A Wide-Angle View at Iterated Shrinkage Algorithms *

M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky

The Computer Science Department
The Technion - Israel Institute of Technology, Haifa 32000, Israel.

ABSTRACT

Sparse and redundant representations – an emerging and powerful model for signals – suggests that a data source could be described as a linear combination of few atoms from a pre-specified and over-complete dictionary. This model has drawn a considerable attention in the past decade, due to its appealing theoretical foundations, and promising practical results it leads to. Many of the applications that use this model are formulated as a mixture of ℓ_2-ℓ_p ($p \leq 1$) optimization expressions. *Iterated Shrinkage* algorithms are a new family of highly effective numerical techniques for handling these optimization tasks, surpassing traditional optimization techniques. In this paper we aim to give a broad view of this group of methods, motivate their need, present their derivation, show their comparative performance, and most important of all, discuss their potential in various applications.

Keywords: Sparse, Redundant, Iterated Shrinkage, Dictionary, Optimization, ℓ_2-norm, ℓ_1-norm.

1. INTRODUCTION

Among the many ways to mathematically model signals, a recent and appealing approach employs sparse and redundant representations.[6] In this model, a signal $\mathbf{x} \in \mathbb{R}^n$ is assumed to be composed as a linear combination of few atoms from a pre-specified and redundant dictionary $\mathbf{D} \in \mathbb{R}^{n \times m}$, i.e.

$$\mathbf{x} = \mathbf{D}\alpha \ , \tag{1}$$

where $\alpha \in \mathbb{R}^m$ is expected to be sparse[†], $\|\alpha\|_0 \ll n$. Various encouraging theoretical results on the uniqueness of sparse enough solutions, and an equivalence between the sparsest solution and the shortest one in ℓ_1, suggest that finding the signal representation α is a computationally feasible task.[15–17,51,52]

On the practical front, a sequence of works employed successfully this model to different applications, such as denoising, inpainting, deblurring, image enhancement in general, compression, and more, all leading to state-of-the-art results.[2,23,24,27,28,40,41,48–50] Interestingly, at their core, many of the applications that use this model lead to an optimization task that mixes ℓ_2 and ℓ_p ($p \leq 1$) expressions in the form[‡]

$$f(\alpha; \mathbf{x}, \mathbf{D}) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p \ . \tag{2}$$

In this notation, f is a function of the vector α, parameterized by $\mathbf{x}$ and $\mathbf{D}$. In the various considered applications, we desire to minimize[§] $f(\alpha)$ with respect to α, assuming that $\mathbf{D}$ and $\mathbf{x}$ are given.

Until recently, these problems were traditionally treated using various classic iterative optimization algorithms ranging from steepest-descent and conjugate gradient to more involved interior-point algorithms.[6,32] However, all these general purpose methods are often found inefficient, requiring too many iterations and too many computations to reach their destination. This is especially the case for high-dimensional problems, as often encountered in image processing.

*This research was partly supported by the Israeli Science Foundation (ISF) grant No. 796/05 and by United States Israel Binational Science Foundation (BSF) grant No. 2004199.

†The ℓ_0-norm counts the number of non-zeros in the vector.

‡While we use here the ℓ_p-norm to measure sparsity, we can use many other similar forms of measures, $\rho(\alpha)$, just as described in.[25]

§We will allow the shortcut notation $f(\alpha)$ throughout this paper, where the context is obvious.

Another alternative solver, suitable for $p = 0$ is a greedy algorithm, such as the Orthogonal Matching Pursuit (OMP).[5,9,10,42] This method seeks a sparse minimizer for (2) one entry at a time. However, this method too is impractical for high dimensions, when the number of non-zeros is relatively high. Also, this algorithm often tends towards a wrong solution due to its greedy nature.

In recent years, a new and alternative family of numerical algorithms is gradually built, addressing the above optimization problems very effectively.[8,22,25,28–30,34,36,37] This family is the *Iterated Shrinkage* algorithms, that extend the classical Donoho-Johnston shrinkage method for signal denoising.[13,18,43,47] Roughly speaking, in these iterative methods each iteration comprises of a multiplication by $\mathbf{D}$ and its adjoint, along with a scalar shrinkage step on the obtained α. Despite their simple structure, these algorithms are shown to be very effective in minimizing $f(\alpha)$ in Equation (2). A thorough theoretical analysis proves the convergence of these techniques, guarantees that the solution is the global minimizer for convex f (for $p = 1$), and studies the rate of convergence these algorithms exhibit.[8,25,34]

There are various *Iterated Shrinkage* methods, with variations between them. These algorithms are shown to emerge from very different considerations, such as the Expectation-Maximization (EM) algorithm in statistical estimation theory,[28] proximal point and surrogate-functions,[8,29] the use of the fixed-point strategy,[1] employment of a parallel coordinate-descent algorithm,[22] variations on greedy methods,[20] and more. This exposes the wealth with which one could design a solver for the above-described optimization task. In this paper we aim to give a broad view of this group of methods, their derivation, their comparative performance, and most important of all, their potential in many applications.

In Section 2 we describe how various applications that rely on sparse and redundant representations lead to a problem structure similar to the one posed in Equation (2). This forms the motivation for developing effective solvers for this optimization task. Section 3 discusses classical means for solving Equation (2), and presents the special case where $\mathbf{D}$ is a unitary matrix, leading to a closed form shrinkage solution. This stands as a motivating fact in the quest for effective shrinkage algorithms that will handle the more general case. Section 4 describes several ways to develop *Iterated Shrinkage* algorithms, based on the methods described above. In Section 5 we present several simulation results that compare the above-discussed methods. We also provide results in deblurring of images in order to demonstrate these algorithms in action. Section 6 concludes this paper, with a discussion on open problems in this arena.

2. THE NEED: SPARSE & REDUNDANT REPRESENTATIONS

Suppose we accept the fundamental assumptions of the sparse and redundant representation model, implying that we expect the signal $\mathbf{x}$ to emerge from the multiplication $\mathbf{x} = \mathbf{D}\alpha$ with a very[¶] sparse α. We now show that this model assumption implies clear and constructive methods for various applications.

Denoising: Consider a noisy version of $\mathbf{x}$, obtained by the addition of a zero-mean unit variance white and i.i.d Gaussian vector $\mathbf{v} \in \mathbb{R}^n$. I.e., we measure $\mathbf{y} = \mathbf{x} + \mathbf{v}$ and we would like to recover $\mathbf{x}$ from it. Assuming a generalized Gaussian distribution of α, $P(\alpha) \sim \exp\{-\lambda\|\alpha\|_p^p\}$, the Maximum-A-posteriori-Probability (MAP) estimation[6,32] leads this denoising task to optimization problem

$$\hat{\alpha} = \arg\min_{\alpha} \ \frac{1}{2}\|\mathbf{y} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p \ , \tag{3}$$

and the result is obtained by $\hat{\mathbf{x}} = \mathbf{D}\hat{\alpha}$. For a sparse enough solution $\hat{\alpha}$, the result is robust with respect to the choice of p, and thus we prefer to use $p = 1$ rather than $p = 0$ to get convexity. As we can see, the objective function we face is $f(\alpha; \mathbf{y}, \mathbf{D})$ in terms of Equation (2).

General (Linear) Inverse Problems: Similar to the above, assume that the original signal $\mathbf{x}$ went through a linear degradation operation $\mathbf{H}$ and then additive noise as above. This means that we measure $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$, and aim to recover $\mathbf{x}$. MAP estimation leads to the expression

$$\hat{\alpha} = \arg\min_{\alpha} \ \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p \ , \tag{4}$$

[¶]How sparse is *very sparse*? There is no clear answer to that, but we shall assume hereafter that (i) α is the sparsest solution to this linear equation; and (ii) it is reconstructible via practical algorithms, such as Basis Pursuit.[15–17]

and the estimated result is $\hat{\mathbf{x}} = \mathbf{D}\hat{\alpha}$, as before. This time, we obtained the objective function $f(\alpha; \mathbf{y}, \mathbf{HD})$ in terms of Equation (2). This structure could serve various tasks, such as denoising (where $\mathbf{H} = \mathbf{I}$), deblurring (where $\mathbf{H}$ stands for a blur operator), inverse-Radon transform in tomography (where $\mathbf{H}$ stands for the set of parallel projections), signal interpolation (demosaicing, inpainting)[41] (where $\mathbf{H}$ represents the various sampling/blurring masks), and more.

Signal Separation: Assume that we are given a mixture signal $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{v}$, where $\mathbf{v}$ is an additive noise, $\mathbf{x}_1$ emerges from a model that uses the dictionary $\mathbf{D}_1$, and similarly $\mathbf{x}_2$ refers to a model that uses $\mathbf{D}_2$. We desire to separate the signal to its three ingredients. MAP is used again to obtain[26, 50]

$$\hat{\alpha}_1, \hat{\alpha}_2 = \arg\min_{\alpha_1,\alpha_2} \frac{1}{2}\|\mathbf{y} - \mathbf{D}_1\alpha_1 - \mathbf{D}_2\alpha_2\|_2^2 + \lambda\|\alpha_1\|_p^p + \lambda\|\alpha_2\|_p^p \ . \tag{5}$$

The estimated pieces are obtained by $\hat{\mathbf{x}}_1 = \mathbf{D}_1\hat{\alpha}_1$ and $\hat{\mathbf{x}}_2 = \mathbf{D}_2\hat{\alpha}_2$. The above problem, known as Morphological Component Analysis (MCA), fits again the general structure described in (2), using the function $f([\alpha_1^T, \alpha_2^T]^T; \mathbf{y}, [\mathbf{D}_1, \mathbf{D}_2])$.

Compression: Given a signal $\mathbf{x}$, we aim to compress it by allowing some pre-specified ℓ_2 error ϵ. This implies that we consider its representation $\hat{\alpha}$ that solves[12]

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_p^p \ \text{ subject to } \ \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 \le \epsilon^2 \ . \tag{6}$$

Instead of solving this problem, we could solve the problem described in Equation (2), while seeking λ that satisfies the above constraint with near-equality. The actual compression is done by quantizing the non-zeros in the vector $\hat{\alpha}$ and transmitting these values, along with their indices. Again we find ourselves handling the same format as in Equation (2).

Compressed-Sensing: Suppose that we aim to sample the signal $\mathbf{x}$ while compressing it. Compressed-sensing suggests that we sense the vector $\mathbf{y} = \mathbf{Q}\mathbf{x}$, where $\mathbf{Q} \in \mathbb{R}^{q\times n}$ contains a set of $q \ll n$ projection directions onto which the signal is projected.[4, 14, 53] Reconstruction of the signal from its samples is obtained by solving

$$\hat{\alpha} = \arg\min_{\alpha} \frac{1}{2}\|\mathbf{y} - \mathbf{Q}\mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p \ , \ \text{ and } \ \tilde{\mathbf{x}} = \mathbf{D}\hat{\alpha} \ , \tag{7}$$

and again we resort to the need to handle Equation (2) with $f(\alpha; \mathbf{y}, \mathbf{QD})$.

The bottom line to the above discussion is that the structure described in Equation (2) is fundamental to all these (and other) applications, and thus we are strongly encouraged to handle this optimization problem with efficient algorithms. In the next section we present the classical (and less effective) techniques that may seen as natural choices for use, and then turn in the section to follow to describe the better performing, *Iterated Shrinkage*, algorithms.

3. MOTIVATION AND INSPIRATION

We now turn to describe several classical optimization techniques that could be (and have been!) used for minimizing the function in Equation (2). Then we will turn to discuss an inspiring case of this function where $\mathbf{D}$ is a unitary matrix, showing that a closed form solution can be obtained. All this stands as motivating evidence for the development of the *Iterated Shrinkage* algorithms that comes in the next section.

3.1 Poorly Performing Classical Solvers

By now, we have built a clear need for the effective minimization of functions of the form

$$f(\alpha; \mathbf{x}, \mathbf{D}) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p \ ,$$

and the question we face is how to do this. The immediate and natural answer that seems appropriate is to suggest general purpose optimization software packages/algorithms that cover this function as a special case. As we argue in this paper, this would be a wrong step to take, as those general purpose tools are ineffective for the problem at hand, due to the sparsity of the desired solution; In fact, far more efficient algorithms could be designed for this specific structure, exploiting this additional knowledge.

One of the most simple algorithms for minimizing $f(\alpha;\mathbf{x},\mathbf{D})$ is the steepest descent, suggesting an update formula of the form:

$$\hat{\alpha}_{i+1} = \hat{\alpha}_i - \mu \cdot \nabla f(\alpha;\mathbf{x},\mathbf{D})|_{\hat{\alpha}_i} = \hat{\alpha}_i - \mu \left[\mathbf{D}^T(\mathbf{D}\hat{\alpha}_i - \mathbf{x}) + \lambda p \cdot \text{sign}\{\alpha\}|\hat{\alpha}_i|^{p-1}\right] \ , \tag{8}$$

where the step-size parameter μ could be found by a line search (1D numerical optimization). Gradient Descent may not work well for non-smooth functions, and smoothing is thus often practiced. This method is known to converge with an asymptotic rate proportional to $(\kappa-1)/(\kappa+1)$, where κ is the condition number of the Hessian of $f(\alpha)$.

An improved algorithm is the conjugate gradient method, which merges the current gradient and the previous update direction, using a pre-specified formula (based on the quadratic case). This new direction is then used as in Equation (8) to update the solution, with μ found by a line search. The asymptotic rate of convergence in this case is better, being $(\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$.

Turning from first order to second order techniques, a Newton iterative algorithm could be proposed, where the Hessian and the gradient of the function around the current solution form a linear set of equations to be solved. The dimension of this linear system, m, implies that for high dimensions a direct solution ($m \times m$ matrix inversion) is impossible, and thus a conjugate gradient iterative solver for this system is employed in each iteration. This is known as the truncated Newton algorithm, when we restrict the number of CG steps.

An interesting and presumably better line of algorithms for handling the minimization of $f(\alpha)$ are interior-point techniques. For the case $p = 1$, our minimization could be replaced by the following quadratic programming task[6] :

$$\arg\min_{\mathbf{z},\delta} \ \lambda \mathbf{1}^T\mathbf{z} + \frac{1}{2}\|\delta\|_2^2 \ \text{ subject to } \ [\mathbf{D},-\mathbf{D}]\mathbf{z} + \delta = \mathbf{x} \ \text{ and } \ \mathbf{z} \geq 0. \tag{9}$$

In this conversion of the problem format, the representation α is separated to its positive and negative parts, being the first and second pieces in the vector $\mathbf{z} \in \mathbb{R}^{2m}$. The vector δ stands for the deviation of $\mathbf{x}$ from $\mathbf{D}\alpha$.

There are various interior point algorithms that solve (9). Common to all is the concept of iterating from within the feasible solution set towards its boundaries, where the solution is known to reside. An effective algorithm could also exploit the ability to work on both the original problem in Equation (9) and its dual (i.e., primal-dual algorithms), thereby using the duality gap as an indication of convergence. The core ingredient in interior-point methods is a Newton iteration, as mentioned above, where the positivity constraint is converted to a barrier (e.g., log) function. This means that each iteration requires a solver for an $m \times m$ linear set of equations (at least), and this governs the complexity of such algorithms. For very large m, this implies that these techniques are no longer practical, unless they employ the truncated Newton method mentioned above, and this impairs their efficiency.

3.2 The Unitary Case - A Source of Inspiration

If $\mathbf{D}$ is unitary, the minimization of $f(\alpha)$ can be manipulated in a sequence of simple steps and turned into a set of m independent and identical one-dimensional optimization tasks, that are easily handled. Starting from Equation (2), and using the identity $\mathbf{D}\mathbf{D}^T = \mathbf{I}$, we get

$$f(\alpha;\mathbf{x},\mathbf{D}) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p = \frac{1}{2}\|\mathbf{D}(\mathbf{D}^T\mathbf{x} - \alpha)\|_2^2 + \lambda\|\alpha\|_p^p \ . \tag{10}$$

Exploiting the fact that ℓ_2-norm is unitary-invariant, we can remove the multiplication by $\mathbf{D}$ in the first term. Denoting $\alpha_0 = \mathbf{D}^T\mathbf{x}$, we obtain

$$f(\alpha;\alpha_0) = \frac{1}{2}\|\alpha_0 - \alpha\|_2^2 + \lambda\|\alpha\|_p^p = \sum_{k=1}^{m}\left[\frac{1}{2}(\alpha_0[k] - \alpha[k])^2 + \lambda|\alpha[k]|^p\right] \ . \tag{11}$$

The minimization of the scalar function $g(x;x_0) = 0.5(x_0 - x)^2 + \lambda|x|^p$ with respect to x requires that the sub-gradient of g contains the zero, $0 \in \partial g(x, x_0)$, implying that we solve the equation

$$0 \in x - x_0 + \lambda p \cdot \begin{cases} (x)^{p-1} & x > 0 \\ (-|x|^{p-1}, +|x|^{p-1}) & x = 0 \\ -(-x)^{p-1} & x < 0 \end{cases} , \tag{12}$$

and this can be solved (analytically in some cases, or empirically in others – this is done once and off-line), leading to the expression $\hat{x}_{opt} = \mathcal{S}_{p,\lambda}(x_0)$ that finds the global minimizer of the scalar objective function $g(x;x_0)$ described above. Note that for the case of $p < 1$, Equation (12) has several solutions, one of them being $x = 0$. Thus, all solutions must be found and compared by the value of the function. The obtained *Shrinkage* function is a curve that maps an input value x_0 to the desired output value $\hat{x}_{opt}$. This function maps values near the origin to zero ($\mathcal{S}_{p,\lambda}(x_0) = 0$ for $|x_0| \leq T$), and those outside this interval are "shrinked", thus the name of this operator. The threshold T and the shrinkage effects are both functions of the parameters p and λ.

Back to our original problem, we have found a closed-form solution to the minimizer of $f(\alpha)$, based on the following two steps: (i) Compute $\alpha_0 = \mathbf{D}^T\mathbf{x}$; and (ii) Apply the operator $\mathcal{S}_{p,\lambda}$ on the entries of α_0 and obtain the desired $\hat{\alpha}$. Note that even if the original function is non-convex, the found solution is globally optimal if $\mathcal{S}_{p,\lambda}$ was designed properly.

The natural question raised by the above discussion is: when turning from a unitary matrix $\mathbf{D}$ to a non-unitary (and perhaps non-square) one, must we loose all this simplicity? As we show in the next section, one could handle the general problem using similar shrinkage tools, but will be required to perform a sequence of them to obtain the desired result.

4. THE NEW MEANS: ITERATED SHRINKAGE ALGORITHMS

There are various ways to develop *Iterated Shrinkage* algorithms. To the best of our knowledge, the first to explicitly propose such an algorithm were Kingsbury and Reeves[37] in the context of using the complex wavelets. However, their construction is done without a clear connection to the objective function in (2). We choose to start our discussion with the proximal (or surrogate) functions that were used by Daubechies, Defrise and De-Mol in their construction. As we show, this is also the foundation for the EM and the Bound-Optimization algorithm, as proposed by Figueiredo and Nowak. Then we turn to describe two entirely different constructions: the first, by Adeyemi and Davies, is based on the fixed-point method in optimization, coupled with the IRLS algorithm.[32] The second method uses a parallel coordinate descent algorithm[22] . Finally, we will discuss the StOMP method by Donoho, Drori, Stark, and Tsaig, that develops an *Iterated Shrinkage* method based on the matching pursuit method.

4.1 Surrogate Functions and the Proximal-Point Method

The discussion below is based on the work of Daubechies, Defrise, and De-Mol.[8] Considering the original function in Equation (2),

$$f(\alpha;\mathbf{x},\mathbf{D}) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p ,$$

let us add to it the following term

$$\Psi(\alpha;\alpha_0,\mathbf{D}) = \frac{c}{2}\|\alpha - \alpha_0\|_2^2 - \frac{1}{2}\|\mathbf{D}\alpha - \mathbf{D}\alpha_0\|_2^2 .$$

The parameter c will be chosen such that the function Ψ is strictly convex, implying that we require its Hessian to be positive definite, $c\mathbf{I} - \mathbf{D}^T\mathbf{D} \succ 0$. This is satisfied by the choice $c > \|\mathbf{D}^T\mathbf{D}\|_2 = \rho(\mathbf{D}^T\mathbf{D})$. This new objective function

$$\tilde{f}(\alpha;\mathbf{x},\alpha_0,\mathbf{D}) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p + \frac{c}{2}\|\alpha - \alpha_0\|_2^2 - \frac{1}{2}\|\mathbf{D}\alpha - \mathbf{D}\alpha_0\|_2^2 , \tag{13}$$

is the *surrogate* function that will be used in the proposed algorithm. As we shall see next, the fact that the term $\|\mathbf{D}\alpha\|_2^2$ drops in the new function, turns the minimization task into a much simpler and effective one. Opening the various terms in Equation (13) and re-organizing it, we obtain a new expression of the form

$$\begin{aligned}\tilde{f}(\alpha;\mathbf{x},\alpha_0,\mathbf{D}) &= \frac{1}{2}\|\mathbf{x}\|_2^2+\frac{1}{2}\|\mathbf{D}\alpha_0\|_2^2+\frac{c}{2}\|\alpha_0\|_2^2-\mathbf{x}^T\mathbf{D}\alpha+\lambda\|\alpha\|_p^p+\frac{c}{2}\|\alpha\|_2^2-c\alpha^T\alpha_0+\alpha^T\mathbf{D}^T\mathbf{D}\alpha_0= \quad (14)\\ &= \text{Const}-\alpha^T\left[\mathbf{D}^T\left(\mathbf{x}-\mathbf{D}\alpha_0\right)+c\alpha_0\right]+\lambda\|\alpha\|_p^p+\frac{c}{2}\|\alpha\|_2^2\ .\end{aligned}$$

The constant in the above expression contains all the terms that are dependent on $\mathbf{x}$ and α_0 alone. Zeroing the derivative of this function with respect to α leads to the equation

$$\nabla\tilde{f}(\alpha;\mathbf{x},\alpha_0,\mathbf{D})=-\left[\mathbf{D}^T\left(\mathbf{x}-\mathbf{D}\alpha_0\right)+c\alpha_0\right]+\lambda p\cdot\text{sign}(\alpha)\cdot|\alpha|^{p-1}+c\alpha=0\ . \quad (15)$$

At this stage, one might be tempted to believe that since this equation to too complex, we better use the above gradient within an iterative scheme of some sort. However, it turns out that this equation is simple and leads to a closed-form solution. In order to show a resemblance between this equation and the analysis in Equation (12), we re-write it as

$$\alpha-\mathbf{v}_0+\frac{\lambda}{c}p\cdot\text{sign}(\alpha)\cdot|\alpha|^{p-1}=0\ , \quad (16)$$

where we use the term

$$\mathbf{v}_0=\frac{1}{c}\mathbf{D}^T\left(\mathbf{x}-\mathbf{D}\alpha_0\right)+\alpha_0\ . \quad (17)$$

This set of m equations is decoupled, enabling us to handle each separately, very similar to the way it was done in Equations (11) and (12), and leading to the same shrinkage operation as before. Thus, the derivative is nulled for the choice

$$\alpha_{opt}=\mathcal{S}_{p,\lambda/c}(\mathbf{v}_0)=\mathcal{S}_{p,\lambda/c}\left(\frac{1}{c}\mathbf{D}^T\left(\mathbf{x}-\mathbf{D}\alpha_0\right)+\alpha_0\right). \quad (18)$$

and this is a global minimizer of the function $\tilde{f}(\alpha;\mathbf{x},\alpha_0,\mathbf{D})$ in Equation (13).

So far we managed to convert the original function f to a new function $\tilde{f}$, for which we are able to get a closed form expression for its global minimizer. This change of the objective function depends on the choice of the vector α_0. The core idea of using the surrogate function is that we minimize the function f iteratively, producing the sequence of results $\{\alpha_i\}_i$, where at the $i+1$-th iteration we minimize $\tilde{f}$ with the assignment $\alpha_0=\alpha_i$. Perhaps the more surprising of all is the fact that the sequence of solutions $\{\alpha_i\}_i$ is proven to converge to the (local) minimizer of the original function f. Thus, the proposed algorithm is simply an iterated application of Equation (19) in the form

$$\alpha_{i+1}=\mathcal{S}_{p,\lambda/c}\left(\frac{1}{c}\mathbf{D}^T\left(\mathbf{x}-\mathbf{D}\alpha_i\right)+\alpha_i\right). \quad (19)$$

We shall refer hereafter to this algorithm as the Separable Surrogate Functionals (SSF) method.

The above approach can be interpreted as the proximal-point algorithm,[45] which is well known in optimization theory[25] . The function $\Psi(\alpha;\alpha_{i-1},\mathbf{D})$ is a distance measure to the previous solution. When added to the original function at the i-th iteration, it promotes proximity between subsequent estimates of the iterative process. Surprisingly, while this is expected to have an effect of slowing down the algorithm, in the case discussed here it is actually leading to a substantial speed-up.

4.2 Expectation-Maximization and Bound-Optimization Approaches

The discussion above could be replaced with a different set of considerations, and yet, one that leads to the very same algorithm. This new perspective is taken from the work of Figueiredo, Nowak, and Bioucas-Dias, that uses the Expectation-Maximization (EM) estimator, or better-yet, its alternative deterministic optimization foundation - the Bound-Optimization method[‖].[28–30]

Starting from the function $f(\alpha)$ in Equation (2), which is hard to minimize, the bound-optimization method suggests to use a related function $Q(\alpha, \alpha_0)$ that has the following properties:

1. **Equality at $\alpha = \alpha_0$:** $Q(\alpha_0, \alpha_0) = f(\alpha_0)$;
2. **Upper-bounding the original function:** $Q(\alpha, \alpha_0) \geq f(\alpha)$ for all α; and
3. **Tangent at α_0:** $\nabla Q(\alpha, \alpha_0)|_{\alpha=\alpha_0} = \nabla\, f(\alpha)|_{\alpha=\alpha_0}$.

Such a function necessarily exists if the Hessian of the original function is bounded, because then one can propose a bounding quadratic function, satisfying these conditions exactly. Using this function, the sequence of solutions generated by the recurrent formula

$$\alpha_{i+1} = \arg\min_{\alpha} Q(\alpha, \alpha_i) \tag{20}$$

is guaranteed to converge to a local minimum of the original function $f(\alpha)$.[11,25,35,38] As can be easily verified, the choice from the previous sub-section,

$$Q(\alpha, \alpha_0) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p + \frac{c}{2}\|\alpha - \alpha_0\|_2^2 - \frac{1}{2}\|\mathbf{D}\alpha - \mathbf{D}\alpha_0\|_2^2,$$

satisfies all these three conditions, and as such, leads to the desired optimization algorithm. Note that beyond the above three conditions, we need to make sure that minimization of Q is easier, as indeed happens, due to its separability. Also, the value $f(\alpha_i)$ is necessarily decreasing since

$$f(\alpha_{i+1}) = Q(\alpha_{i+1}, \alpha_i) = \min_{\alpha} Q(\alpha, \alpha_i) \leq Q(\alpha_i, \alpha_i) = f(\alpha_i). \tag{21}$$

In the above inequality, the relation $\min_\alpha Q(\alpha, \alpha_i) \leq Q(\alpha_i, \alpha_i)$ is guaranteed to give a strong decrease in the function's value due to the second property[**].

4.3 An IRLS-Based Algorithm

One of the more popular algorithms for minimization of $f(\alpha)$ in Equation (2) is the Iterative-Reweighed Least-Squares (IRLS) algorithm, that converts non-ℓ_2-norms to ℓ_2 ones by an introduction of weighting.[32] In this approach, an expression of the form $\|\alpha\|_p^p$ is replaced by $\alpha^T\mathbf{W}^{-1}(\alpha)\alpha$, where $\mathbf{W}(\alpha)$ is a diagonal matrix, holding the values $|\alpha[k]|^{2-p}$ in its main diagonal. While this change by itself is trivial, when plugged wisely into an iterative process, it leads to an appealing algorithm. Consider the function

$$f(\alpha; \mathbf{x}, \mathbf{D}) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\alpha^T\mathbf{W}^{-1}(\alpha)\alpha\ .$$

Assume that we hold a current solution α_0, and we aim to update it. This is done in two stages. The first updates α assuming a fixed $\mathbf{W}$. This requires a minimization of simple quadratic function, leading to the linear system

$$\nabla f(\alpha; \mathbf{x}, \mathbf{D}) = -\mathbf{D}^T(\mathbf{x} - \mathbf{D}\alpha) + 2\lambda\mathbf{W}^{-1}(\alpha)\alpha = 0\ . \tag{22}$$

[‖] This technique is also known as the Majorization-Maximization method.

[**] By the Tailor expansion, it is easily verified that being tangent to the original function f at α_i implies a descent of at least $\nabla f(\alpha_i)^T\mathbf{H}^{-1}\nabla f(\alpha_i)$, where $\mathbf{H}$ is an upper bound on the original function's Hessian.

Thus, inversion of the matrix $\mathbf{D}^T\mathbf{D} + 2\lambda\mathbf{W}^{-1}$, either directly (for low-dimensional problems) or iteratively (several CG steps) leads to the desired update. Once found, we update $\mathbf{W}$ based on the new values of the found solution. This is the IRLS core as proposed in[32] , and its performance parallels those of the techniques mentioned in Section 2 (specifically, the truncated Newton method).

An interesting twist over the above algorithm is proposed in[1] , leading to yet another *Iterated Shrinkage* algorithm. Consider Equation (22), and write it slightly different, by adding and subtracting $c\alpha$. The constant $c \geq 1$ is a relaxation constant, whose role is to be discovered soon. Thus,

$$-\mathbf{D}^T\mathbf{x} + (\mathbf{D}^T\mathbf{D} - c\mathbf{I})\alpha + (2\lambda\mathbf{W}^{-1}(\alpha) + c\mathbf{I})\alpha = 0 \ . \tag{23}$$

A well-known technique in optimization, known as the fixed-point iteration method, suggests a way to construct iterative algorithms by assigning (wisely!) iteration counter to the appearance of α in the equation. In our case, the proposed assignment is

$$\mathbf{D}^T\mathbf{x} - (\mathbf{D}^T\mathbf{D} - c\mathbf{I})\alpha_i = (2\lambda\mathbf{W}^{-1}(\alpha_i) + c\mathbf{I})\alpha_{i+1} \ , \tag{24}$$

leading to the iterative equation

$$\begin{aligned} \alpha_{i+1} &= \left(\frac{2\lambda}{c}\mathbf{W}^{-1}(\alpha_i) + \mathbf{I}\right)^{-1}\left(\frac{1}{c}\mathbf{D}^T\mathbf{x} - \frac{1}{c}(\mathbf{D}^T\mathbf{D} - c\mathbf{I})\alpha_i\right) \\ &= \mathbf{S}\cdot\left(\frac{1}{c}\mathbf{D}^T(\mathbf{x} - \mathbf{D}\alpha_i) + \alpha_i\right) \ . \end{aligned} \tag{25}$$

where we have defined the diagonal matrix

$$\mathbf{S} = \left(\frac{2\lambda}{c}\mathbf{W}^{-1}(\alpha_i) + \mathbf{I}\right)^{-1} = \left(\frac{2\lambda}{c}\mathbf{I} + \mathbf{W}(\alpha_i)\right)^{-1}\mathbf{W}(\alpha_i) \ . \tag{26}$$

This matrix plays the role of shrinkage of the entries in the vector $\frac{1}{c}\mathbf{D}^T(\mathbf{x} - \mathbf{D}\alpha_i) + \alpha_i$, just as it is done in Equation (19). Indeed, every entry is multiplied by the scalar value

$$\frac{|\alpha_i[k]|^{2-p}}{\frac{2\lambda}{c} + |\alpha_i[k]|^{2-p}}, \tag{27}$$

such that for large values of $|\alpha_i[k]|$ this factor is nearly 1, whereas for small values of $|\alpha_i[k]|$ this value tends to zero, just as shrinkage does. Note that this algorithm cannot be initialized with zeros, as in previous methods; in our tests we use a vector with a constant (0.001) in all its entries. Interestingly, once an entry in the solution becomes zero, it can never be "revived", implying that this algorithm may get stuck, avoiding the near-by local minimum.

The constant c is crucial for the convergence of this algorithm, and its choice is similar to the one in the SSF algorithm. The iterative algorithm in Equation (25) applies the matrix $\mathbf{S}\cdot(\frac{1}{c}\mathbf{D}^T\mathbf{D} - \mathbf{I})$ sequentially. For convergence, we must require that the spectral radius of this matrix is below 1. Since $\|\mathbf{S}\|_2 \leq 1$, this implies that c should be chosen to be larger than $\rho(\mathbf{D}^T\mathbf{D})/2$, which is twice more tolerant, compared to the the choice made in the SSF algorithm.

4.4 The Parallel Coordinate Descent (PCD) Algorithm

We now turn to describe the PCD algorithm[22,25] . The construction below starts from a simple coordinate descent algorithm, but then merges a set of such descent steps into one easier joint step, leading to the Parallel Coordinate Descent (PCD) *Iterated Shrinkage* method. The rationale practiced here should remind the reader of the BCR algorithm by Sardy, Bruce, and Tseng,[46] that handles a special case of $\mathbf{D}$ being a union of unitary matrices.

Returning to $f(\alpha)$ in Equation (2), we start by proposing a Coordinate Descent (CD) algorithm that updates one entry at a time in α while keeping the rest untouched. A sequence of such rounds of m steps (addressing

each of the entries in $\alpha \in \mathbb{R}^m$) is necessarily converging. Interestingly, as we are about to show, each of these steps is obtained via shrinkage, similar to the process described in Equation (12).

Assuming that the current solution is α_0, we desire to update the k-th entry around its current value $\alpha_0[k]$. This leads to a 1D function of the form

$$g(z) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha_0 - \mathbf{d}_k(z - \alpha_0[k])\|_2^2 + \lambda|z|^p \ . \tag{28}$$

The vector $\mathbf{d}_k$ is the k-th column in $\mathbf{D}$. The term $\mathbf{d}_k(z - \alpha_0[k])$ removes the effect of the old value and adds the new one. Zeroing the derivative of this function w.r.t. z leads to

$$\begin{aligned} \frac{dg(z)}{dz} &= -\mathbf{d}_k^T(\mathbf{x} - \mathbf{D}\alpha_0 - \mathbf{d}_k(z - \alpha_0[k])) + p\lambda \text{sign}(z)|z|^{p-1} \\ &= (z - \alpha_0[k])\|\mathbf{d}_k\|_2^2 - \mathbf{d}_k^T(\mathbf{x} - \mathbf{D}\alpha_0) + p\lambda \text{sign}(z)|z|^{p-1} = 0 \ . \end{aligned} \tag{29}$$

A re-organization of the last expression gives

$$z - \alpha_0[k] - \frac{1}{\|\mathbf{d}_k\|_2^2}\mathbf{d}_k^T(\mathbf{x} - \mathbf{D}\alpha_0) + p\frac{\lambda}{\|\mathbf{d}_k\|_2^2}\text{sign}(z)|z|^{p-1} = 0 \ ,$$

implying a shrinkage operation, just as in Equation (12), of the form

$$\alpha_k^{opt} = \mathcal{S}_{p,\lambda/\|\mathbf{d}_k\|_2^2}\left(\frac{1}{\|\mathbf{d}_k\|_2^2}\mathbf{d}_k^T(\mathbf{x} - \mathbf{D}\alpha_0) + \alpha_0[k]\right) \ . \tag{30}$$

While this algorithm may work well in low-dimensional cases, it is impractical in the general high-dimension cases, where the matrix $\mathbf{D}$ is not held explicitly, and only multiplications by it and its adjoint are possible. This is because its use requires the extraction of columns from $\mathbf{D}$, one at a time.

Thus, we consider a modification of the above method. We rely on the following property: When minimizing a function, if there are several descent directions, then any non-negative combination of them is also a descent direction. Thus, we propose a simple linear combination of the set of all the above m steps in Equation (30). Since each of these steps handles one entry in the destination vector, we can write this sum as

$$\mathbf{v}_0 = \sum_{k=1}^{m} \mathbf{e}_k \cdot \mathcal{S}_{p,\lambda/\|\mathbf{d}_k\|_2^2}\left(\frac{1}{\|\mathbf{d}_k\|_2^2}\mathbf{d}_k^T(\mathbf{x} - \mathbf{D}\alpha_0) + \alpha_0[k]\right) = \begin{bmatrix} \mathcal{S}_{p,\lambda/\|\mathbf{d}_1\|_2^2}\left(\frac{1}{\|\mathbf{d}_1\|_2^2}\mathbf{d}_1^T(\mathbf{x} - \mathbf{D}\alpha_0) + \alpha_0[1]\right) \\ \vdots \\ \mathcal{S}_{p,\lambda/\|\mathbf{d}_k\|_2^2}\left(\frac{1}{\|\mathbf{d}_k\|_2^2}\mathbf{d}_k^T(\mathbf{x} - \mathbf{D}\alpha_0) + \alpha_0[k]\right) \\ \vdots \\ \mathcal{S}_{p,\lambda/\|\mathbf{d}_m\|_2^2}\left(\frac{1}{\|\mathbf{d}_m\|_2^2}\mathbf{d}_m^T(\mathbf{x} - \mathbf{D}\alpha_0) + \alpha_0[m]\right) \end{bmatrix} . \tag{31}$$

In the above expression, $\mathbf{e}_k$ is the trivial m-dimensional vector with zeros in all entries and 1 in the k-th one. Rewriting of this expression leads to the following simpler formula

$$\mathbf{v}_0 = \mathcal{S}_{p,\text{diag}(\mathbf{D}^T\mathbf{D})^{-1}\lambda}\left(\text{diag}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T(\mathbf{x} - \mathbf{D}\alpha_0) + \alpha_0\right) \ . \tag{32}$$

The term $\text{diag}(\mathbf{D}^T\mathbf{D})$ consists of the norms of the columns of the dictionary $\mathbf{D}$. These are used both for the weighting of the back-projected error $\mathbf{D}^T(\mathbf{x} - \mathbf{D}\alpha_0)$, and for the shrinkage operator. This set of weights can be computed off-line, before the algorithm starts, and there are fast ways to approximate these weights[††]. Notice that the obtained formula does not call for an extraction of columns from $\mathbf{D}$ as in (30), and the operations required are a direct multiplication of $\mathbf{D}$ and its adjoint by vectors, and in Equation (19).

[††]A stochastic algorithm proposed in[25] works as follows: for a white Gaussian vector $\mathbf{u} \sim \mathcal{N}(0, \mathbf{I}) \in \mathbb{R}^n$, the multiplication $\mathbf{D}^T\mathbf{u}$ has a covariance matrix $\mathbf{D}^T\mathbf{D}$. Thus, by choosing a set of such random vectors and computing their multiplication by $\mathbf{D}^T$, the variance of each entry is an approximation of the required norms.

While each of the CD directions is guaranteed to descend, their linear combination is not necessarily descending without a proper scaling. Thus, as proposed in,[22] we consider this direction and perform a line search along it. This means that the actual iterative algorithm is of the form

$$\begin{aligned}\alpha_{i+1} &= \alpha_i + \mu\left(\mathbf{v}_i - \alpha_i\right) = \\ &= \alpha_i + \mu\left(\mathcal{S}_{p,\text{diag}(\mathbf{D}^T\mathbf{D})^{-1}\lambda}\left(\text{diag}(\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T(\mathbf{x}-\mathbf{D}\alpha_i)+\alpha_i\right)-\alpha_i\right),\end{aligned} \tag{33}$$

with μ chosen by a line-search algorithm of some sort. This means that we are required to perform 1D optimization of the function

$$h(\mu) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\left(\alpha_i + \mu(\mathbf{v}_i - \alpha_i)\right)\|_2^2 + \lambda\|\alpha_i + \mu(\mathbf{v}_i - \alpha_i)\|_p^p \ ,$$

which requires only two more multiplications of $\mathbf{D}$ by $\mathbf{v}_i$ and α_i. This overall process will be referred to hereafter as the PCD algorithm.

As we can see, compared to the SSF algorithm in Equation (19), the one derived here is different in two ways: (i) the norms of the atoms in $\mathbf{D}$ play an important role in weighting the back-projected error, whereas the previous algorithm uses a constant; and (ii) the new algorithm requires a line-search to obtain a descent. We will return to discuss these differences towards the end of this section, in an attempt to bridge between the two methods.

4.5 StOMP: A Variation on Greedy Methods

The last algorithm we describe within this family of *Iterated Shrinkage* methods is the Stage-wise Orthogonal Matching Pursuit (StOMP).[20,27] As opposed to the previous methods, its construction does not emerge from an attempt to minimize $f(\alpha)$ in Equation (2), although it could be attributed to the minimization of this function with $p = 0$. Instead, the creators of StOMP embark from the Orthogonal Matching Pursuit (OMP) and the Least-Angle Regression (LARS) algorithms.[5,9,10,21,42] We should also mention that the StOMP algorithm was especially tailored for cases where the matrix $\mathbf{D}$ is random, such as in compressed-sensing, although it can be used in other cases quite successfully.[27] We now turn to describe both OMP and its variant – the StOMP.

We initialize the process by fixing the solution to be zero, $\alpha_0 = 0$, and thus the residual vector is $\mathbf{r}_0 = \mathbf{x} - \mathbf{D}\alpha_0 = \mathbf{x}$. We also define the support of the found solution and set it to be empty, $I_0 = \phi$. Turning to the first iteration and beyond ($i \geq 1$), the following steps are performed:

Back-Project the Residual: Compute $\mathbf{b}_i = \mathbf{D}^T(\mathbf{x} - \mathbf{D}\alpha_{i-1}) = \mathbf{D}^T\mathbf{r}_{i-1}$. Most of the obtained values are assumed to be zero-mean i.i.d. Gaussian due to the assumed randomness of the dictionary. Few outliers in the above vector are those referring to possible atoms in the construction of $\mathbf{x}$.

Thresholding: We define the set of dominant entries in $\mathbf{b}_i$ as $J_i = \{k|\ 1 \leq k \leq m,\ |b_i[k]| > T\}$, with a threshold chosen based on the permitted error $\|\mathbf{x} - \mathbf{D}\alpha\|_2^2$. If the threshold is chosen such that $|J_i| = 1$, we get the OMP, as only one entry is found and updated.

Union of Supports: We update the support of the desired solution to be $I_i = I_{i-1} \cup J_i$. Thus, this support is incrementally growing.

Restricted Least-Squares: Given the support and assuming that $|I_i| < n$, we solve the least-squares problem

$$\arg\min_{\alpha} \|\mathbf{x} - \mathbf{D}\mathbf{P}\alpha\|_2^2 \ , \tag{34}$$

where $\mathbf{P} \in \mathbb{R}^{m \times |I_i|}$ is an operator that chooses the $|I_i|$ entries from the vector α that are permitted to be non-zeros. This minimization is done using a sequence of K_0 Conjugate-Gradient (CG) iterations, where each iteration multiplies by $\mathbf{DP}$ and its adjoint.

Update of the Residual: Having found the updated vector α_i, we compute the residual to be $\mathbf{r}_i = \mathbf{x} - \mathbf{D}\alpha_i$.

As said above, when the thresholding step chooses one entry at a time, this is exactly the OMP method. The above process should be iterated several times (the authors recommend 10 iterations, as each iteration can potentially add many elements to the support), leading to the desired sparse solution of the approximated linear system $\mathbf{x} \approx \mathbf{D}\alpha$.

StOMP comes with a thorough theoretical analysis that considers the case of random dictionaries.[20,27] In such a case, the values in $\mathbf{b}_i$ could be interpreted as a sparse noisy vector, where the noise can be reliably assumed to be white zero mean Gaussian. Thus, a thresholding step that follows is nothing but a denoiser algorithm. In such a scenario, the creators of StOMP show an interesting phase transition from success to failure. However, despite this thorough analysis, when considering the original objective $f(\alpha)$ in Equation (2) with a general value of p, it is unclear whether StOMP is a still a minimizer of this function and whether convergence to a local minima point is guaranteed. Note that a simple change of the hard-thresholding by a soft one is meaningless, because the thresholding step in StOMP is used only for detecting additional atoms to be added to the support, and thus the magnitudes in $\mathbf{b}_i$ are never used.

We add the following observation that will be important in comparing the various algorithms. Every iteration of StOMP parallels in complexity to K_0 iterations of the SSF, IRLS-based, and the PCD algorithms. In that respect, this structure should remind the reader with the interior-point and the truncated Newton algorithms' inner solver, as described in Section 2.

4.6 Acceleration using Sequential Subspace Optimization (SESOP)

All the above algorithms apart from StOMP can be further accelerated in several ways. First, the idea of performing a line search, as described in the PCD is relevant to the SSF and the IRLS-based algorithms. All that is required is to use Equations (19) or (25) to compute a temporary result α_{temp}, and then define the solution as $\alpha_{i+1} = \alpha_i + \mu(\alpha_{temp} - \alpha_i)$, optimizing $f(\alpha_{i+1})$ with respect to the scalar μ. Unfortunately, this cannot be implemented with StOMP, because it is not clear which objective function this algorithm aims to minimize.

A second, and much more effective speed-up option for the SSF, IRLS-based, and PCD algorithms is the deployment of the Sequential Subspace Optimization (SESOP) method[25,44] . We describe it briefly here, referring the interested reader to the above references for more details. The original SESOP algorithm proposes to obtain the next iterate α_{i+1} via optimization of function f over an affine subspace spanned by the set of q recent steps $\{\alpha_{i-k} - \alpha_{i-k-1}\}_{k=0}^{q-1}$, and the current gradient. This $q+1$-dimensional optimization task can be addressed using a Newton algorithm. The main computational burden in this process is the need to multiply these directions by $\mathbf{D}$, but these q multiplications can be stored in previous iterations, thus enabling the SESOP speed-up algorithm with hardly any additional cost.

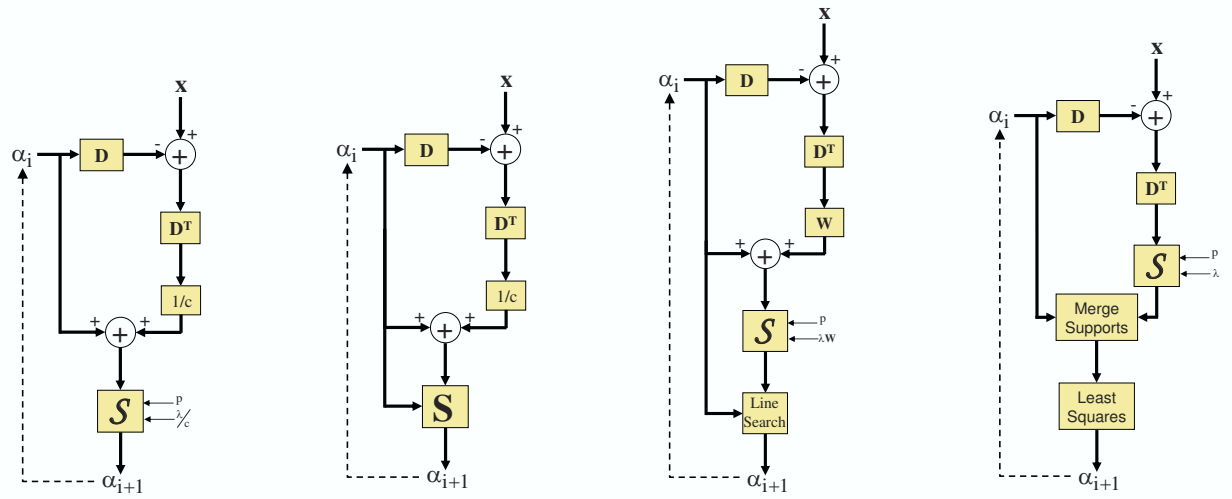
In the simple quadratic optimization case, SESOP algorithm with $q \geq 1$ amounts to regular CG. The $q = 0$ case is equivalent to a line-search. Preconditioning, like multiplication of the current gradient by inverse of the Hessian diagonal, may significantly accelerate the method.

In the context of the current paper, one can use the direction $(\alpha_{temp} - \alpha_i)$, provided by SSF, IRLS-based or PCD algorithm instead of the gradient used in the standard SESOP.[25] We shall demonstrate this approach in the next section and show how effective it is.

4.7 Bottom Line – *Iterated Shrinkage* Algorithms

Figure 1 presents the four *Iterated Shrinkage* algorithms described above as block diagrams. As can be seen, although different from each other in various ways, all rely on the same main computational stages, such as the term $\mathbf{x} - \mathbf{D}\alpha$, its back projection by multiplication by $\mathbf{D}^T$, and a threshodling/shrinkage step.

In this paper we deliberately avoided discussing in depth the theoretical study of the algorithms presented here. Nevertheless, we should state the following few facts: the SSF and the PCD algorithms are guaranteed to converge to a local minimizer of the function in Equation (2).[3,8,25] For $p = 1$, where this objective function becomes convex, a convergence to the global minimizer is guaranteed, making these algorithms exact solvers of the basis pursuit denoising problem. The work in[34] considers the $p = 0$ and shows that the sparsest solution is recoverable if it is sparse enough, depending on properties of $\mathbf{D}$, implying that in such a case we get the global minimizer as well. A similar claim appears also for StOMP[20] .



(a): SSF (b): IRLS-Based (c): PCD (d): StOMP

Figure 1. Block diagrams of the four obtained *Iterated Shrinkage* algorithms: (a) The Separable Surrogate Functional (SSF) method;[8,28–30] (b) The IRLS-based algorithm;[1] (c) The Parallel Coordinate Descent (PCD) method;[22,25] and (d) The Stage-wise Orthogonal Matching Pursuit (StOMP).[20,27] Note that the 4 block-diagrams seems to be very similar, but there are delicate differences between them in various locations.

Among the four core algorithms described, the SSF, the IRLS-based, and the PCD are closer to each other. Which of these is faster? Recall that the constant c in SSF must satisfy $c > \rho(\mathbf{D}^T\mathbf{D})$, and in the IRLS method half this value. On the other hand, the PCD uses a scaling of the same term using the weight matrix $\text{diag}(\mathbf{D}^T\mathbf{D})^{-1}$. As an example, for a dictionary built as a union of N unitary matrices, we require $c > N$, implying a weight of $1/N$ in Equation (19). In the PCD algorithm, the weight matrix is simply identity due to the normalized columns in $\mathbf{D}$. Thus, PCD gives $O(N)$ times stronger effect to the term $\mathbf{D}^T(\mathbf{x} - \mathbf{D}\alpha_i)$, and because of that we expect it to perform better. We shall verify this assumption in the next section, where we conduct a series of comparative experiments. We should remind the reader that the convergence speed is also expected to improve dramatically due to the SESOP and line search accelerations.

With all the above methods, their theoretical analysis, and the speed-up techniques that are added on top, one might get the impression that this field has covered all bases, with no open problems for future work. This, in fact, would be a very wrong conclusion. While we do know a lot more today on *Iterated Shrinkage* algorithms, there are various important questions, yet to be explored. We mention here few of these open questions:

Constrained Problem: In many cases, one wants to solve a variant of the above minimization problem, where the penalty $\|\mathbf{D}\alpha - \mathbf{x}\|_2^2$ is replaced by a constraint, demanding that this error goes below a threshold ϵ^2. While it is true that for a specific value of λ the two problems are equivalent, this value of λ is dependent on $\mathbf{x}$, implying that we need to solve a sequence of the above optimization procedures, and sweep through values of λ to satisfy the constraint. Could we propose a direct method that avoids this complication? It appears that the answer is positive – see [3] for a variant of the SSF algorithm that does the opposite (fixing the sparsity as constraint and minimizing the representation error).

Global Minimization: We mentioned several times throughout this paper that for $0 \leq p < 1$ a convergence to a local minimum solution is guaranteed. On the other hand, we are aware of results that guarantee global minimum solution for the case $p = 0$, when the solution is sparse enough. Thus, we believe that a similar global convergence could be guaranteed for the case $0 \leq p < 1$ in some cases.

A Closer Look at StOMP: What can be claimed about StOMP for the $p > 0$ case? better yet, how should we modify StOMP to serve these cases better? How can StOMP be speeded-up using a SESOP-like method? All these topics are yet to be studied.

Truncated Iteration Effect: Since the proposed algorithms are obviously implemented with finite and very small number of iterations, it is worth studying the theoretical and empirical behavior of such truncated iterative processes.

Choice of Sparsity Measure: Which p to use, or more generally, what measure $\rho(\alpha)$ we should choose, to lead to better results in the applications we serve? The answer to this question could be a learning procedure of the form described in [33] , where examples are used to learn the optimal measure.

New Iterated Shrinkage Methods: Can we develop new methods in this family of algorithms, or perhaps create variations on the above algorithms? Due to the diverse possibilities we see in the existing techniques, this seems to be a promising direction for further research. Two such very recent contributions that are not yet published are found in [7,31] .

5. SOME RESULTS: AN EXPERIMENTAL STUDY

In this section we present two sets of experiments; The first are synthetic ones aiming to study the convergence behavior of the various algorithms; The second target a specific application in image processing – image deblurring – using *Iterated Shrinkage* algorithms.

5.1 Part 1: A Synthetic Comparative Study

In order to compare the various algorithms's rate of convergence, we define one problem on which all are tested. We target the minimization of the function

$$f(\alpha; \mathbf{x}, \mathbf{D}) = \frac{1}{2}\|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda\rho(\alpha) \ .$$

where $\mathbf{D}$ is a 2D un-decimated separable wavelet transform of size $128^2 \times 128^2 \cdot 4$ (i.e., with $4:1$ redundancy), that operates on an image of size 128×128, with 2 levels and using the Daubechies 4-taps mother wavelet, as used in[30‡‡]. The function ρ was chosen to be a smoothed version of ℓ_1, as defined and used in[25] . The image $\mathbf{x}$ was created synthetically, by drawing a random sparse vector of size 128^2 with 66 non-zeros, α^*, multiplying it by $\mathbf{D}$, and adding to this a white Gaussian noise. The noise power was set such that the Signal to Noise Ratio $\|\mathbf{D}\alpha^*\|_2/\|\mathbf{x} - \mathbf{D}\alpha^*\|_2$ is 10. The parameter λ was set to $\lambda = 0.008$, so as to give good denoising performance, when seeking the minimizer of the above function.

The algorithms we compared herein are the following:

1. Gradient-based minimization algorithms with varying SESOP dimensions. SESOP-0 refers to the Normalized Steepest Descent (NSD), and SESOP-1 is similar (but not equivalent) to the Conjugate Gradient (CG). We also test the Conjugate Gradient algorithm, as described in Section 3.
2. Pre-conditioned gradient-based optimization with varying SESOP dimension. Pre-conditioning is obtained by multiplying the gradient of the function by the inverse of the function's Hessian main diagonal. Again, we test these PRE-SESOP-K with $K = 0, 1, 5, 8$.
3. Truncated Newton algorithm, using the Hessian and the gradient of the function to approximate a solution of a linear system via CG iterations.
4. SSF with and without SESOP. In that respect, SSF-SESOP-0 is the SSF algorithm with a line search. We test SSF-SESOP-K for $K = 0, 1, 5, 8$.
5. IRLS-based algorithm with and without SESOP, as in the SSF case.
6. PCD algorithm with and without SESOP, as in the SSF case.
7. From the *Sparselab* software package [19] we test StOMP and PDCO – an interior point primal-dual algorithm.

‡‡We obtained the code for this transform from Mario Figueiredo, who adapted it from the Rice-Wavelab package.

Figure 2 presents graphs showing the value of the objective as a function of the iteration for the above algorithms. In all these graphs, the value is shown after a substraction of the minimal value (found empirically, after sufficient number of iterations). The graphs start by presenting the classical techniques, choosing the best of them (PRE-SESOP-1), and comparing against a new group with the SSF algorithms. These comparisons proceed with the PCD and finally with the IRLS, using the same technique of comparing against the best so far.

Similarly, since the minimization of $f(\alpha; \mathbf{x}, \mathbf{D})$ can be interpreted as a denoising method with $\mathbf{D}\alpha_i$ as the denoised estimate, Figure 3 presents the improvement in SNR obtained as a function of the iteration, for these algorithms. ISNR is defined as

$$ISNR = 10 \cdot \log \left\{ \frac{\|\mathbf{x} - \mathbf{D}\alpha^*\|_2^2}{\|\mathbf{D}\alpha_i - \mathbf{D}\alpha^*\|_2^2} \right\} \text{ [dB]} , \tag{35}$$

where $\mathbf{D}\alpha^*$ is the original image we aim to estimate.

In these graphs we do not present all methods, and concentrate on the main differences. We found that CG and SESOP-1 perform very closely and thus SESOP-1 was omitted. Similarly, the truncated Newton method was found to perform very poorly (in fact, it is the worst among all methods tested, and this sheds some light on hih complexity of the PDCO interior-point algorithm, as will be shown hereafter), due to the complexity of each iteration, and thus it was omitted too. We do not present the SESOP-8 versions of the *Iterated-Shrinkage* algorithms here as all of them were found to perform comparable to SESOP-5.

There are many conclusions one can draw from these figures; we list here the main of them:

- Generally speaking, *Iterated Shrinkage* algorithms perform much better than the general purpose optimization techniques. This is true both for the minimization of the function and for denoising purposes;
- Among the various *Iterated Shrinkage* methods, the one that seems to converge the fastest (without speedup) in the first few iterations is the PCD. When accelerated with SESOP, the fastest method is the IRLS, but it get stuck, as predicted, missing the global minimum of the function.
- SESOP acceleration improves all the methods it is coupled with. The best performing algorithm is the PCD-SESOP combination.
- In terms of ISNR, the best techniques are the PCD-SESOP and SSF-SESOP (1-5), surpassing all others.

Before ending this part of the experiments, we report of yet another test, this time running two algorithms implemented in the *Sparselab* software package [19] – StOMP and PDCO. Since these are external software functions beyond our control, we concentrate on the number of activations of the transform (or its adjoint), and the ISNR reached for this computational effort.

While we cannot consider StOMP as a minimizer of the function in equation (2), it is still expected to perform well as a denoising algorithm. Thus, we use the same settings, and apply StOMP for denoising of $\mathbf{x}$. The stopping rule for this algorithm is the noise power posed as a threshold to the residual error. The rest of the parameters governing the run of StOMP are taken from the *Sparselab* package[19] , and manually optimized from there in order to get better ISNR.

In this test StOMP leads to an ISNR of 9.87dB, obtained after applying 202 activations of the transform or its adjoint. PCD-SESOP uses less operations ($\approx$ 160 transforms), getting an ISNR of 12.1dB. The reason for the inferior behavior of StOMP might be its tendency to provide a support for the solution that is much denser than required. In this test we get 658 non-zeros, implying that least-squares fits the noise to these wide set of atoms. As a side note we mention that if the support (66 non-zeros) would have been perfectly found by the algorithm, a least-squares estimator over these items would have led to ISNR=25.1dB. This means that while PCD does perform better, both algorithms have a wide range for further improvement. Note that if we re-run PCD with $p < 1$ and initialize with its previous ($p = 1$) solution, we may get a much improved denoising effect. The test with PDCO leads to 930 required transforms, that are part of 18 overall iterations. These lead to an ISNR of 10.6dB, implying that this algorithm is much slower than all of the tested methods described above.

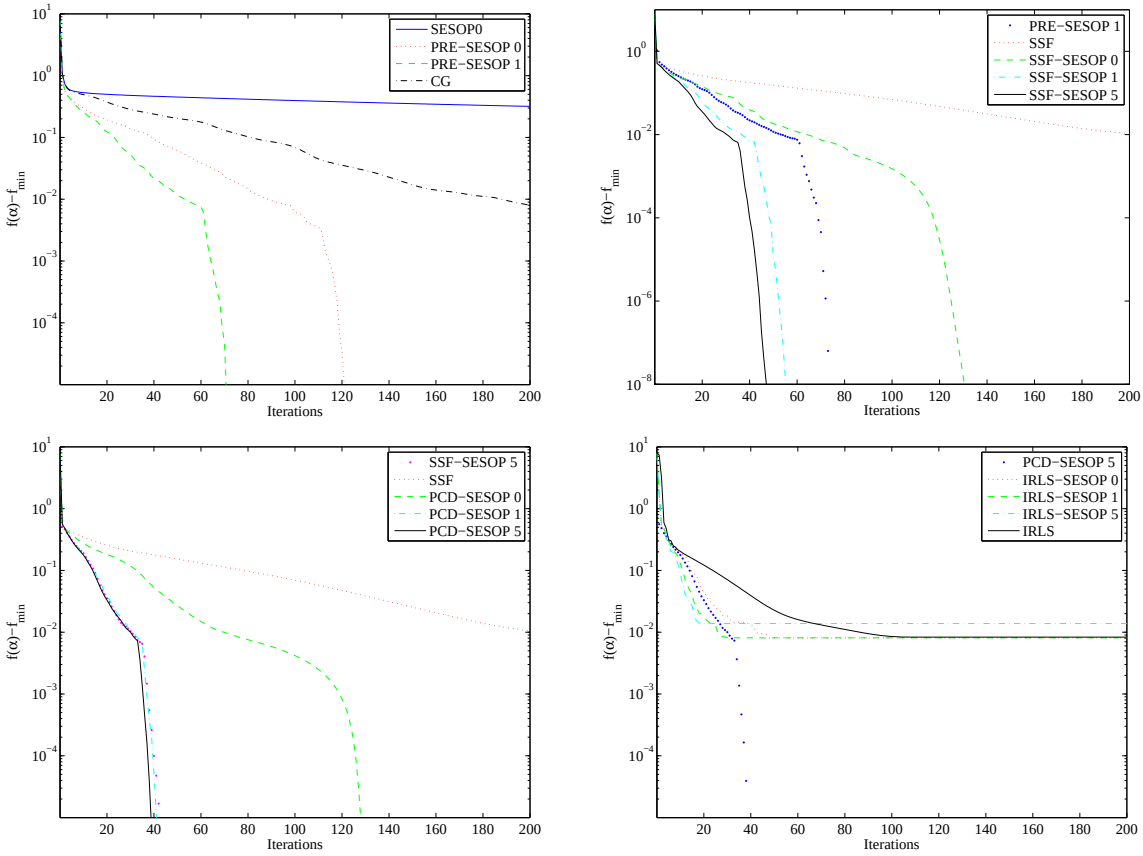


Figure 2. The objective function value as a function of the iterations for the various tested algorithms.

5.2 Part 2: Applications – Image Deblurring

In order to demonstrate the applicative side of the above-discussed algorithms, we present here one application – image deblurring. Following the explanations given in Section 2, we assume that the original signal $\mathbf{x}$ went through a *known* blur operation $\mathbf{H}$, followed by an additive zero-mean white Gaussian noise with *known* variance σ^2. Thus, we measure $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$, and aim to recover $\mathbf{x}$, with the assumption that $\mathbf{x}$ could be described as $\mathbf{D}\alpha$ with a *known* dictionary $\mathbf{D}$ and a sparse vector $\mathbf{D}\alpha$. MAP estimation leads to the need to minimize

$$\hat{\alpha} = \arg\min_{\alpha} \ \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{D}\alpha\|_2^2 + \lambda\|\alpha\|_p^p \ ,$$

and the restored image is $\hat{\mathbf{x}} = \mathbf{D}\hat{\alpha}$.

In the coming experiments we address the above penalty function with the following details:

- **Dictionary:** We use the redundant wavelet transform as in the previous section, using Daubechies' 2-taps filter (Haar), and 3 layers of resolution, leading to a redundancy factor of $7:1$.
- **Blur Kernel:** Following the tests in,[29] we consider a 15×15 kernel with values being $1/(i^2+j^2+1)$ for $-7 \le i, j \le 7$, normalized to have a unit sum.
- **Noise Properties:** We test for two noise powers, $\sigma^2 = 8$, and $\sigma^2 = 2$.
- **Algorithms Tested:** Two algorithms are considered – the StOMP with a stoping rule based on the noise power, and PCD-SESOP0 as in the previous experiment. We do not proceed with the complete comparison as in the previous section, and consider only one relatively simple *Iterated Shrinkage* algorithm that is known to perform relatively well.

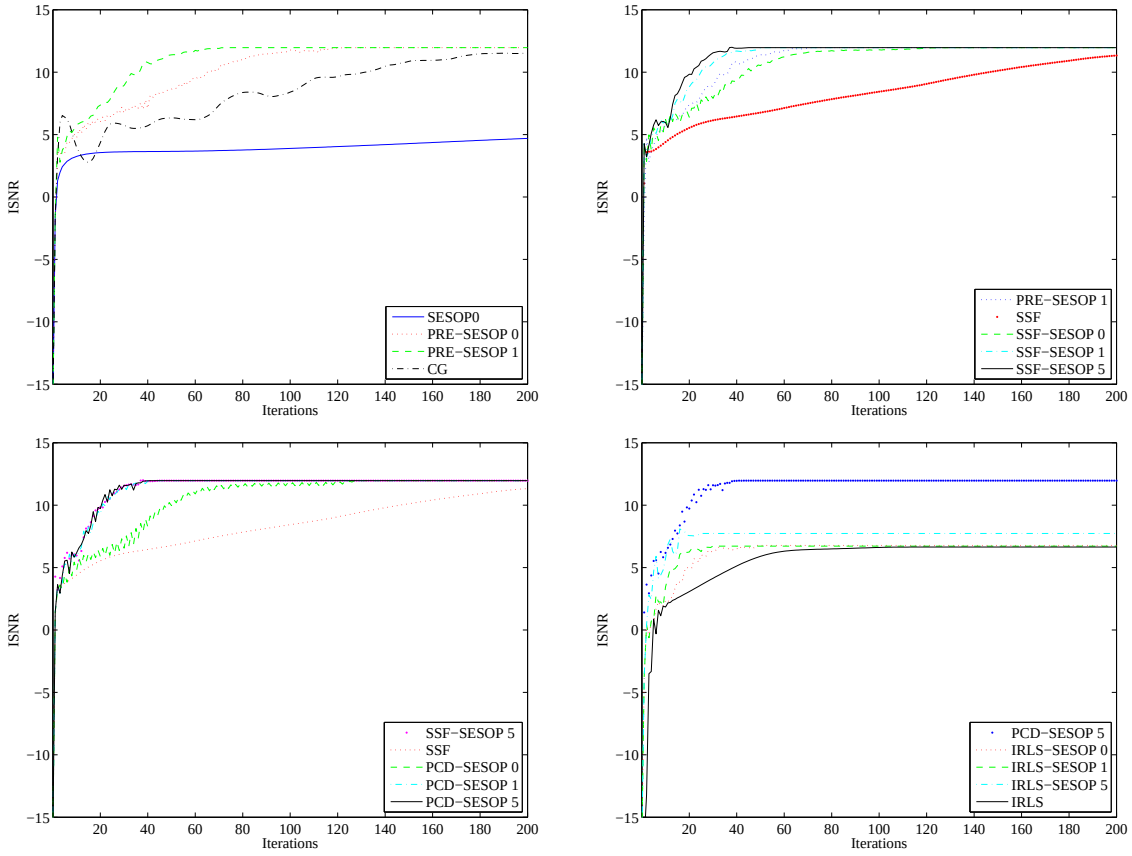


Figure 3. The ISNR as a function of the iterations for the various tested algorithms.

In this paper we do not aim to compete with the existing work on image deblurring, and restrict our view to the problem of optimization involved. Better results could be obtained by optimizing the various parameters of the above experiments, introducing weights to the sparse coefficients, replacing the dictionary with a better one, considering pre-denoising step, changing the sparsity measure function, and more. Nevertheless, as we show herein, the results obtained are close to the best known ones in the literature, demonstrating the effectiveness and relevance of *Iterated Shrinkage* methods.

Table 1 summarizes the results we get, and compares them to those found in.[29] Similar to the conclusions drawn in previous set of synthetic experiments, StOMP is found to be inferior to PCD, even when this algorithm is not activated with it's fastest versions. This inferiority is exhibited both on the final quality and the computational requirements. Finally, we observe that the results reported in[29] are better than those we achieve here, implying that our tests are not an exact replica of theirs, and there is still room for further improvements.

Method	$\sigma^2 = 2$		$\sigma^2 = 8$	
	ISNR	Complexity	ISNR	Complexity
Ref.[29]	7.46 dB	Unknown	5.24 dB	Unknown
PCD-SESOP0	6.97 dB	120 transforms ($\lambda = 0.075$)	5.06 dB	90 transforms ($\lambda = 0.15$)
StOMP	5.96 dB	450 transforms	4.24 dB	374 transforms

Table 1 - Summary of deblurring results. Complexity of the algorithms is presented through a count of the number of multiplications by **HD** and its adjoint.

Figure 4 shows the images obtained by the PCD-SESOP0 method for illustration of the achievable results.

Figure 4. Deblurring results with the PCD-SESOP0 algorithm: Original image (left), measured blurry and noisy image (middle), and restored (right). Top images refer to the experiment with $\sigma^2 = 2$, and the bottom ones for $\sigma^2 = 8$.

As one can see, the results still suffer from ringing artifacts, but the sharpness of the images surpasses that of the measurements.

6. CONCLUSIONS

When considering the minimization of ℓ_2-ℓ_p mixture penalty function with sparse solutions, *Iterated-Shrinkage* algorithms pose an appealing family of techniques, that surpass classical tools. In this paper we reviewed some of the activity in this field, describing several such algorithms, and their comparative performance. This work is far from done, as there are new methods that we did not cover here, and interesting applications that could benefit a lot from these tools, which we have not explored. We hope that this paper (and hopefully a longer version of it to appear in a journal) will stimulate activity on these topics and widespread these algorithm's use in practice.

ACKNOWLEDGEMENTS

The authors would like to thank Mario Figueiredo and his co-authors for sharing their software with us. Special thanks are also in order to Michael Saunders for his help in activating the PDCO solver. We are also thankful to Michael Lustig for fruitful discussions on the various algorithms explored.

REFERENCES

1. T. Adeyemi and M.E. Davies, Sparse representations of images using overcomplete complex wavelets, *IEEE SP 13th Workshop on Statistical Signal Processing*, pages 805–809, Bordeaux, France, 17-20 July 2006.
2. J. Bioucas-Dias, Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors, *IEEE Trans. on Image processing*, 15(4):937–951, April 2006.
3. T. Blumensath and M.E. Davies, Iterative thresholding for sparse approximations, submitted to Journal of Fourier Analysis and Applications.
4. E.J. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, *IEEE Trans. on Information Theory*, 52(2):489–509, 2006.

5. S. Chen, S.A. Billings, and W. Luo, Orthogonal least squares methods and their application to non-linear system identification, *International Journal of Control*, 50(5):1873–96, 1989.
6. S.S. Chen, D.L. Donoho, and M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM Journal on Scientific Computing*, 20(1):33–61 (1998).
7. I. Daubechies, M. Fornasier, and I. Loris, Accelerated projected gradient method for linear inverse problems with sparsity Constraints, arXiv: 0706.4297v1, 28 June, 2007, submitted.
8. I. Daubechies, M. Defrise, and C. De-Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Communications on Pure and Applied Mathematics*, LVII:1413–1457, 2004.
9. G. Davis, S. Mallat, and M. Avellaneda, Adaptive greedy approximations, *Journal of Constructive Approximation*, 13:57–98, 1997.
10. G. Davis, S. Mallat, and Z. Zhang, Adaptive time-frequency decompositions, *Optical-Engineering*, 33(7):2183–91, 1994.
11. A. Dempster, N. Laird, D. Rubin, Maximum likelihood estimation from incomplete data via the EM algorithm, *J. Roy. Statist. Soc. B*, 39:1–38, 1977.
12. R.A. DeVore, B. Jawerth, and B.J. Lucier, Image compression through wavelet transform coding, *IEEE Trans. Information Theory*, 38(2):719–746, 1992.
13. D.L. Donoho, De-noising by soft thresholding, *IEEE Trans. on Information Theory*, 41(3):613–627, 1995.
14. D.L. Donoho, Compressed sensing, *IEEE Trans. on Information Theory*, 52(4):1289–306, April 2006.
15. D.L. Donoho and M. Elad, Optimally sparse representation in general (non-orthogonal) dictionaries via l1 minimization, *Proc. of the National Academy of Sciences*, 100(5):2197–2202, 2003.
16. D.L. Donoho and M. Elad, On the stability of the basis pursuit in the presence of noise, *Signal Processing*, 86(3):511–532, March 2006.
17. D.L. Donoho, M. Elad, and V. Temlyakov, Stable recovery of sparse overcomplete representations in the presence of noise, *IEEE Trans. on Information Theory*, 52(1):6–18, 2006.
18. D.L. Donoho and I.M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, *Biometrika*, 81(3):425–455, 1994.
19. D.L. Donoho, V. Stodden, and Y. Tsaig, About SparseLab, Stanford University, http://sparselab.stanford.edu, Version 2.0, March, 2007.
20. D.L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck, Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit, submitted to *IEEE Trans. on Signal Processing*.
21. B. Efron, T. Hastie, I.M. Johnstone, and R. Tibshirani, Least angle regression, *The Annals of Statistics*, 32(2):407–499, 2004.
22. M. Elad, Why simple shrinkage is still relevant for redundant representations?, to appear in the *IEEE Trans. on Information Theory*.
23. M. Elad and M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, *IEEE Trans. on Image Processing* 15(12):3736–3745, December 2006.
24. M. Elad, B. Matalon, and M. Zibulevsky, Image denoising with shrinkage and redundant representations, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), NY, June 17-22, 2006.
25. M. Elad, B. Matalon, and M. Zibulevsky, Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization, to appear in *Applied and Computational Harmonic Analysis*.
26. M. Elad, J.-L. Starck, P. Querre, and D.L. Donoho, Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA), *Journal on Applied and Comp. Harmonic Analysis*, 19:340–358, 2005.
27. M.J. Fadili and J.L. Starck, Sparse representation-based image deconvolution by iterative thresholding, *Astronomical Data Analysis ADA'06*, Marseille, France, September, 2006.
28. M.A. Figueiredo and R.D. Nowak, An EM algorithm for wavelet-based image restoration, *IEEE Trans. Image Processing*, 12(8):906–916, 2003.
29. M.A. Figueiredo, and R.D. Nowak, A bound optimization approach to wavelet-based image deconvolution, IEEE International Conference on Image Processing - ICIP 2005, Genoa, Italy, 2:782–785, September 2005.
30. M.A. Figueiredo, J.M. Bioucas-Dias, and R.D. Nowak, Majorization-minimization algorithms for wavelet-based image restoration, to appear in the *IEEE Trans. on Image Processing*.

31. M. Figueiredo, R. Nowak, and S. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, submitted, 2007.
32. I.F. Gorodnitsky and B.D. Rao, Sparse signal reconstruction from limited data using FOCUSS: A re-weighted norm minimization algorithm, *IEEE Trans. on Signal Processing*, 45(3):600–616, 1997.
33. Y. Hel-Or and D. Shaked, A Discriminative approach for Wavelet shrinkage denoising submitted to the *IEEE Trans. on Image Processing*.
34. K.K. Herrity, A..C. Gilbert, and J.A. Tropp, Sparse approximation via iterative thresholding, *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2006.
35. D. Hunter, K. Lange, A tutorial on MM algorithms, *Amer. Statist.*, 58:30–37, 2004.
36. S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, A Method for large-scale ℓ_1-regularized least squares problems with applications in signal processing and statistics, submitted.
37. N.G. Kingsbury and T.H. Reeves, Overcomplete image coding using iterative projection-based noise shaping, International Conf. on Image Processing (ICIP) (3):597–600, 2002.
38. K. Lange, D.R. Hunter, I. Yang, Optimization transfer using surrogate objective functions (with discussion), *J. Comput. Graph. Statist*, 9(1):1–59, 2000.
39. I. Loris, G. Nolet, I. Daubechies, and F.A. Dahlen, Tomographic inversion using l1-norm regularization of wavelet coefficients, arXiv:physics/0608094v1, 8 August, 2006, submitted.
40. M. Lustig, D.L. Donoho, and J.M. Pauly, Sparse MRI: The application of compressed sensing for rapid MR imaging, submitted to *Magnetic Resonance in Medicine*.
41. J. Mairal, M. Elad, and G. Sapiro, Sparse representation for color image restoration, submitted to the *IEEE Trans. on Image Processing*.
42. S. Mallat and Z. Zhang, Matching pursuits with time-frequency dictionaries, *IEEE Trans. Signal Processing*, 41(12):3397–3415, 1993.
43. P. Moulin and J. Liu, Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors, *IEEE Trans. on Information Theory*, 45(3):909–919, 1999.
44. G. Narkiss and M. Zibulevsky, Sequential subspace optimization method for large-scale unconstrained optimization, technical report CCIT No. 559, Technion, The Israel Institute of Technology, Haifa, 2005.
45. R.T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM J. Control Optim.*, 14(5): 877–898, 1976.
46. S. Sardy, A.G. Bruce, and P. Tseng, Block coordinate relaxation methods for nonparametric signal denoising with wavelet dictionaries, *Journal of computational and Graphical Statistics*, 9:361–379, 2000.
47. E.P. Simoncelli and E.H. Adelson, Noise removal via Bayesian wavelet coring, Proceedings of the International Conference on Image Processing, Laussanne, Switzerland. September 1996.
48. J.-L. Starck, E.J. Candès, and D.L. Donoho, The curvelet transform for image denoising, *IEEE Trans. on Image Processing*, 11:670–684, 2002.
49. J.-L. Starck, M. Elad, and D.L. Donoho, Redundant multiscale transforms and their application for morphological component separation, *Advances in Imaging And Electron Physics*, 132:287–348, 2004.
50. J.-L. Starck, M. Elad, and D.L. Donoho, Image decomposition via the combination of sparse representations and a variational approach. *IEEE Trans. on Image Processing*, 14(10):1570–1582, 2005.
51. J.A. Tropp, Greed is good: Algorithmic results for sparse approximation, *IEEE Trans. on Information Theory*, 50(10):2231–2242, October 2004.
52. J.A. Tropp, Just relax: Convex programming methods for subset selection and sparse approximation, *IEEE Trans. on Information Theory*, 52(3):1030–1051, March 2006.
53. Y. Tsaig, Sparse Solution Of Underdetermined Linear Systems - Algorithms And Applications, PhD thesis, Stanford, 2007.