noise from images while preserving the edges and texture, as demonstrated by the experimental results.

II. ROBUST DICTIONARY LEARNING (RDL) ALGORITHM

Let us denote the training dataset by \( T \), containing \( N \) training examples \( \{y_n\}_{n=1}^N \) in \( \mathbb{R}^m \) corrupted by additive noise. The objective is to learn a dictionary \( D \) containing \( K \) atoms, tailored to the dataset \( T \), such that \( D \) represents all the examples in \( T \) using sparse coefficient vectors \( \{x_n\}_{n=1}^N \). Typically, \( D \) is over-complete, meaning that \( m < K \). The symbols \( Y \) and \( X \) denote the matrices constructed by stacking \( y_n \)'s and \( x_n \)'s in the columns. In order to achieve robustness to noise, we propose to solve the following optimization problem:

\[
\min_{D, x_n} \sum_{n=1}^N \|y_n - Dx_n\|_1 + \frac{1}{2} \sum_{n=1}^N \lambda_n \|x_n\|_1.
\]

(1)

The use of \( \ell_1 \) metric on the data term is the key to noise robustness. The cost function in (1) can be interpreted from a Bayesian perspective, where a Laplacian model is assumed for the prior as well as for the additive noise. The parameters \( \lambda_n \) in the regularization term adjusts the sparsity of the resulting solution. Typically, one begins the iterations by setting \( W_{1n}^{(0)} \) and \( W_{2n}^{(0)} \) to diagonal matrices of appropriate dimensions. Typically, one begins the iterations by setting \( W_{1n}^{(0)} \) and \( W_{2n}^{(0)} \) to identity matrices, and they are updated in the \((k + 1)^{st}\) iteration as

\[
W_{1n}^{(k+1)}(j) = \frac{1}{\|y_n - Dx_n^{(k)}\|_1 + \epsilon} \\
W_{2n}^{(k+1)}(j) = \frac{1}{\|x_n^{(k)}\|_1 + \epsilon},
\]

where \( W(j) \) denotes the \( j^{th} \) diagonal entry of a diagonal matrix \( W \), and \( (x)^j \) denotes the \( j^{th} \) entry of a vector \( x \). A small positive constant \( \epsilon \) is added to the denominator to ensure stability of the solution.

One can also choose to work with an equivalent constrained version of (1), which is of the form

\[
\min_{x_n} \|y_n - Dx_n\|_1 \text{ subject to } \|x_n\|_1 \leq \tau_n.
\]

(3)

The values of \( \tau_n \) control the sparsity of the resulting solution, smaller values of \( \tau_n \) resulting in higher sparsity. Efficient convex optimization packages, for example CVX, are available to solve (3). To further reduce the sensitivity of the solution to noise, we propose to apply an appropriate threshold on the small-magnitude entries of \( X \), following sparse coding. Thus the proposed approach can be interpreted as \( \ell_1 \) minimization followed by pruning.

B. Updating the dictionary

Similar to the K-SVD algorithm, we adopt a sequential updating strategy for the dictionary atoms \( \{d_j\}_{j=1}^K \). To simultaneously update the \( j^{th} \) atom in the dictionary and the corresponding row in \( X \), we focus on the error matrix \( E_j = Y - \sum_{i \neq j} d_i x_i^T \), resulting from the absence of \( d_j \), where \( x_i^T \) denotes the \( i^{th} \) row of \( X \). In order to obtain a refined estimate of \( d_j \) and \( x_i \), one is required to solve

\[
\min_{d_j, x_i} \|E_j - d_j x_i^T\|_1 \text{ subject to } S(x_i^T) = \Omega_j \text{ and } \|d_j\|_2 = 1.
\]

(4)

where \( \|A\|_1 \) denotes the sum of the absolute values of the entries of matrix \( A \). We seek to minimize the cost function subject to the constraint that \( S(x_i^T) = \Omega_j \), where \( S(\cdot) \) denotes the support operator. Therefore, it suffices to consider only those columns of \( E_j \) whose indices are in the support set \( \Omega_j \), which is denoted by \( E_j \mid_{\Omega_j} \). Furthermore, to get rid of scale ambiguity, we restrict \( D \) to have unit-length atoms, that is, \( \|d_i\|_2 = 1 \) for all \( i \). As a consequence, solving (4) is equivalent to solving an optimization problem of the form

\[
\min_{u,v} \|E - uv^T\|_1 = \min_{u,v} \sum_{n=1}^M \|e_n - v_n u\|_1 \\
\text{subject to } \|u\|_2 = 1,
\]

(5)

where \( e_n \) is the \( n^{th} \) column of \( E \) and \( v_n \) denotes the \( n^{th} \) entry of the vector \( v \). The key idea behind the RDL approach is to approximate the \( \ell_1 \)-norm using a suitably re-weighted \( \ell_2 \) metric in the following manner:

\[
\sum_{n=1}^M \|e_n - v_n u\|_1 \approx \sum_{n=1}^M (e_n - v_n u)^T W_n (e_n - v_n u).
\]
Differentiating with respect to \( u \) and setting to zero, we get
\[
\mathbf{u} = \left( \sum_{n=1}^{M} \nu_n^2 \mathbf{W}_n \right)^{-1} \left( \sum_{n=1}^{M} \nu_n \mathbf{W}_n \mathbf{e}_n \right).
\]

Similarly, for a fixed \( \mathbf{u} \), we have \( \nu_n = \frac{\mathbf{u}^T \mathbf{W}_n \mathbf{e}_n}{\mathbf{u}^T \mathbf{W}_n \mathbf{u}} \). Once the update of \( \mathbf{u} \) and \( \mathbf{v} \) is over, the \( j \)th diagonal element of \( \mathbf{W} \) should be updated as
\[
w_n(j) \leftarrow \frac{1}{\langle \mathbf{e}_n - \nu_n \mathbf{u} \rangle}.
\]

Dictionary update is performed by setting \( \mathbf{d}_j = \mathbf{u}_0 \) and \( \mathbf{x}_j = \mathbf{v}_0^T \), where \((\mathbf{u}_0, \mathbf{v}_0)\) solves (5). The step-by-step description to find the solution of (5) is given in Algorithm 1.

**Algorithm 1 (RDL):** To find \( \mathbf{u} \) and \( \mathbf{v} \) such that \( \| \mathbf{E} - \mathbf{u} \mathbf{v}^T \|_1 \) is minimized, subject to \( \| \mathbf{u} \|_2 = 1 \).

1. **Input:** Error matrix \( \mathbf{E} \), number of iterations \( J \).
2. **Initialization:**
   - Set the iteration counter \( p \) to zero.
   - Set \( \mathbf{u}^{(p)} \leftarrow \mathbf{a} \) and \( \mathbf{v}^{(p)} \leftarrow \sigma \mathbf{b} \), where \( \mathbf{a} \) and \( \mathbf{b} \) are the left and right singular vectors, respectively, corresponding to the largest singular value \( \sigma \) of \( \mathbf{E} \).
   - Initialize the weight matrices \( \mathbf{W}_n^{(p)} \leftarrow \mathbf{I} \), for \( n = 1, 2, \ldots, M \), where \( \mathbf{I} \) denotes the identity matrix and \( M \) is the number of columns in \( \mathbf{E} \).
3. **Repeat the following steps \( J \) times:**
   - \( w_n^{(p+1)}(j) \leftarrow \frac{1}{\langle \mathbf{e}_n - \nu_n^{(p)} \mathbf{u}^{(p)} \rangle} \),
   - \( \mathbf{u}^{(p+1)} \leftarrow \left[ \sum_{n=1}^{M} \left( \nu_n^{(p)} \right)^2 \mathbf{W}_n^{(p+1)} \right]^{-1} \left[ \sum_{n=1}^{M} \nu_n^{(p)} \mathbf{W}_n^{(p+1)} \mathbf{e}_n \right] \),
   - \( \nu_n^{(p+1)} \leftarrow \frac{\left( \mathbf{u}^{(p+1)} \right)^T \mathbf{W}_n \mathbf{e}_n}{\left( \mathbf{u}^{(p+1)} \right)^T \mathbf{W}_n \left( \mathbf{u}^{(p+1)} \right)} \), for \( n = 1, 2, \ldots, M \).
   - \( p \leftarrow p + 1 \).
4. **Scaling:** Set \( \mathbf{u} \leftarrow \frac{\mathbf{u}^{(p)}}{\| \mathbf{u}^{(p)} \|_2} \) and \( \mathbf{v} \leftarrow \| \mathbf{u}^{(p)} \|_2 \mathbf{v}^{(p)} \)
5. **Output:** Solution \((\mathbf{u}, \mathbf{v})\) to (5).

The RDL algorithm is computationally more expensive than the \( K \)-SVD algorithm, which requires computing the singular value decomposition (SVD) of the error matrix to update each atom of \( D \). The RDL algorithm starts with the \( K \)-SVD update and keeps refining the estimate by solving a series of weighted \( \ell_2 \) minimization problems. However, the quality of the solution obtained using RDL is better than that obtained using \( K \)-SVD, in the case where the training set is noise contaminated, as indicated by the experimental results in the following section.

### III. Experimental Results

#### A. Synthesized signal experiment

To validate the RDL algorithm, we conduct experiments on the synthesized training sets, for Gaussian as well as Laplacian noise. Since the ground-truth dictionary \( D \) is known in the experiments, we use two performance metrics, namely the atom detection rate (ADR) and distance \( \kappa(D, \hat{D}) \) of the estimated dictionary \( \hat{D} \) from the ground-truth \( D \). To compute ADR, we count the number of recovered atoms in \( D \). An atom \( \mathbf{d}_i \) in \( D \) is considered to be recovered if \( \| \mathbf{d}_i^T \mathbf{d}_j \|_2 \) exceeds 0.99 for some \( j \). Since \( D \) and \( \hat{D} \) contain unit length atoms, this criterion ensures a near-accurate atom recovery. Finally, to compute ADR, we take the ratio of the number of recovered atoms to the total number of atoms. The metric \( \kappa \) measures the closeness of \( \hat{D} \) to \( D \), and is defined as
\[
\kappa = \frac{1}{K} \sum_{k=1}^{K} \min_{j} \left( 1 - \| \mathbf{d}_k^T \mathbf{d}_j \|_2 \right).
\]

The value of \( \kappa \) satisfies \( 0 < \kappa < 1 \). If \( \kappa \) is close to 0, it indicates that the recovered dictionary closely matches the ground-truth. The ground-truth dictionary is created by randomly generating \( K = 20 \) vectors as the atoms, with unit length, in \( m = 16 \) dimensional space. Each column of the coefficient matrix \( X \) has \( s = 3 \) non-zeros, with their locations chosen uniformly at random and values drawn from the Gaussian distribution with zero mean and unity variance. Subsequently, \( D \) is combined with \( X \) and contaminated by additive noise, to generate \( N = 800 \) samples. The variation of \( \kappa \) and ADR with iterations, averaged over 25 independent trials, is demonstrated in Figure 1. To facilitate fair comparison of the RDL with the \( K \)-SVD algorithm, true values of the \( \ell_1 \) and \( \ell_0 \) norms of the ground-truth coefficient vectors, that is, true values of \( \tau_{\mathbf{x}s} \) in (3) and the parameter \( s \) are supplied to the algorithms, respectively. The RDL algorithm, as can be seen from Figure 1, results in a conspicuously superior performance over its \( \ell_2 \)-based counterpart \( K \)-SVD, both in terms of \( \kappa \) and ADR. One can observe similar robustness of the RDL algorithm to both Gaussian and Laplacian noise from the experimental results. The ensemble-averaged (over
Details of the algorithms are described below separately for Barbara image, which contains some textures and patterns. One with the Tower extracted from the noisy input image, with an overlap of one.

B. Image denoising

In image denoising experiments, patches of size \(4 \times 4\) are extracted from the noisy input image, with an overlap of one pixel in both horizontal and vertical directions. Subsequently, the noisy patches are vectorized, and stacked in columns of the data matrix \(Y\). We conduct two image denoising experiments: one with the Tower image corrupted by heavy-tailed Laplacian noise and the other one with Gaussian noise contaminated Barbara image, which contains some textures and patterns. Details of the algorithms are described below separately for Gaussian and non-Gaussian noise.

1) Gaussian noise: Since \(\ell_2\) metric is more suitable for suppressing additive Gaussian noise over smooth regions, we first classify different parts of the image into smooth and non-smooth regions. The Canny edge detection algorithm is used on a smoothed version of the raw input image, obtained using a simple wavelet thresholding algorithm. Subsequently, at each pixel location, we extract a patch of size \(4 \times 4\) and classify them in one of two categories, namely ‘smooth patch’ and ‘edge patch’. Finally we construct two training matrices, one for each kind. To denoise the smooth patches, we run the usual \(\ell_2\) based \(K\)-SVD algorithm, whereas on the edge patches, the RDL algorithm is deployed. The output image is obtained by computing the average of the resulting denoised patches, the RDL algorithm is deployed. The output image is denoised and the corresponding values are obtained by averaging over 25 realizations. Number of iterations is 20.

### Table I

| Input SNR (dB) | 10    | 20    | 30    | 40    | 50    |
|---------------|-------|-------|-------|-------|-------|
| Distance from ground-truth (\(\kappa\)) | 0.0190 | 0.0351 | 0.0086 | 0.0200 | 0.0054 |
| ADR           | 0.8700 | 0.6100 | 0.9600 | 0.7700 | 0.9600 |

(COLOR ONLINE) Performance comparison of RDL and \(K\)-SVD on synthetic dataset in case of additive Laplacian noise for different values of input SNR: In each cell, the left (in black) and right (in red) entries correspond to the RDL and \(K\)-SVD algorithms, respectively. The values are obtained by averaging over 25 realizations. Number of iterations is 20.

### Table II

| Input SNR (dB) | 10    | 20    | 30    | 40    | 50    |
|---------------|-------|-------|-------|-------|-------|
| Distance from ground-truth (\(\kappa\)) | 0.1656 | 0.2142 | 0.0103 | 0.0121 | 0.1452 |
| ADR           | 0.8350 | 0.5200 | 0.9000 | 0.8000 | 0.9536 |

(COLOR ONLINE) Performance comparison of RDL and \(K\)-SVD on synthetic dataset in case of Gaussian noise corruption for different values of input SNR: In each cell, the left (in black) and right (in red) entries correspond to the RDL and \(K\)-SVD algorithms, respectively. The values are obtained by averaging over 25 realizations. Number of iterations is 20.

### Table III

|       | Lenna | Barbara | House | Tower | Boat | Mandrill | Cameraman | Parrot |
|-------|-------|---------|-------|-------|------|----------|-----------|--------|
| \(K\)-SVD | 27.74 | 27.75 | 28.92 | 28.86 | 27.73 | 25.06 | 27.52 | 28.78 |
| RDL   | 28.10 | 28.18 | 29.56 | 29.40 | 28.07 | 25.13 | 27.68 | 29.18 |

Output PSNR (in dB) comparison of KSVD and RDL algorithms on real images corrupted by additive Gaussian noise, for input PSNR of 20.16 dB. PSNR values are averaged over 5 noise realizations.

### Table IV

|       | Lenna | Barbara | House | Tower | Boat | Mandrill | Cameraman | Parrot |
|-------|-------|---------|-------|-------|------|----------|-----------|--------|
| \(K\)-SVD | 20.84 | 20.84 | 21.26 | 21.06 | 20.74 | 19.60 | 20.59 | 21.20 |
| RDL   | 24.19 | 25.30 | 25.89 | 25.81 | 24.14 | 22.40 | 22.52 | 24.36 |

Output PSNR (in dB) comparison of KSVD and IRLS algorithms on real images corrupted by Laplacian noise, for input PSNR of 14.18 dB. Output PSNR values are averaged over 5 independent noise realizations.

25 realizations) values of ADR and the distance \(\kappa\) from the ground-truth, corresponding to Laplacian noise contamination, for different values of input SNR are tabulated in Table I. Resulting values show clear improvement over the \(K\)-SVD, especially for low values of input SNR. Improvement in ADR, as we can observe from Table I, over the \(K\)-SVD algorithm is approximately 20%. Similar trend is observed for Gaussian noise corruption as well, and the corresponding values are given in Table II.

B. Image denoising

In image denoising experiments, patches of size \(4 \times 4\) are extracted from the noisy input image, with an overlap of one pixel in both horizontal and vertical directions. Subsequently, the noisy patches are vectorized, and stacked in columns of the data matrix \(Y\). We conduct two image denoising experiments: one with the Tower image corrupted by heavy-tailed Laplacian noise and the other one with Gaussian noise contaminated Barbara image, which contains some textures and patterns. Details of the algorithms are described below separately for Gaussian and non-Gaussian noise.

1) Gaussian noise: Since \(\ell_2\) metric is more suitable for suppressing additive Gaussian noise over smooth regions, we first classify different parts of the image into smooth and non-smooth regions. The Canny edge detection algorithm is used on a smoothed version of the raw input image, obtained using a simple wavelet thresholding algorithm. Subsequently, at each pixel location, we extract a patch of size \(4 \times 4\) and classify them in one of two categories, namely ‘smooth patch’ and ‘edge patch’. Finally we construct two training matrices, one for each kind. To denoise the smooth patches, we run the usual \(\ell_2\) based \(K\)-SVD algorithm, whereas on the edge patches, the RDL algorithm is deployed. The output image is obtained by computing the average of the resulting denoised patches with overlap. The outcomes of the experiment on the ‘Barbara’ image are shown in the second row of Figure 2. The corresponding PSNR values, averaged over 5 independent noise realizations, are indicated below the respective images. The average output PSNR values over 5 realizations for different images, corresponding to an input PSNR of 20.16 dB are given in Table I.
Fig. 2. Comparison of denoising performance of the RDL and the $K$-SVD algorithms. The first and second rows correspond to Laplacian and Gaussian noise, respectively. (a) and (e): ground truth clean image. (b) and (f): noisy input image. (c) and (g): denoised output obtained using the $K$-SVD algorithm. (d) and (h): denoised image obtained using the RDL algorithm.

dB are reported in Table III. We observe a consistent increase in the output PSNR value by approximately 0.30 dB in case of the RDL, over its $\ell_2$ minimization based counterpart.

2) Non-Gaussian noise: In the case where the input image is corrupted by non-Gaussian noise, it is not required to perform the patch classification step, since the $\ell_1$ metric is the optimum choice for the overall image. Rest of the procedure is identical to the Gaussian noise case. The RDL algorithm is used to train the dictionary on the corpus of noisy patches extracted from the image, and the resulting overlapping denoised patches are averaged to obtain the final denoised output image. Experimental results on the ‘Tower’ image are shown in the first row of Figure 2. The RDL algorithm results in an improvement of 2.30 dB in the PSNR value over the $K$-SVD algorithm. PSNR of the denoised output for different images corresponding to an input PSNR of 14.18 dB is given in Table IV. Similar to the case of Gaussian noise, we observe that the RDL algorithm results in an improvement of output PSNR over $K$-SVD, by a bigger margin of approximately 3 – 4 dB.

IV. CONCLUSIONS

We have developed a dictionary learning algorithm, referred to as the RDL, aimed at minimizing the $\ell_1$ error on the data term and applied the algorithm for image denoising. The motivation behind using the $\ell_1$ metric as a measure of data fidelity is to achieve robustness to noise and alleviate the problem of over-smoothing the regions of texture and edges in images. Experiments on synthesized dataset indicate the superiority of the proposed algorithm over its $\ell_2$ based counterpart, namely the $K$-SVD algorithm, in terms of ADR and the distance of the recovered dictionary to the ground-truth. Denoising experiments on real images show that the RDL algorithm results in output images with better PSNR values, compared with the $K$-SVD algorithm. Textures and edges in the images are also better preserved.

REFERENCES

[1] I. Daubechies, R. D. Vore, M. Fornasier, and C. S. Gunturk, “Iteratively re-weighted least squares minimization for sparse recovery,” Communications on Pure and Applied Maths., vol. LXIII, pp. 1–38, 2010.
[2] D. L. Donoho, “Compressed sensing,” IEEE Trans. Info. Theory, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[3] E. J. Candes, J. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” arXiv:math/0503066v2, Dec. 2005.
[4] E. J. Candes and T. Tao, “Near-Optimal signal recovery from random projections: Universal encoding strategies?,” IEEE Trans. Info. Theory, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.
[5] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” IEEE Trans. on Sig. Proc., vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[6] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” IEEE Trans. on Image Proc., vol. 15, no. 12, pp. 3736–3745, Dec. 2006.
[7] J. Yang, J. Wright, T. S. Huang, and Y. Ma, “Image super-resolution via sparse representation,” IEEE Trans. on Image Proc., vol. 19, no. 11, pp. 2861–2873, Nov. 2010.
[8] S. Sahoo and W. Lu, “Image Inpainting using Sparse Approximation with Adaptive Window Selection,” WISP, Oct. 2011.
[9] J. Mairal, M. Elad, and G. Sapiro, “Sparserepresentation for color image restoration,” IEEE Trans. on Image Proc., vol. 17, pp. 5369, Jan. 2008.
[10] S. P. Kasiviswanathan, H. Wang, A. Banerjee, and P. Melville, “Online $\ell_1$ dictionary learning with application to novel document detection,” Advances in Neural Information Processing Systems, vol. 25, pp. 2267–2275, 2012.