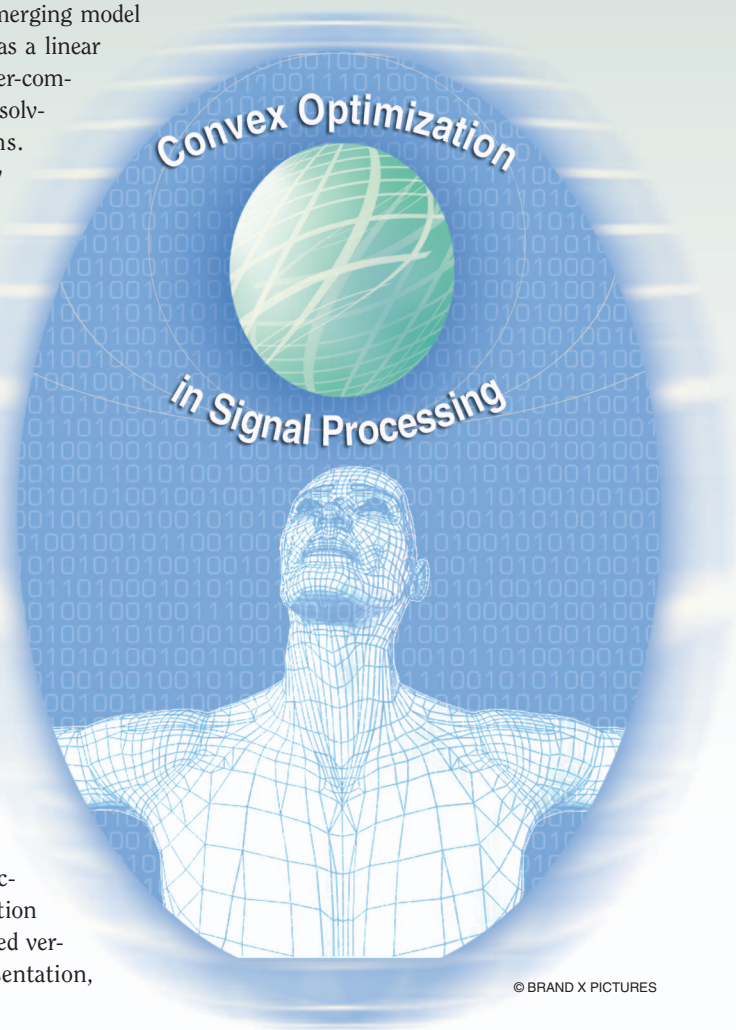
Michael Zibulevsky and Michael Elad

# L1-L2 Optimization in Signal and Image Processing

## Iterative shrinkage and beyond

Sparse, redundant representations offer a powerful emerging model for signals. This model approximates a data source as a linear combination of few atoms from a prespecified and over-complete dictionary. Often such models are fit to data by solving mixed $\ell_1$-$\ell_2$ convex optimization problems. Iterative-shrinkage algorithms constitute a new family of highly effective numerical methods for handling these problems, surpassing traditional optimization techniques. In this article, we give a broad view of this group of methods, derive some of them, show accelerations based on the sequential subspace optimization (SESOP), fast iterative soft-thresholding algorithm (FISTA) and the conjugate gradient (CG) method, present a comparative performance, and discuss their potential in various applications, such as compressed sensing, computed tomography, and deblurring.






### INTRODUCTION

Among the many ways to mathematically model signals, a recent and appealing approach employs sparse and redundant representations (see e.g., [1] and [2]). In this model, the signal of interest $\mathbf{x} \in \mathbb{R}^n$ is assumed to be composed as a linear combination of few atoms from a prespecified and redundant dictionary $\mathbf{A} \in \mathbb{R}^{n \times m}$, i.e.,

$$\mathbf{x} = \mathbf{A}z, \tag{1}$$

where $z \in \mathbb{R}^m$ is expected to be sparse, $\|z\|_0 \ll n$. The $\ell_0$-"norm" notation used here stands for the number of nonzeros in the vector $z$. Given a signal $\mathbf{x}$ and a dictionary $\mathbf{A}$, the atom-decomposition problem aims to find the sparsest representation for $\mathbf{x}$. A relaxed version of this problem permits a small deviation in the representation, leading to the problem

Digital Object Identifier 10.1109/MSP.2010.936023



1053-5888/10/$26.00©2010IEEE

$$\min_{z} \|z\|_0 \text{ subject to } \|\mathbf{x} - \mathbf{A}z\|_2 \leq \epsilon, \tag{2}$$

where $\epsilon$ stands for the permissible deviation of the representation $\mathbf{A}z$ from the original signal $\mathbf{x}$. As this problem is known to be NP-hard [1], approximation algorithms are often considered. One appealing such method is basis-pursuit (BP) [3] or LASSO [4], which replaces the $\ell_0$ by an $\ell_1$-norm. Various encouraging theoretical results on the uniqueness of a sufficiently sparse solution to this problem and equivalence between the sparsest solution and the minimal $\ell_1$-norm solution all suggest that finding the signal's representation $z$ is a computationally feasible task [1], [2], [5]–[8].

On the practical front, a sequence of papers in recent years successfully apply this model to different applications in signal and image processing, such as denoising, inpainting, deblurring, compressed sensing, source-separation, and more, all leading to state-of-the-art results [9]–[12], [14]–[16].

Many of these applications lead to an optimization task that mixes $\ell_2$ and $\ell_1$ expressions in the form

$$f(z) = \frac{1}{2}\|\mathbf{x} - \mathbf{A}z\|_2^2 + \lambda\|z\|_1. \tag{3}$$

In our notation, $f$ is a function of the vector $z$, where $\mathbf{x}$ and $\mathbf{A}$ are assumed to be known. This problem is a variant of the one posed in (2), where the constraint is replaced with a penalty. The parameter $\lambda$ replaces the threshold $\epsilon$ in (2), in governing the tradeoff between the representation error and its sparsity. We should note that while the $\ell_1$-norm is used here to measure the sparsity of $z$, one could use many other similar additive measures $\rho(z)$, for which most of the derivations in this article remain the same (see [17] for more details). In various applications, the desire is to minimize $f(z)$ with respect to $z$. Thus, the use of sparse and redundant representations in these applications is achieved by solving a convex program.

Until recently, this form of optimization problems was traditionally treated using: 1) various classical iterative optimization algorithms, ranging from steepest descent, CG, and iterative reweighed least-squares, and all the way to the more involved interior-point algorithms [3]; 2) homotopy solvers [18], [19]; and 3) greedy techniques [20]. However, these methods are often inefficient, requiring many iterations and excessive central processing unit time to reach their solution. Furthermore, the homotopy and greedy techniques are impractical in high-dimensional problems (e.g., $m = 10^6$), as often encountered in image processing problems.

In recent years, an alternative family of numerical algorithms has gradually built, addressing the above optimization problems very effectively. These emerged initially as a heuristic approach for dealing with the sparse representation problem in a computationally effective way [22]–[24]. These were followed by more thorough derivation and analysis [25]–[33], [17]. This family is the iterative-shrinkage algorithms, which extend the classical Donoho-Johnstone shrinkage method [34]–[36]. Roughly speaking, in these methods, each iteration consists of a multiplication by $\mathbf{A}$ and its adjoint, along with a scalar shrinkage step on the obtained $z$. Despite their simple structure, these algorithms are shown to be very effective in minimizing $f(z)$ in (3). A thorough theoretical analysis proves the convergence of these techniques, guarantees that the solution is the global minimizer for convex $f$, studies the rate of convergence these algorithms exhibit, and proposes speedup techniques for further improvement [26], [37], [17].

## THE NEED: SPARSE AND REDUNDANT REPRESENTATIONS

Suppose we accept the fundamental assumptions of the sparse and redundant representation model, implying that the signal $\mathbf{x}$ emerges from the multiplication $\mathbf{x} = \mathbf{A}z$ with a very sparse $z$. How sparse is "very" sparse? There is no conclusive answer to this question, but we shall assume hereafter that 1) $z$ is the sparsest solution to the linear equation $\mathbf{x} = \mathbf{A}z$, and 2) it is reconstructible via practical algorithms, such as BP [5], [6], i.e., the solution of BP recovers $z$ exactly. 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 independent and identically distributed (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 an independent Laplacian distribution on the entries of $z$, $P(z) \sim \exp\{-\lambda\|z\|_1\}$, the maximum a posteriori probability (MAP) estimation [3] leads this denoising task to optimization problem

$$\hat{z} = \arg\min_{z} \frac{1}{2}\|\mathbf{y} - \mathbf{A}z\|_2^2 + \lambda\|z\|_1, \tag{4}$$

and the result is obtained by $\hat{\mathbf{x}} = \mathbf{A}\hat{z}$. As we can see, the objective function we face is $f(z)$ in (3).

### *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

$$\hat{z} = \arg\min_{z} \frac{1}{2}\|\mathbf{y} - \mathbf{H}\mathbf{A}z\|_2^2 + \lambda\|z\|_1, \tag{5}$$

and the estimated result is $\hat{\mathbf{x}} = \mathbf{A}\hat{z}$, as before. This time, we obtained an objective function exactly as in (3) with the effective dictionary $\mathbf{HA}$. This structure could serve various tasks, such as denoising (where $\mathbf{H} = \mathbf{I}$), deblurring (where $\mathbf{H}$ is the blur operator), inverse-Radon transform in tomography (where $\mathbf{H}$ is the projection operator), and signal interpolation (demosaicing, inpainting), 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 $\mathbf{A}_1$, and $\mathbf{x}_2$ refers to a model that uses $\mathbf{A}_2$. We desire to separate the signal into its three components. MAP is used again to obtain [12]

$$\hat{z}_1, \hat{z}_2 = \arg\min_{z1,z2} \frac{1}{2}\|\mathbf{y} - \mathbf{A}_1 z_1 - \mathbf{A}_2 z_2\|_2^2 + \lambda\|z_1\|_1 + \lambda\|z_2\|_1. \quad (6)$$

The estimated pieces are obtained by $\hat{\mathbf{x}}_1 = \mathbf{A}_1\hat{z}_1$ and $\hat{\mathbf{x}}_2 = \mathbf{A}_2\hat{z}_2$. The above problem, known as morphological component analysis (MCA) [12], fits again the general structure described in (3), using the $z = [z_1^T, z_2^T]^T$ and $[\mathbf{A}_1, \mathbf{A}_2]$ replacing $\mathbf{A}$ in (3).

### *COMPRESSION*

Given a signal $\mathbf{x}$, we aim to compress it by allowing some prespecified $\ell_2$ error $\epsilon$. This implies that we consider its representation $\hat{z}$ that solves

$$\hat{z} = \arg\min_z \|z\|_1 \ \text{ subject to } \ \|\mathbf{x} - \mathbf{A}z\|_2^2 \le \epsilon^2. \quad (7)$$

Instead of solving this problem, we could solve the problem described in (3), while seeking $\lambda$ that satisfies the above constraint with near-equality. The actual compression is done by quantizing the nonzeros in the vector $\hat{z}$ and transmitting these values, along with their indices. Again we find ourselves handling the same format as in (3).

### *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 [14], [15]. Reconstruction of the signal from its samples is obtained by solving

$$\hat{z} = \arg\min_z \frac{1}{2}\|\mathbf{y} - \mathbf{Q}\mathbf{A}z\|_2^2 + \lambda\|z\|_1, \text{ and } \ \tilde{\mathbf{x}} = \mathbf{A}\hat{z}, \quad (8)$$

and again we resort to the need to handle (3).

The bottom line is that the structure described in (3) is fundamental to all these (and other) applications, and thus we are strongly motivated to handle this optimization problem with efficient algorithms.

## THE UNITARY CASE—A SOURCE OF INSPIRATION

### *A CLOSED-FORM SOLUTION FOR THE UNITARY CASE*

If $\mathbf{A}$ is unitary, the minimization of $f(z)$ can be manipulated in a sequence of simple steps and turned into a set of $m$ independent and identical one-dimensional (1-D) optimization tasks, that are easily handled. Starting from (3), and using the identity $\mathbf{A}\mathbf{A}^T = \mathbf{I}$, we get

$$f(z) = \frac{1}{2}\|\mathbf{x} - \mathbf{A}z\|_2^2 + \lambda\|z\|_1 = \frac{1}{2}\|\mathbf{A}(\mathbf{A}^T\mathbf{x} - z)\|_2^2 + \lambda\|z\|_1. \quad (9)$$

Exploiting the fact that $\ell_2$-norm is unitarily invariant, we can remove the multiplication by $\mathbf{A}$ in the first term. Denoting $z_0 = \mathbf{A}^T\mathbf{x}$, we obtain

$$f(z) = \frac{1}{2}\|z - z_0\|_2^2 + \lambda\|z\|_1 = \sum_{k=1}^{m}\left[\frac{1}{2}(z_0[k] - z[k])^2 + \lambda|z[k]|\right]. \quad (10)$$

Minimization of the scalar function $g(\tau) = 0.5(\tau_0 - \tau)^2 + \lambda|\tau|$ with respect to $\tau$ is easily obtained as

$$\tau_{\text{opt}} = \begin{cases} 0 & |\tau_0| \le \lambda \\ \tau_0 - \text{sign}(\tau_0)\lambda & \text{otherwise} \end{cases} = \mathcal{S}_\lambda(\tau_0). \quad (11)$$

This function maps values near the origin to zero, and others are shrunken towards zero, thus the name of this operator.

Back to our original problem, we found a closed-form solution to the minimizer of $f(z)$, with the following two steps: 1) Compute $z_0 = \mathbf{A}^T\mathbf{x}$, and 2) apply $\mathcal{S}_\lambda$ on the entries of $z_0$ and obtain the desired $\hat{z}$. The natural question raised is: when turning from a unitary matrix $\mathbf{A}$ to a nonunitary (and perhaps nonsquare) one, must we lose all this simplicity? To answer this question, we consider first a somewhat more general form of $\mathbf{A}$—a concatenation of unitary matrices, where the shrinkage step is performed iteratively.

### *THE BLOCK-COORDINATE-RELAXATION ALGORITHM*

What happens when $\mathbf{A}$ is built as a union of several unitary matrices? The answer to this was given by Sardy, et al. [38], leading to their block-coordinate-relaxation (BCR) algorithm. We present this method here very briefly, as it is a direct extension of the unitary case solver shown above. For simplicity, we shall assume that $\mathbf{A} = [\mathbf{\Psi}, \mathbf{\Phi}]$, where $\mathbf{\Psi}$ are $\mathbf{\Phi}$ are $n \times n$ unitary matrices. Out minimization can be rewritten with respect to the two $n$-length portions of $z$, denoted as $z_\Psi$ and $z_\Phi$,

$$f(z) = f(z_\Psi, z_\Phi) = \frac{1}{2}\|\mathbf{x} - \mathbf{A}z\|_2^2 + \lambda\|z\|_1 = \frac{1}{2}\|\mathbf{b} - \mathbf{\Psi}z_\Psi - \mathbf{\Phi}z_\Phi\|_2^2 + \lambda\|z_\Psi\|_1 + \lambda\|z_\Phi\|_1. \quad (12)$$

The core idea in the BCR algorithm is to minimize $f(z_\Psi, z_\Phi)$ with respect to the two parts of $z$ separately and alternately, explaining the name BCR. Assume that we hold a current solution at the $k$th iteration, denoted by $z^k$, built of the two portions, $z_\Psi^k$ and $z_\Phi^k$. Assuming that $z_\Phi^k$ is kept fixed, the function $f(z_\Psi, z_\Phi^k)$ can be written as

$$f(z_\Psi, z_\Phi^k) = \frac{1}{2}\|\tilde{\mathbf{x}} - \mathbf{\Psi}z_\Psi\|^2 + \lambda\|z_\Psi\|_1, \quad (13)$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{\Phi} z_{\Phi}^{k}$. This is the same function as in the unitary case, for which a closed-form solution is readily available to us. This solution is given as

**AMONG THE MANY WAYS TO MODEL SIGNALS MATHEMATICALLY, A RECENT AND APPEALING APPROACH EMPLOYS SPARSE AND REDUNDANT REPRESENTATIONS.**

$$z_{\Psi}^{k+1} = \mathcal{S}_{\lambda}(\mathbf{\Psi}^{T}\tilde{\mathbf{x}}) = \mathcal{S}_{\lambda}(\mathbf{\Psi}^{T}(\mathbf{x} - \mathbf{\Phi} z_{\Phi}^{k})). \tag{14}$$

Similarly, once $\mathbf{x}_{\Psi}^{k+1}$ has been computed and now kept fixed, the function $f(\mathbf{x}_{\Psi}^{k+1}, \mathbf{x}_{\Phi})$ is easily minimized with respect to $\mathbf{x}_{\Phi}$ with a closed-form expression, given by

$$\mathbf{x}_{\Phi}^{k+1} = \mathcal{S}_{\lambda}(\mathbf{\Phi}^{T}(\mathbf{b} - \mathbf{\Psi}\mathbf{x}_{\Psi}^{k+1})). \tag{15}$$

This way, alternating between these two update stages, the overall function is monotonically decreasing and proven to converge to a global minimum of the penalty function.

An alternative to the above sequential process can be proposed, where the two updates are done in parallel. As before, we start with a current solution at the $k$th iteration, $z^{k}$, built of the two portions $z_{\Psi}^{k}$ and $z_{\Phi}^{k}$. We can propose an update of the two parts in parallel using

$$\begin{aligned} z_{\Psi}^{k+1} &= \mathcal{S}_{\lambda}(\mathbf{\Psi}^{T}(\mathbf{x} - \mathbf{\Phi} z_{\Phi}^{k})) \\ z_{\Phi}^{k+1} &= \mathcal{S}_{\lambda}(\mathbf{\Phi}^{T}(\mathbf{x} - \mathbf{\Psi} z_{\Psi}^{k})). \end{aligned}$$

The only difference from the previous algorithm is in the update of $\mathbf{x}_{\Phi}^{k+1}$ that uses $\mathbf{x}_{\Psi}^{k}$ instead of $\mathbf{x}_{\Psi}^{k+1}$. Merging these two into one formula can be done by the following simple steps and exploiting the fact that the shrinkage operator operates on scalars independently,

$$\begin{aligned} z^{k+1} &= \mathcal{S}_{\lambda}\left(\begin{bmatrix} \mathbf{\Psi}^{T}(\mathbf{x} - \mathbf{A} z^{k} + \mathbf{\Psi} z_{\Psi}^{k}) \\ \mathbf{\Phi}^{T}(\mathbf{x} - \mathbf{A} z^{k} + \mathbf{\Phi} z_{\Phi}^{k}) \end{bmatrix}\right) \\ &= \mathcal{S}_{\lambda}(\mathbf{A}^{T}(\mathbf{x} - \mathbf{A} z^{k}) + z^{k}). \end{aligned} \tag{16}$$

This way we got an interesting closed-form formula for the update of the entire solution from one iteration to the next. Naturally, we should wonder whether this formula (or a variation of it) could be used for more general matrices $\mathbf{A}$. Surprisingly, the answer is positive, as we show next.

## ITERATIVE-SHRINKAGE ALGORITHMS

There are various iterative-shrinkage algorithms, with important variations between them. Early signs of these methods appear in Jansson's deconvolution [21] for spectroscopic and chromatographic measurement systems, and later in the work by Hoch, et al. in 1990 in the context of maximum-entropy estimation applied to nuclear magnetic resonance spectroscopy [22]. The work by Starck, et al. in 1995 considered the general image deblurring problem with such an algorithm [23]. Kingsbury and Reeves followed in 2002 with a similar approach for complex wavelets [24]. However, all these attempts were done without a clear connection to the objective function in (3).

Iterative-shrinkage algorithms can be derived using very different considerations, such as the expectation-maximization (EM) algorithm in statistical estimation theory [25], proximal point and surrogate-functions [26], [27], [30], the use of the fixed-point strategy [29], employment of a parallel coordinate-descent algorithm [28], [17], variations on greedy methods [39], Bregman iterative regularization [32], and more. This exposes the wealth with which one could design a solver for the above-described optimization task.

In this article, we present two of these algorithms in detail. We deliberately omit a discussion on the theoretical study of these algorithms. Nevertheless, we should state the following main fact: these algorithms are guaranteed to converge to a global minimizer of the function in (3) [26], [17], [31].

As this field of research is very active, new contributions on novel iterative shrinkage methods, their analysis, or applications employing them, are emerging almost on a daily basis. We mention here a partial list of such work [31]–[33], [40] as we will not be able to provide a full and updated picture of this "racing train."

### *SURROGATE FUNCTIONS AND THE PROXIMAL-POINT METHOD*

We start with the proximal (or surrogate) objectives that were used by Daubechies, et al. in their construction [26]. Considering the original function in (3),

$$f(z) = \frac{1}{2}\|\mathbf{x} - \mathbf{A} z\|_{2}^{2} + \lambda\|z\|_{1},$$

let us add to it the following term

$$\text{dist}(z, z_{0}) = \frac{c}{2}\|z - z_{0}\|_{2}^{2} - \frac{1}{2}\|\mathbf{A} z - \mathbf{A} z_{0}\|_{2}^{2}.$$

The parameter $c$ will be chosen such that the function dist is strictly convex (w.r.t. $z$), implying that we require its Hessian to be positive definite: $c\mathbf{I} - \mathbf{A}^{T}\mathbf{A} > 0$. This is satisfied by the choice $c > \|\mathbf{A}^{T}\mathbf{A}\|_{2} = \lambda_{\max}(\mathbf{A}^{T}\mathbf{A})$ (the maximal eigenvalue of the matrix $\mathbf{A}^{T}\mathbf{A}$). This new objective function

$$\begin{aligned} \tilde{f}(z, z_{0}) = &\frac{1}{2}\|\mathbf{x} - \mathbf{A} z\|_{2}^{2} + \lambda\|z\|_{1} \\ &+ \frac{c}{2}\|z - z_{0}\|_{2}^{2} - \frac{1}{2}\|\mathbf{A} z - \mathbf{A} z_{0}\|_{2}^{2}, \end{aligned} \tag{17}$$

is the surrogate objective that will be used in the proposed algorithm. As we shall see next, the fact that the term $\|\mathbf{A} z\|_{2}^{2}$ disappears from $\tilde{f}$, turns the minimization task into a much simpler one. Reorganizing (17), we obtain a new expression of the form

$$\begin{aligned} \tilde{f}(z, z_{0}) = &\text{Const} - z^{T}[\mathbf{A}^{T}(\mathbf{x} - \mathbf{A} z_{0}) + c z_{0}] \\ &+ \lambda\|z\|_{1} + \frac{c}{2}\|z\|_{2}^{2}. \end{aligned} \tag{18}$$

The constant in the above expression and in our later derivations contains all the terms that are dependent on $\mathbf{x}$ and $z_0$ alone. Defining the term

$$\mathbf{v}_0 = \frac{1}{c}\mathbf{A}^T(\mathbf{x} - \mathbf{A}z_0) + z_0, \tag{19}$$

the surrogate objective above can be rewritten as

$$\begin{aligned}\widetilde{f}(z, z_0) &= \text{Const} - cz^T\mathbf{v}_0 + \lambda\|z\|_1 + \frac{c}{2}\|z\|_2^2 \\ &= \text{Const} + \lambda\|z\|_1 + \frac{c}{2}\|z - \mathbf{v}_0\|_2^2.\end{aligned} \tag{20}$$

Notice the resemblance between this formulation and the one obtained in the unitary case, in (10). This immediately implies that the minimizer of the surrogate objective is

$$z_{\text{opt}} = \mathcal{S}_{\lambda/c}(\mathbf{v}_0) = \mathcal{S}_{\lambda/c}\left(\frac{1}{c}\mathbf{A}^T(\mathbf{x} - \mathbf{A}z_0) + z_0\right). \tag{21}$$

So far we managed to convert the original function $f$ to a new function $\widetilde{f}$, 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 $z_0$. The core idea of using the surrogate objective is that we minimize the function $f$ iteratively, producing the sequence of results $\{z_i\}_i$, where at the $i+1$th iteration we minimize $\widetilde{f}$ with the assignment $z_0 = z_i$. Interestingly, it can be shown that the sequence of solutions $\{z_i\}_i$ converges to the minimizer of the original function $f$. Thus, the proposed algorithm is simply given by

$$z_{i+1} = \mathcal{S}_{\lambda/c}\left(\frac{1}{c}\mathbf{A}^T(\mathbf{x} - \mathbf{A}z_i) + z_i\right). \tag{22}$$

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 [41], a well-known method in optimization theory [17]. The function $\text{dist}(z, z_0)$ 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 a slowing-down effect, in the case discussed here it is actually enabling a substantial speed-up relatively to the conventional gradient descent.

The discussion above could be replaced with a different set of considerations, and yet, one that leads to the very same algorithm. This alternative perspective appears in the work of Figueiredo, et al. that uses the expectation-maximization (EM) estimator, or better yet, its alternative deterministic optimization foundation—the bound-optimization method [25], [27], [30]. This technique is also known as the majorization-minimization method.

Starting from the function $f(z)$ in (3), which is hard to minimize, the bound-optimization method suggests to use a related function $Q(z, z_0)$ that has the following properties:

1) equality at $z = z_0$: $Q(z_0, z_0) = f(z_0)$
2) upper bounding the original function: $Q(z, z_0) \geq f(z)$ for all $z$.

Under certain regularity conditions, the sequence of solutions generated by the recurrent formula

$$z_{i+1} = \arg\min_z Q(z, z_i) \tag{23}$$

converges to a local minimum of the original function $f(z)$ [42], [43], [17]. As can be easily verified, the choice from the previous subsection, $Q(z, z_0) = f(z) + \text{dist}(z, z_0)$ satisfies the above two conditions, and as such, leads to the desired optimization algorithm. Note that minimization of $Q(\cdot)$ is easy due to its separability.

### THE PARALLEL COORDINATE DESCENT ALGORITHM

We now describe the parallel coordinate descent (PCD) algorithm [28], [17]. It starts from a simple coordinate descent algorithm, but merges a set of such descent steps into one easier joint step, leading to the PCD iterative shrinkage method. The rationale practiced here should remind the reader of the BCR algorithm mentioned previously, and the parallel-update modification we proposed for it.

Returning to $f(z)$ in (3), we can propose a coordinate descent (CD) algorithm that updates $z$ one entry at a time. A sequence of such rounds of $m$ steps (addressing each coordinate of $z \in \mathbb{R}^m$) is necessarily converging. Interestingly, as we are about to show, each of these steps is obtained via shrinkage, similar to the process described above.

Assuming that the current solution is $z_0$, we desire to update the $k$th entry around its current value $z_0[k]$. This leads to a 1-D function of the form

$$g(\gamma) = \frac{1}{2}\|\mathbf{x} - \mathbf{A}z_0 - \mathbf{a}_k(\gamma - z_0[k])\|_2^2 + \lambda|\gamma|. \tag{24}$$

The vector $\mathbf{a}_k$ is the $k$th column in $\mathbf{A}$. The term $\mathbf{a}_k(\gamma - z_0[k])$ removes the effect of the old value and adds the new one. Several simple algebraic steps leads to an equivalent expression

$$g(\gamma) = \frac{1}{2}\|\mathbf{a}\|_2^2 \cdot (\gamma - \gamma_0)^2 + \lambda|\gamma|, \tag{25}$$

where we have defined

$$\gamma_0 = \frac{\mathbf{a}_k^T(\mathbf{x} - \mathbf{A}z_0)}{\|\mathbf{a}_k\|_2^2} + z_0[k].$$

As before, we find ourselves with a formulation similar to the one found in (10), which implies that the optimal value for $z[k]$ is

$$z_k^{\text{opt}} = \mathcal{S}_{\lambda/\|\mathbf{a}_k\|_2^2}\left(\frac{\mathbf{a}_k^T(\mathbf{x} - \mathbf{A}z_0)}{\|\mathbf{a}_k\|_2^2} + z_0[k]\right). \tag{26}$$

While using this update rule and sweeping through the entries of $z$ repeatedly may work well in low-dimensional cases, it is impractical in some of the most important applications, which are based on attacking very large problem sizes ($> 10^6$) using implicit representations of the matrix $\mathbf{A}$. In such applications, $\mathbf{A}$

is never explicitly available, but instead one has available function calls that can rapidly apply $\mathbf{A}$ or $\mathbf{A}^T$ to appropriate vectors. Equation (27) requires to explicitly have specific columns of $\mathbf{A}$ available on demand, and is inappropriate for use in such implicit-operator settings.

Thus, we consider a modification of the above method. We rely on the following property: When minimizing a convex smooth function, if there are several descent directions departing from the same point, then any nonnegative combination of them is also a descent direction. As we show in the section "Modified Penalty Functions," we smooth the $\ell_1$-term and thus we can assume that the objective is smooth. Thus, we propose a simple addition of these $m$ steps in (27). Since each of these steps handles one entry in the destination vector, we can write this sum as

$$\begin{aligned}\mathbf{v}_0 &= \sum_{k=1}^{m} \mathbf{e}_k \cdot \mathcal{S}_{\lambda/\|\mathbf{a}_k\|_2^2}\left(\frac{\mathbf{a}_k^T(\mathbf{x}-\mathbf{A}z_0)}{\|\mathbf{a}_k\|_2^2} + z_0[k]\right) \\ &= \mathcal{S}_{\mathbf{W}\lambda}(\mathbf{W}\mathbf{A}^T(\mathbf{x}-\mathbf{A}z_0) + z_0), \end{aligned} \tag{27}$$

where $\mathbf{W} = \mathrm{diag}(\mathbf{A}^T\mathbf{A})^{-1}$. This term consists of the norms of the columns of the dictionary $\mathbf{A}$. These are used both for the weighting of the back-projected error $\mathbf{A}^T(\mathbf{x}-\mathbf{A}z_0)$ and for the shrinkage operator. This set of weights can be computed offline, before the algorithm starts, and there are fast ways to approximate them [17]. Notice that the obtained formula does not call for an extraction of columns from $\mathbf{A}$ as in (27), and the operations required are a direct multiplication of $\mathbf{A}$ and its adjoint by vectors and in (23).

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 [28], we consider this direction and perform a line search along it. This means that the actual iterative algorithm is of the form

$$z_1 = z_0 + \mu(\mathbf{v}_0 - z_0), \tag{28}$$

with $\mu$ chosen by a line-search algorithm of some sort. This requires another multiplication by $\mathbf{A}$. This overall process will be referred to hereafter as the PCD algorithm.

Compared to the SSF algorithm in (23), the PCD differs in two ways: 1) the norms of the atoms in $\mathbf{A}$ play an important role in weighting the back-projected error, whereas the previous algorithm uses a constant; and 2) the new algorithm requires a line-search to obtain a descent direction. Recall that the constant $c$ in SSF must satisfy $c > \lambda_{\max}(\mathbf{A}^T\mathbf{A})$. On the other hand, the PCD uses a scaling of the same term using the weight matrix $\mathrm{diag}(\mathbf{A}^T\mathbf{A})^{-1}$. As an example, for a dictionary built as a union of $N$ unitary matrices, the SSF requires $c > N$, implying a weight of $1/N$ in (23). In the PCD algorithm, the weight matrix is simply the identity due to the normalized columns in $\mathbf{A}$. Thus, PCD gives $O(N)$ times stronger weight to the term $\mathbf{A}^T(\mathbf{x}-\mathbf{A}z_i)$, and because of that we expect it to perform better.

There is another interpretation of the above shrinkage method, worth mentioning. In the realm of solving least-squares (LS) problems, the Gauss-Seidel algorithm can be interpreted as a sequential CD solver. Alternatively, one can compute the coordinate descent steps for all coordinates without moving away from the current solution and merge all into one step—this results with the well-known Jacobi method for linear systems, which parallels the PCD developed here.

## ACCELERATION TECHNIQUES

### *LINE SEARCH*

The above algorithms can be further accelerated in several ways. First, the idea of performing a line search as described for the PCD is relevant to the SSF algorithm as well. All that is required is to use (23) to compute a temporary result $\mathbf{v}_i$, and then define the solution as $z_{i+1} = z_i + \mu(\mathbf{v}_i - z_i)$, optimizing $f(z_{i+1})$ with respect to the scalar $\mu$.

### *SEQUENTIAL SUBSPACE OPTIMIZATION*

A second and much more effective speed-up option for these algorithms is the deployment of the SESOP method [44]. The story of SESOP begins with the CG method [45]. Quadratic CG (i.e., CG applied to a quadratic function) has remarkable convergence properties. Its linear convergence rate (see for example, [46]) is $(\sqrt{r}-1)/(\sqrt{r}+1)$, where $r$ is the condition number of the Hessian of the objective. This rate is much better than the steepest descent rate, $(r-1)/(r+1)$.

One can also rely on a $1/i^2$ sublinear worst-case convergence of the quadratic CG, which does not depend on the Hessian conditioning (see, for example, [47], [48], and [44])

$$f(z_{i+1}) - f_{\text{optimal}} \le \frac{L\|z_0 - z_{\text{optimal}}\|^2}{i^2}, \tag{29}$$

where $i$ is the iteration index and $L$ is the Lipschitz constant of the gradient of $f$. The presented convergence rates are intimately related to the well-known expanding manifold property of quadratic CG: At every iteration the method minimizes the objective function over an affine subspace spanned by directions of all previous propagation steps and gradients.

In the case of a smooth convex function (not necessarily quadratic), one could propose a similar algorithm that preserves the expanding manifold property. Such an algorithm should minimize the objective function over an affine subspace spanned by directions of all previous propagation steps and the latest gradient. This method inherits the $1/i^2$ convergence of CG, however, the cost of an iteration of such a method will increase with iteration count.

To alleviate this problem, Nemirovski [47] suggested to restrict the optimization subspace just to three directions: the current gradient, the sum of all previous steps, and a "cleverly" weighted sum of all previous gradients (a version of such weights is given in [44]). The resulting ORTH-method inherits

the optimal worst case $1/i^2$ convergence (29), but it does not coincide with CG, when the objective is quadratic, and typically converges slower than CG, when the function become "almost" quadratic in the neighborhood of solution.

The SESOP method [44] extends the ORTH subspaces with several directions of the last propagation steps. This way, the method, while preserving a $1/i^2$ convergence for smooth convex functions, becomes equivalent to the CG in the quadratic case. This property boosts the efficiency of the algorithm.

In [17], SESOP was merged with iterative-shrinkage algorithms, by substituting the current gradient direction with the direction of the shrinkage step. In the quadratic case, the resulting PCD-SESOP method is equivalent to the diagonally preconditioned CG. On the other hand, the PCD and SSF directions provide much faster progress at initial steps, when compared to the ordinary nonlinear CG. This partially explains extremal efficiency of these methods on difficult problems, presented in the experimental section below.

The low-dimensional subspace optimization task at every iteration of SESOP can be addressed using the Newton algorithm. The main computational burden in this process is the need to multiply the spanning directions by $\mathbf{A}$, but these multiplications can be stored in previous iterations, thus enabling the SESOP speed-up with hardly any additional cost. In our experience, the use of additional ORTH directions does not improve practical convergence speed, and therefore, we do not use them in our simulations.

### NESTEROV METHODS AND FISTA

In 1983, Nesterov [49] suggested a method with $1/i^2$ rate of convergence of type (29), which stands as an alternative to ORTH. Nesterov's method combines gradient directions with results of previous iterations in a "clever" way that does not require a subspace optimization or a line search. On the other hand, it is typically slower than CG in the quadratic case; experiments in [44] show the advantage of SESOP over Nesterov's approach.

Recently in [50], Nesterov generalized his method to the case of an objective function consisting of two convex terms. One term is smooth, with a limited Lipschitz constant of its gradient, and the second one is easily optimized but can be nonsmooth. This situation resembles the $\ell_2$-$\ell_1$ objective function considered in this article. Using SSF-type steps (instead of gradients), combined with results of previous iterations, [50] presents a method without a line-search, with a $1/i^2$ convergence rate as in (29), where the Lipschitz constant of the gradient is determined by the smooth term only. Along similar lines, an alternative method, termed FISTA, was developed in [33]. Put simply, FISTA applies an iteration step very similar to the SSF, as given in (22)

$$z_{i+1} = \mathcal{S}_{\lambda/c}\left(\frac{1}{c}\mathbf{A}^T(\mathbf{x} - \mathbf{A}z_i) + z_i\right).$$

However, instead of applying this formula on $z_i$, it is applied on a modified vector $\hat{z}_i$ given by

$$\hat{z}_i = z_i + \frac{t_i - 1}{t_i + 1}(z_i - z_{i-1}),$$

where $z_0 = 0$ and $t_i = (1 + \sqrt{1 + 4t_{i-1}^2})/2$ with $t_1 = 1$. We present tests of FISTA in the experimental section that follows.

### MERGING WITH CONJUGATE GRADIENTS

The last acceleration option we mention here is a generalization of the Polak-Ribiere CG method [46], which may incorporate either the PCD or the SSF directions. Starting with the search direction obtained by PCD (or SSF, for that matter), $\mathbf{d}^0 = \mathbf{d}^0_{\text{pcd}}$, at the $i$th iteration we compute $z^{i+1}$ via an exact line-search along the direction $\mathbf{d}^i$ and then prepare the next direction as $\mathbf{d}^{i+1} = \mathbf{d}^{i+1}_{\text{PCD}} + \beta_i \mathbf{d}^i$, where

$$\beta_i = \frac{\nabla f(z^{i+1})^T(\mathbf{d}^{i+1}_{\text{PCD}} - \mathbf{d}^i_{\text{PCD}})}{\nabla f(z^i)^T \mathbf{d}^i_{\text{PCD}}}.$$

Here $\mathbf{d}^i_{\text{PCD}}$ is a PCD direction at point $z^i$. We call this method PCD-CG. In the case of a quadratic objective function, PCD-CG is equivalent to the diagonally preconditioned CG method, and thus its fast asymptotic convergence. Global convergence of PCD-CG and SSF-CG have not been studied yet.

## MODIFIED PENALTY FUNCTIONS

Just before we turn to demonstrate the above algorithms in experiments, we return to the definition of the objective function (3) and discuss its generalization. In this function we can substitute the $\ell_1$-norm with a general penalty function

$$f_s(z) = \frac{1}{2}\|\mathbf{x} - \mathbf{A}z\|_2^2 + \lambda \sum_k \varphi_s(z[k]), \tag{30}$$

where $\varphi_s(\tau)$ should be chosen such that 1) it serves the application in mind better while 2) giving an analytical closed-form expression for the scalar nonlinearity that implements the shrinkage operation as in (11).

We start with a smooth and convex choice for $\varphi_s$, which tends to accelerate convergence. While the natural choice for such a function is the replacement of the $|\tau|$ by $\varphi_s(\tau) = |\tau|^p, p > 1$, this does not lead to a closed-form expression for the scalar nonlinearity [36]. An appealing alternative is given by [17]

$$\varphi_s(\tau) = |\tau| - s\ln(1 + |\tau|/s), \;\; s \in (0, \infty), \tag{31}$$

which approximates well $|\tau|^p$, up to a scaling factor that depends on $s$. This choice of $\varphi(\tau)$ gives the following antisymmetric scalar nonlinearity

$$\mathcal{S}_{\lambda,s}(\tau_0) = \frac{|\tau_0| - \lambda - s + \sqrt{(|\tau_0| - \lambda - s)^2 + 4s|\tau_0|}}{2\text{sign}(\tau_0)}. \tag{32}$$

Turning to concave penalty choices of $\varphi(\tau)$, a function like $|\tau|^p$, $0 < p < 1$ leads to smaller distortion of large coefficients, thereby enabling a "refined" optimization that

further sparsifies the solution, while allowing large nonzero entries in it. Again, as this option does not permit a closed-form expression for the scalar nonlinearity, we can use the logarithmic function

$$\varphi_s(\tau) = s\ln(1+|\tau|/s),\ s\in(0,\infty), \tag{33}$$

or its smoothed alternative

$$\varphi_{\mathrm{cp}}(\tau) = \frac{\ln(1+|\tau|/s)}{\ln(1+1/s)-\frac{1}{p}\ln(1+p/s)} - \frac{1}{p}\ln(1+p|\tau|/s),$$

where $p$ and $s$ control the smoothness at the origin and the concavity. Both functions allow the derivation of a closed-form scalar nonlinearity.

## AN EXPERIMENTAL STUDY

In this section, we present four sets of experiments. The first corresponds to image deblurring, the second to tomographic reconstruction, the third discusses compressed sensing, and the fourth is an experiment adopted from [51], using a synthetic matrix $\mathbf{A}$ termed geoGaussian. Common to all these tests is the desire to minimize the function posed in (3), or more precisely, its smoothed version given in (31) and (32) with $s = 0.001$.

The algorithms we explore in these tests are the SSF [using $c = \lambda_{\max}(\mathbf{A}^T\mathbf{A})$] and the PCD, their accelerated versions with CG and SESOP (using seven dimensions, and up to seven Newton steps per update), the CG, and the SESOP algorithms directly, the L-BFGS algorithm, FISTA [33], and the L1-LS interior-point method [40]. For the L-BFGS algorithm, we used the MATLAB `MINFUNC` package by Mark Schmidt.

We now turn to present the four experiments and their results.

### *IMAGE DEBLURRING*

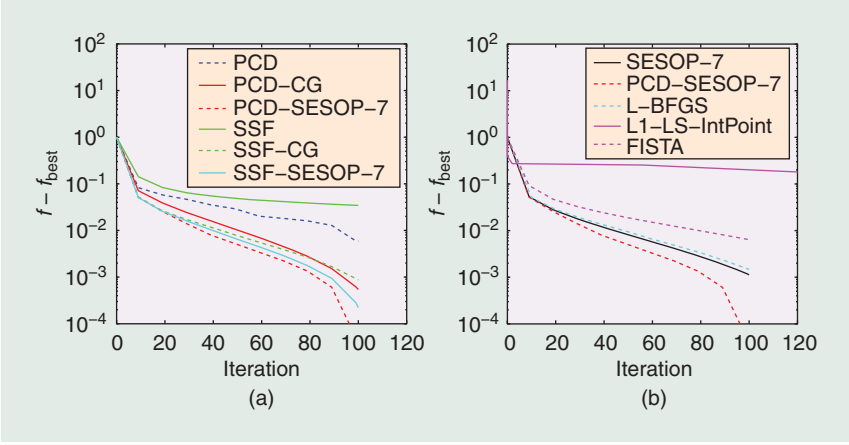
Following the explanations given in the section "The Need: Sparse and Redundant Representations," we assume that the original image $\mathbf{x}$ (these are the actual pixels of the ideal image) went through a known blur operation $\mathbf{H}$, followed by an additive noise with known variance $\sigma^2$. Thus, we measure the blurred and noisy image $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$, and aim to recover $\mathbf{x}$, with the assumption that $\mathbf{x}$ could be described as $\mathbf{A}z$ with a known dictionary $\mathbf{A}$ and a sparse vector $z$. Our goal is to minimize

$$\hat{z} = \arg\min_z \frac{1}{2}\|\mathbf{y}-\mathbf{H}\mathbf{A}z\|_2^2 + \lambda\|z\|_1,$$

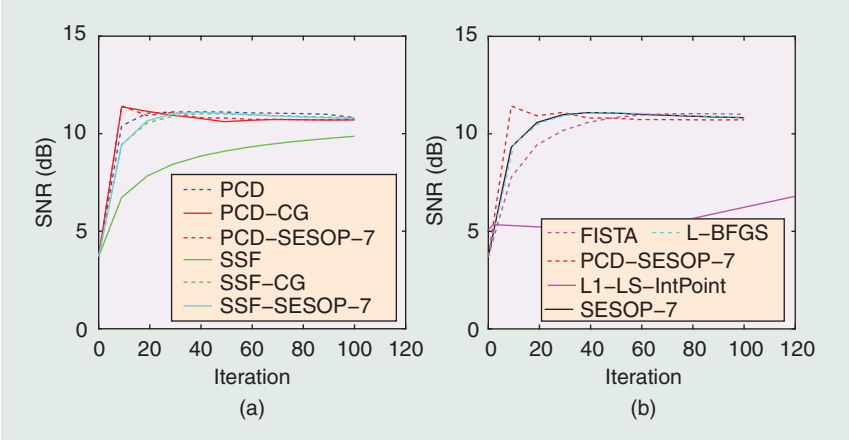
and the restored image is $\hat{\mathbf{x}} = \mathbf{A}\hat{z}$. The ingredients of this penalty function are:

- Original image: We experiment on the image `Phantom` of size $128 \times 128$ pixels, as shown in Figure 3.
- Dictionary: We use the redundant (undecimated) wavelet transform using Daubechies' four-taps filter and four layers of resolution, leading to a redundancy factor of ten to one.
- Blur kernel: Following the tests in [27], we consider a $15 \times 15$ 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: The additive noise is white zero-mean Gaussian noise with $\sigma = 0.01$, where the image pixels are in the range $[0, 1)$.

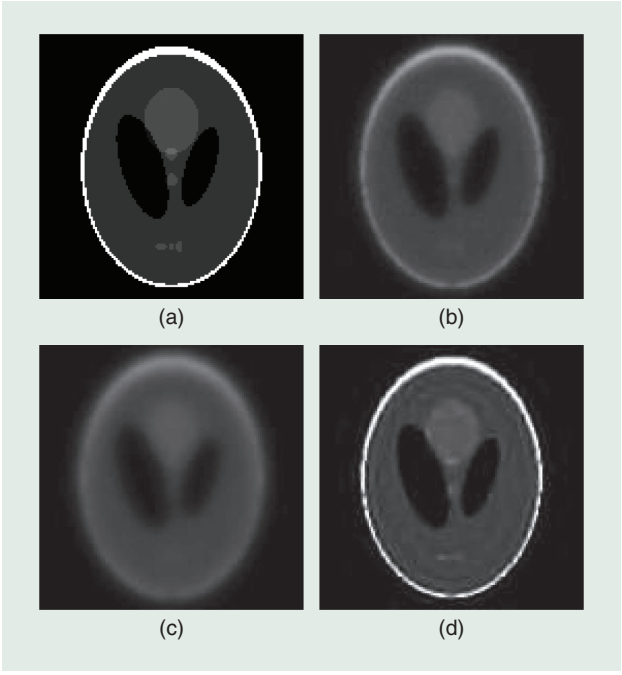
Figures 1–3 show the results obtained for $\lambda = 0.001$. This value was found manually to lead to satisfactory results. Automatic tuning of this parameter for best performance is possible, using the generalized Stein unbiased risk estimator (GSURE) method (see [52] and [53]). Figure 1 presents the value of the objective for various tested algorithms. As can be



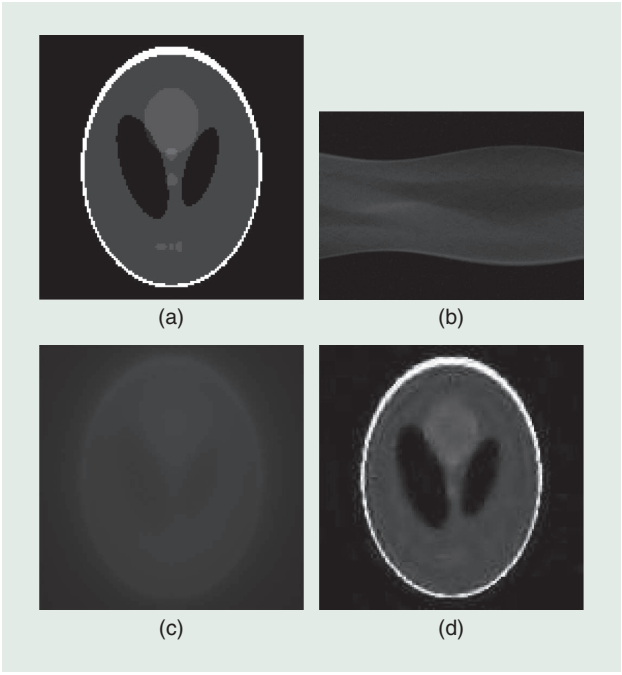

**[FIG1] Deblurring: The objective function value as a function of iteration number for the various tested algorithms.**




**[FIG2] Deblurring: The SNR as a function of iteration number for the various tested algorithms.**




[FIG3] Deblurring: (a) The original image, (b) the observed image, (c) the initialization image, and (d) the recovered image obtained by 20 iterations of the PCD-SESOP-7 algorithm.



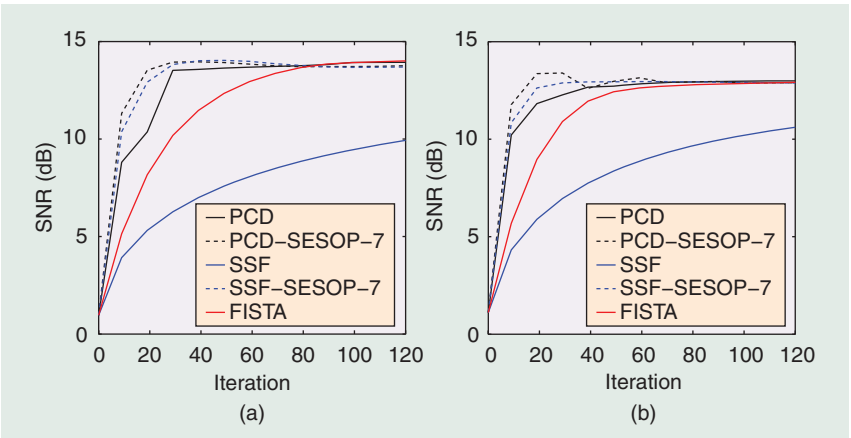

[FIG4] Tomography: (a) The original image, (b) the observed projections, (c) the initialization image, and (d) the recovered image obtained by 20 iterations of the PCD-SESOP-7 algorithm.

seen, while the core PCD and SSF algorithms are slow, their accelerated versions are much better, and in fact the best among the group of methods explored. The PCD and SSF with the SESOP acceleration (employing a subspace of seven dimensions) converge faster than the CG algorithm, the L-BFGS, and the L1-LS and FISTA. Note that the L1-LS method uses a sequence of inner-CG steps and thus the wide gap between the markings in the graph.

Figure 2 addresses the true goal of our experiment—the recovery of the original image. We measure the quality of a proposed solution by signal-to-noise ratio (SNR),

$$\mathrm{SNR} = 10 \cdot \log_{10}\left(\frac{\|\mathbf{x}\|_2^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}\right). \tag{34}$$

The higher this value is, the better the quality of the solution. The results again point to the accelerated PCD (with SESOP-7 or CG) as the best algorithms, both giving a high-quality outcome after few iterations. Since there are methods that automatically stop when the SNR is at its peak [52], [53], this implies that in terms of the output quality, we need not iterate beyond the tenth iteration in this case.




[FIG5] Tomography: The effect of the signal dimension. In (a), the graph corresponds to the tomography experiment applied on the image `Phantom` of size 32 × 32 pixels. In (b), the graph corresponds to a higher resolution version of this image of size 256 × 256 pixels.

Figure 3 shows the original image, `Phantom`, on which this test was performed. The figure also shows the observed image $\mathbf{y}$, the initialization image $\mathbf{A}\hat{z}_0$, and the PCD-SESOP-7 result after 20 iterations. In all the experiments, we initialize by $\hat{z}_0 = \mathrm{const} \cdot \mathbf{A}^T\mathbf{H}^T\mathbf{y}$, where the constant is chosen as the maximal singular value of $\mathbf{HD}$ for proper normalization. As one can see, while the result still suffers from some artifacts, its sharpness surpasses that of the measurement by far.

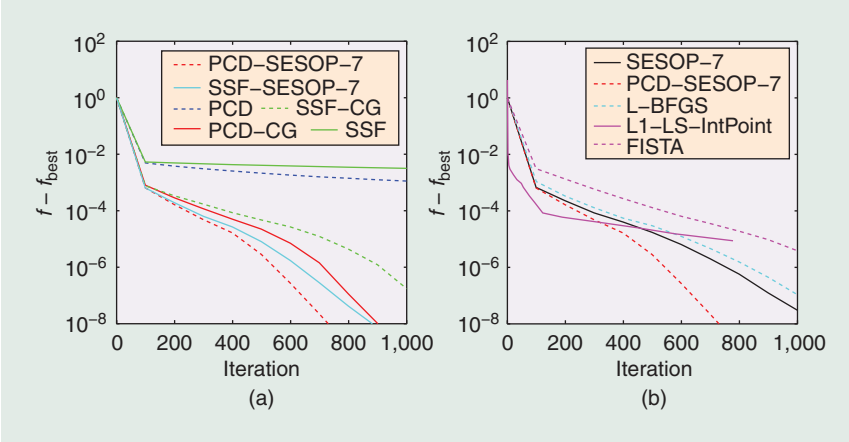
### TOMOGRAPHIC RECONSTRUCTION

Operating on the same image `Phantom`, using the same dictionary, and following the same formulation as above, we now turn to the second experiment, where the

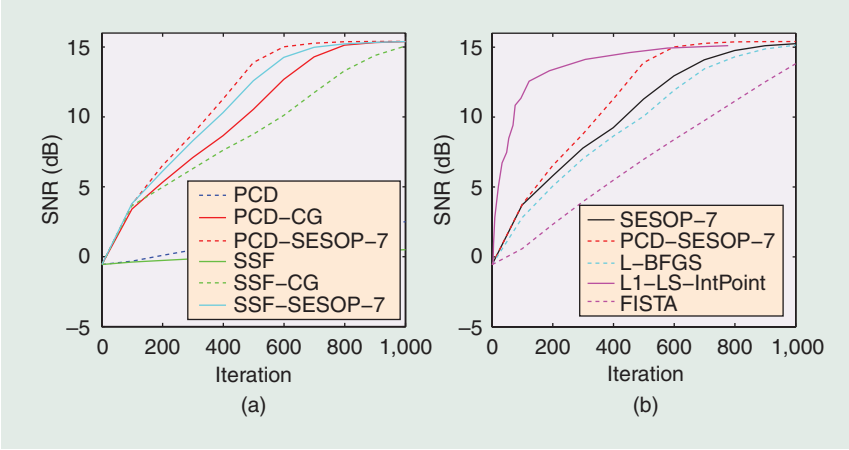
degradation operator $\mathbf{H}$ consists of the Radon-projection. This operator produces 256 line-projections, each containing $\sqrt{2} \times 128 = 181$ scalars, and $\lambda$ is chosen as 0.001 as before.

Figure 4 shows the results obtained, following the same structure as in Figure 3. We omit the graphs showing the value of the objective and the SNR as functions of the iteration, as they are very similar to those seen in the deblurring test (see Figures 1 and 2).

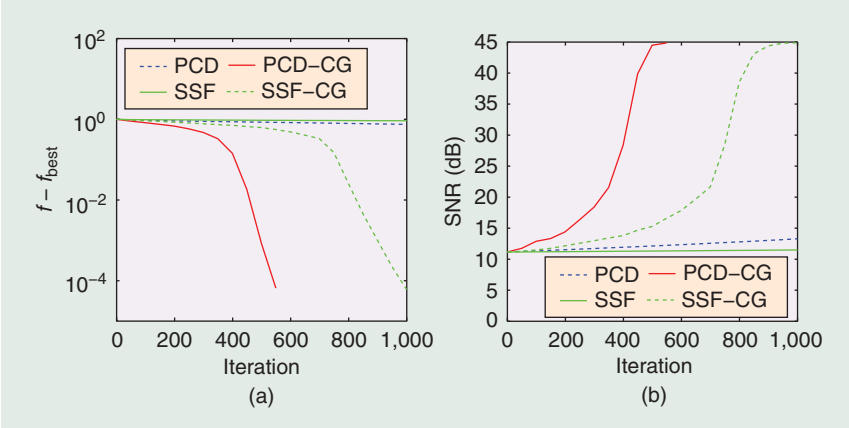
Figure 5 demonstrates a different aspect of iterative-shrinkage algorithms—their tendency to be robust to the dimension of the signal treated. Building on the same tomography experiment, we perform two similar tests, one with an image of size $32 \times 32$ pixels and the other with a much larger image of size $256 \times 256$ pixels. Figure 5 summarizes the results obtained for a selected set of algorithms, and as can be seen, the behavior is nearly identical. This implies that the same number of iterations is required to recover the image, regardless of its dimension. This also suggests that the complexity of these recovery processes remains linear with respect to the number of unknowns.




**[FIG6] Compressed sensing: The objective function value as a function of iteration number for the various tested algorithms.**




**[FIG7] Compressed sensing: The SNR as a function of iteration number for the various tested algorithms.**




**[FIG8] Compressed sensing: The objective function value and the SNR as a function of iteration number for the various tested algorithms, using the concave penalty term.**

## *COMPRESSED SENSING*

Our third experiment corresponds to the recovery of an image from random measurements of it, also known as compressed sensing. As opposed to the previous two experiments, we operate on a synthetic sparse image $\mathbf{x}$ of size $64 \times 64$ pixels, with 5% of its pixels (in random locations) being nonzeros, drawn from an i.i.d. Gaussian distribution $\mathcal{N}(0, 10)$. Such an image may correspond to actual images obtained in astronomy or in radar imaging applications. Since the signal in question is sparse, the dictionary we use in our formulation is the trivial identity operator, $\mathbf{A} = \mathbf{I}$.

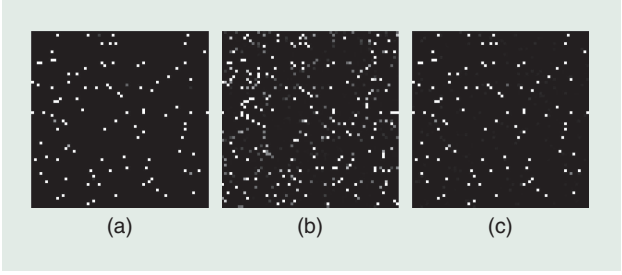
The degradation operator $\mathbf{H}$ (or $\mathbf{Q}$ to better fit the description in the section "The Need: Sparse and Redundant Representations") consists of 410 projections, obtained by choosing at random 410 samples of the two-dimensional-fast Fourier transform applied on $\mathbf{x}$. Thus, there are twice as many measurements in $\mathbf{y}$, compared to the cardinality in $\mathbf{x}$, from which we aim to recover $\mathbf{x}$. The values in $\mathbf{y}$ are further contaminated by additive noise with $\sigma = 0.01$.

Figures 6 and 7 show the results obtained for $\lambda = 0.001$, following the same structure as in the previous experiments. While most of the algorithms behave as before, the L1-LS shows a marked difference, being highly effective and fast in getting a

high-quality recovered result. The reason for the very different behavior of the L1-LS in this simulation, as compared to the two previous tests, is the following: In cases where the number of nonzero coefficients in $z$ is relatively small (as in this simulation), L1-LS is expected to be very effective, because the Hessian of the log-barrier function becomes a low-rank matrix, with a rank corresponding to the number of active constraints (number of nonzero coefficients in the signal). In this case, the inner CG method for solving the Newton system becomes very effective.

**ITERATIVE-SHRINKAGE ALGORITHMS CONSTITUTE A NEW FAMILY OF EFFECTIVE NUMERICAL METHODS FOR HANDLING THESE PROBLEMS, SURPASSING TRADITIONAL OPTIMIZATION TECHNIQUES.**

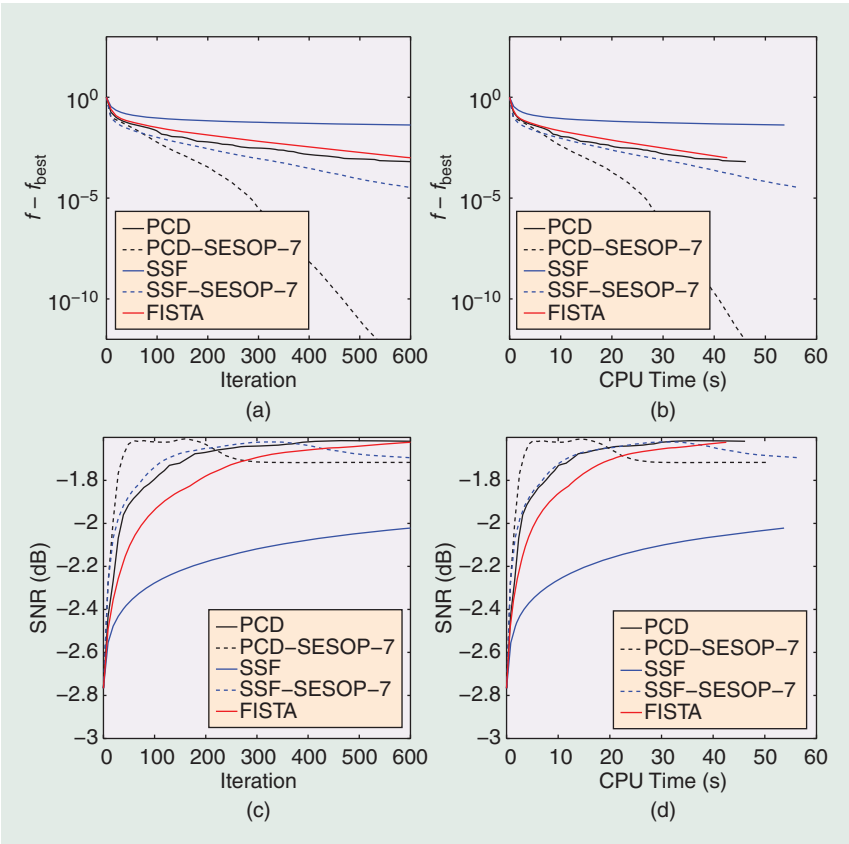
The obtained results can be dramatically improved by replacing the $\ell_1$ term in the objective function by a concave penalty (34) with parameter $s = 1$. This promotes further sparsity in the outcome, with the hope to get closer to the ideal image x. We use the result of the PCD-SESOP-7 with 1,000 iterations from the previous round as the initialization for the same group of algorithms. Figure 8 shows the objective function and the SNR for this refinement round, with a clear and substantial improvement obtained by the PCD-CG. Figure 9 shows the original image, the initialization to this refinement stage, and the resulting image, which appears as a near-perfect recovery of the desired image.




**[FIG9] Compressed sensing: (a) The original image, (b) the image obtained by 20 iterations of the PCD-SESOP-7 algorithm, and (c) the recovered image obtained using the concave objective function, employing 500 iterations of the PCD-CG.**

### *SYNTHETIC EXPERIMENT WITH LORIS DATA*

The fourth and last experiment we bring in this article is adopted from [51]. This problem uses an explicit matrix H of size 1,848 × 8,192, termed $K_4$. This matrix is built from a Gaussian random matrix, by shaping its singular values to fit a classical and highly ill-conditioned geoscience problem (known as $K_1$). As for the compressed sensing case, the ideal signal $z^*$ is created synthetically, as a vector of length 8,192 with 5% of its entries (in random locations) being nonzeros, drawn from an i.i.d. Gaussian distribution $\mathcal{N}(0, 10)$. The dictionary in this case is the identity matrix, and our goal is the recovery of $z^*$ from a noisy measurements $\mathbf{H}z^* + \mathbf{v}$, where $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \sigma\mathbf{I})$ with $\sigma = 1e-4$. In [51], this is considered to be the most difficult problem handled.




**[FIG10] The $K_4$ experiment: Using $\lambda = 1e-3$.**

Beyond the natural goal of comparing various algorithms, in this experiment we aim to address the following additional issues: 1) run time and its relation to number of iterations and 2) the behavior of the explored algorithms for very small value of $\lambda$, for which iterative-shrinkage algorithms are known to deteriorate. Figure 10 presents the objective function and he SNR, both as a function of the iteration count, and as a function of time, all for $\lambda = 1e-3$. Figure 11 presents the results of the same tests applied for $\lambda = 1e-6$.

As we can see, FISTA converges much faster than SSF, which corresponds to the observations in [51]. On the other hand, PCD-SESOP and SSF-SESOP are

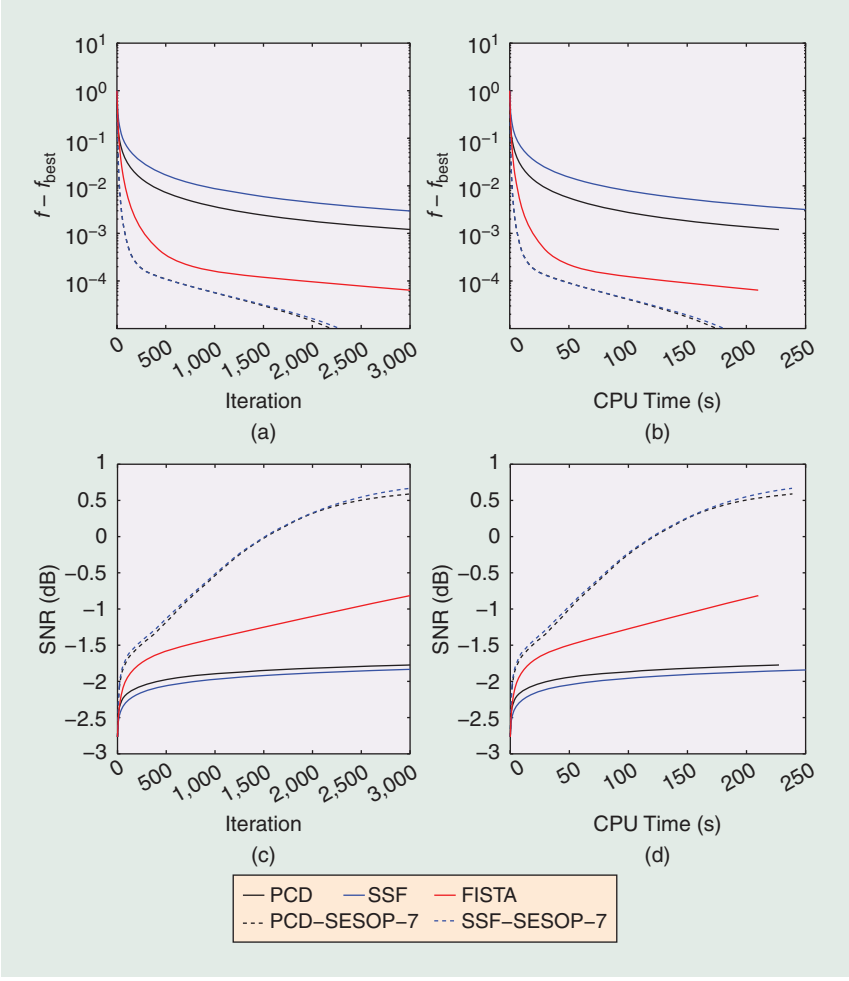
far superior to all other methods. There is no fast matrix multiplication in this problem, and therefore the iteration cost is dominated by products of vectors with A, and a single SESOP iteration is about as fast as one iteration of FISTA.

We also performed experiments (not shown here) with matrices $K_2$ and $K_3$ from [51], and observed the same relation between the algorithms, as seen above.

## CONCLUSIONS

When considering the minimization of an $\ell_2$-$\ell_1$ mixture penalty function with sparse solutions, iterative-shrinkage algorithms pose an appealing family of techniques that may surpass classical tools. In this article, we reviewed the activity in this field, described several such algorithms, explored ways to further accelerate these methods, and studied their comparative performance versus their leading competitors.

This article presented a set of experiments that test these algorithm for practical applications. These include image deblurring, tomographic reconstruction, compressed sensing of sparse images, and a test borrows from [51], which imitates a highly ill-conditioned inverse problem from geophysics. These experiments all indicated that iterative shrinkage methods are indeed very effective, and especially so if merged with SESOP acceleration. Among the various methods explored, the PCD-SESOP performs the best in most of the tests. FISTA, a recently introduced alternative acceleration algorithm, may become faster than the SESOP approach in cases where the operator A is implicit and has a fast deployment algorithm.




[FIG11] The $K_4$ experiment: Using $\lambda = 1e-6$.

The work on this breed of algorithms is far from done, as there are new methods to propose, various analysis questions that remain open, and interesting applications that could greatly benefit from these tools, which have not yet been explored. We hope that this article will stimulate the readers to further study these topics, and expand the knowledge and capabilities of this special branch of convex optimization. To support such an activity, we released a complete MATLAB package that reproduces all the figures in this article. This package is freely accessed for download at http://ie.technion.ac.il/~mcib/ under the name "iterative-shrinkage package."

## ACKNOWLEDGMENTS

This research was partly supported by the Israel Science Foundation grant number 1031/08, and by the European Community's FP7-FET program, SMALL project, under grant agreement number 225913. We would like to thank Mario Figueiredo and Ignace Loris for sharing their software packages with us. We are also thankful to Michael Lustig for fruitful discussions on the various algorithms explored.

## AUTHORS

*Michael Zibulevsky* (mzib@cs.technion.ac.il) received his M.Sc. degree in electrical engineering from MIIT, Moscow, and the D.Sc. degree in operations research (nonlinear optimization) from Technion, Israel, in 1996. He spent two years with the University of New Mexico working on blind source separation via sparse signal representations, forming the foundation to this field. He is with the Department of Computer Science at Technion. His area of interests include nonlinear optimization, signal/image processing, computer tomography, ultrasound imaging, and EEG/MEG. He is also active in the field of sparse approximation and its application to image processing.

*Michael Elad* (elad@cs.technion.ac.il) received his B.Sc, M.Sc., and D.Sc. degrees in 1986, 1988, and 1997, respectively, from the Department of Electrical Engineering at Technion,

Israel. Since 2003, he has been a faculty member in the Computer Science Department at Technion. He works in the field of signal and image processing, specializing in particular on sparse representations and over-complete transforms. He received Technion's Best Lecturer Award six times, the 2007 Solomon Simon Mani Award for excellence in teaching, and the 2008 Henri Taub Prize for academic excellence. He is an associate editor for *IEEE Transactions on Image Processing* and *SIAM Journal on Imaging Sciences*. He is a Senior Member of IEEE.

## REFERENCES


[1] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[2] S. Mallat, *A Wavelet Tour of Signal Processing—The Sparse Way*, 3rd ed. Burlington, MA: Academic, 2009.

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[4] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. R. Statist. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[5] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization," *Proc. Nat. Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, 2003.

[6] D. L. Donoho and M. Elad, "On the stability of the basis pursuit in the presence of noise," *Signal Process.*, vol. 86, no. 3, pp. 511–532, Mar. 2006.

[7] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[8] J. A. Tropp, "Just relax: Convex programming methods for subset selection and sparse approximation," *IEEE Trans. Inform. Theory*, vol. 52, no. 3, pp. 1030–1051, Mar. 2006.

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

[10] J. Bioucas-Dias, "Bayesian wavelet-based image deconvolution: A GEM algorithm exploiting a class of heavy-tailed priors," *IEEE Trans. Image Processing*, vol. 15, no. 4, pp. 937–951, Apr. 2006.

[11] M. J. Fadili and J. L. Starck, "Sparse representation-based image deconvolution by iterative thresholding," in *Proc. Astronomical Data Analysis ADA'06*, Marseille, France, Sept. 2006.

[12] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *J. Appl. Comput. Harmon. Anal.*, vol. 19, no. 3, pp. 340–358, 2005.

[13] J. Bobin, J.-L. Starck, J. Fadili, Y. Moudden, and D. L. Donoho, "Morphological component analysis: An adaptive thresholding strategy," *IEEE Trans. Image Processing*, vol. 16, no. 11, pp. 2675–2681, 2007.

[14] E. J. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, Dec. 2006.

[15] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[16] M. Lustig, D. L. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007.

[17] M. Elad, B. Matalon, and M. Zibulevsky, "Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization," *Appl. Comput. Harmon. Anal.*, vol. 23, pp. 346–367, Nov. 2007.

[18] B. Efron, T. Hastie, I. M. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[19] Y. Tsaig and D. L. Donoho, "Fast solution of $\ell_1$-norm minimization problems when the solution may be sparse," *IEEE Trans. Inform. Theory*, vol. 54, no. 11, pp. 4789–4812, Nov. 2008.

[20] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[21] P. A. Jansson, *Deconvolution: With Applications in Spectroscopy*. New York: Academic, 1984.

[22] J. C. Hoch, A. S. Stern, D. L. Donoho, and I. M. Johnstone, "Maximum-entropy reconstruction of complex (phase-sensitive) spectra," *J. Magn. Reson.*, vol. 86, no. 2, pp. 236–246, Feb. 1990.

[23] J.-L. Starck, F. Murtagh, and A. Bijaoui, "Multiresolution support applied to image filtering and restoration," *Graph. Models Image Processing*, vol. 57, no. 5, pp. 420–431, Sept. 1995.

[24] N. G. Kingsbury and T. H. Reeves, "Overcomplete image coding using iterative projection-based noise shaping," in *Proc. Int. Conf. Image Processing (ICIP)*, 2002, vol. 3, pp. 597–600.

[25] M. A. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[26] I. Daubechies, M. Defrise, and C. De-Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[27] M. A. Figueiredo and R. D. Nowak, "A bound optimization approach to wavelet-based image deconvolution," in *Proc. IEEE Int. Conf. Image Processing—ICIP 2005*, Genoa, Italy, Sept. 2005, vol. 2, pp. 782–785.

[28] M. Elad, "Why simple shrinkage is still relevant for redundant representations?," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5559–5569, Dec. 2006.

[29] T. Adeyemi and M. E. Davies, "Sparse representations of images using overcomplete complex wavelets," in *Proc. IEEE SP 13th Workshop Statistical Signal Processing*, Bordeaux, France, July 17–20, 2006, pp. 805–809.

[30] M. A. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak, "Majorization-minimization algorithms for wavelet-based image restoration," *IEEE Trans. Image Processing*, vol. 16, no. 12, pp. 2980–2991, 2007.

[31] T. Blumensath and M. E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal. Applicat.*, vol. 14, no. 5–6, pp. 629–654, Dec. 2008.

[32] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for $\ell_1$-minimization with applications to compressed sensing," *SIAM J. Imaging Sci.*, vol. 1, no. 1, pp. 143–168, 2008.

[33] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[34] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[35] E. P. Simoncelli and E. H. Adelson, "Noise removal via Bayesian wavelet coring," in *Proc. Int. Conf. Image Processing*, Laussanne, Switzerland, Sept. 1996, pp. 379–383.

[36] P. Moulin and J. Liu, "Analysis of multiresolution image denoising schemes using generalized Gaussian and complexity priors," *IEEE Trans. Inform. Theory*, vol. 45, no. 3, pp. 909–919, 1999.

[37] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM J. Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, Nov. 2005.

[38] S. Sardy, A. G. Bruce, and P. Tseng, "Block coordinate relaxation methods for nonparametric signal denoising with wavelet dictionaries," *J. Comput. Graph. Statist.*, vol. 9, no. 2, pp. 361–379, 2000.

[39] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal.*, vol. 26, no. 3, pp. 301–321, May 2009.

[40] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "A Method for large-scale $\ell_1$-regularized least squares problems with applications in signal processing and statistics," *IEEE J. Select. Topics Signal Processing*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[41] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. Control Optim.*, vol. 14, no. 5, pp. 877–898, 1976.

[42] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *J. R. Statist. Soc. Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[43] K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions (with discussion)," *J. Comput. Graph. Statist.*, vol. 9, no. 1, pp. 1–59, 2000.

[44] G. Narkiss and M. Zibulevsky, "Sequential subspace optimization method for large-scale unconstrained optimization," *Technion, Israel Inst. Technol.*, Haifa, Tech. Rep. CCIT 559, 2005.

[45] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Natl. Bur. Stand.*, vol. 49, no. 6, pp. 409–436, 1952.

[46] J. Nocedal and S. Wright, *Numerical Optimization*. New York: Springer, 1999.

[47] A. Nemirovski, "Orth-method for smooth convex optimization," (in Russian), *Izvestia AN SSSR, Ser. Tekhnicheskaya Kibernetika*, vol. 2, 1982 (Transl.: Eng. Cybern. Soviet J. Comput. Syst. Sci.).

[48] A. S. Nemirovskii and D. B. Yudin, *Problem Complexity and Method Efficiency*. New York: Wiley, 1983.

[49] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," (in Russian), *Dokl. Akad. Nauk SSSR*, vol. 269, pp. 543–547, 1983.

[50] Y. E. Nesterov. (2007). *Gradient methods for minimizing composite objective function. CORE Report* [Online]. Available: http://www.ecore.be/DPs/dp_1191313936.pdf

[51] I. Loris, "On the performance of algorithms for the minimization of $\ell_1$-penalized functionals," *Inverse Probl.*, vol. 25, no. 3, pp. 1–16, 2009.

[52] C. Vonesch, S. Ramani, and M. Unser, "Recursive risk estimation for nonlinear image deconvolution with a wavelet-domain sparsity constraint," in *Proc. 15th IEEE Int. Conf. Image Processing (ICIP)*, 2008, pp. 665–668.

[53] R. Giryes, M. Elad, and Y. C. Eldar, "The projected GSURE for automatic parameter tuning in iterative shrinkage methods," submitted for publication.