Fast Image Recovery Using Variable Splitting and Constrained Optimization

Manya V. Afonso, José M. Bioucas-Dias, *Member, IEEE*, and Mário A. T. Figueiredo, *Fellow, IEEE*

Abstract—We propose a new fast algorithm for solving one of the standard formulations of image restoration and reconstruction which consists of an unconstrained optimization problem where the objective includes an ℓ_2 data-fidelity term and a nonsmooth regularizer. This formulation allows both wavelet-based (with orthogonal or frame-based representations) regularization or total-variation regularization. Our approach is based on a variable splitting to obtain an equivalent constrained optimization formulation, which is then addressed with an augmented Lagrangian method. The proposed algorithm is an instance of the so-called *alternating direction method of multipliers*, for which convergence has been proved. Experiments on a set of image restoration and reconstruction benchmark problems show that the proposed algorithm is faster than the current state of the art methods.

Index Terms—Augmented Lagrangian, compressive sensing, convex optimization, image reconstruction, image restoration, inverse problems, total variation, variable splitting, wavelets.

I. INTRODUCTION

A. Problem Formulation

I MAGE restoration/reconstruction is one of the earliest and most classical linear inverse problems in imaging, dating back to the 1960s [1]. In this class of problems, a noisy indirect observation $\mathbf{y}$, of an original image $\mathbf{x}$, is modeled as

$$\mathbf{y} = \mathbf{B}\mathbf{x} + \mathbf{n}$$

where $\mathbf{B}$ is the matrix representation of the direct operator and $\mathbf{n}$ is noise. As is common, we are adopting the vector notation for images, where the pixels on an $M \times N$ image are stacked into a an (NM)-vector in, e.g., lexicographic order. In the sequel, we denote by n the number of elements of $\mathbf{x}$, thus, $\mathbf{x} \in \mathbb{R}^n$, while $\mathbf{y} \in \mathbb{R}^m$ (m and n may or may not be equal).

In the particular case of image deblurring/deconvolution, $\mathbf{B}$ is the matrix representation of a convolution operator; if this convolution is periodic, $\mathbf{B}$ is then a (block) circulant matrix. This type of observation model describes well several physical mechanisms, such as relative motion between the camera and the subject (motion blur), bad focusing (defocusing blur), or a number of other mechanisms which are well modeled by a convolution. In more general image reconstruction problems, $\mathbf{B}$ represents some linear direct operator, such as a set of tomographic projections (Radon transform), a partially observed (e.g., Fourier) transform, or the loss of part of the image pixels.

It is well known that the problem of estimating $\mathbf{x}$ from $\mathbf{y}$ is ill-posed; thus, this inverse problem can only be solved satisfactorily by adopting some sort of regularization (or prior information, in Bayesian inference terms). One of the standard formulations of wavelet-based, regularization of image restoration/reconstruction problems is built as follows. Let the unknown image $\mathbf{x}$ be represented as a linear combination of the elements of some frame, i.e., $\mathbf{x} = \mathbf{W}\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^d$, and the columns of the $n \times d$ matrix $\mathbf{W}$ are the elements of a wavelet[1] frame (an orthogonal basis or a redundant dictionary). Then, the coefficients of this representation are estimated from the noisy image, under one of the well-known sparsity inducing regularizers, such as the ℓ_1 norm (see [15], [18], [21]–[23], and further references therein). Formally, this leads to the following optimization problem:

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{B}\mathbf{W}\boldsymbol{\beta} - \mathbf{y}\|_2^2 + \tau\,\phi(\boldsymbol{\beta}) \tag{1}$$

where $\phi : \mathbb{R}^d \to \bar{\mathbb{R}}$, usually called the *regularizer* or *regularization function*, is often nonsmooth, or maybe even nonconvex, and $\tau \geq 0$ is the regularization parameter. This formulation is referred to as the *synthesis approach* [19], since it is based on a synthesis equation where $\mathbf{x}$ is synthesized from its representation coefficients ($\mathbf{x} = \mathbf{W}\boldsymbol{\beta}$) which are the object of the estimation criterion. Of course, the final image estimate is computed as $\widehat{\mathbf{x}} = \mathbf{W}\widehat{\boldsymbol{\beta}}$.

An alternative formulation applies a regularizer directly to the unknown image, leading to criteria of the form

$$\widehat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{B}\,\mathbf{x} - \mathbf{y}\|_2^2 + \tau\,\phi(\mathbf{x}) \tag{2}$$

where $\phi : \mathbb{R}^n \to \bar{\mathbb{R}}$ is the regularizer. This type of criteria are usually called *analysis approaches*, since they are based on a regularizer that analyzes the image itself, $\phi(\mathbf{x})$, rather than the coefficients of a representation thereof. Arguably, the

[1] We adopt the generic term "wavelet" to mean any wavelet-like multiscale representation, such as "curvelets," "beamlets," "ridgelets."

best known and most often used regularizer used in analysis approaches to image restoration is the total variation (TV) norm [11], [40]. Wavelet-based analysis approaches are also possible [19], but will not be considered in this paper.

Finally, it should be mentioned that problems (1) and (2) can be seen as the Lagrangians of constrained optimization problems: (1) is the Lagrangian of the constrained problem

$$\min_{\boldsymbol{\beta}} \phi(\boldsymbol{\beta}) \quad \text{subject to} \quad \|\mathbf{B}\mathbf{W}\boldsymbol{\beta} - \mathbf{y}\|_2^2 \leq \varepsilon \qquad (3)$$

while (2) is the Lagrangian of

$$\min_{\mathbf{x}} \phi(\mathbf{x}) \quad \text{subject to} \quad \|\mathbf{B}\mathbf{x} - \mathbf{y}\|_2^2 \leq \varepsilon. \qquad (4)$$

Specifically, a solution of (3) (for any ε such that this problem is feasible) is either the null vector, or else is a minimizer of (1), for some $\tau > 0$ (see [39, Theorem 27.4]). A similar relationship exists between problems (2) and (4).

B. Previous Algorithms

For any problem of nontrivial dimension, matrices $\mathbf{B}\mathbf{W}$, $\mathbf{B}$, and $\mathbf{W}$ cannot be stored explicitly, and it is costly, even impractical, to access portions (lines, columns, blocks) of them. On the other hand, matrix-vector products involving $\mathbf{B}$ or $\mathbf{W}$ (or their conjugate transposes $\mathbf{B}^H$ and $\mathbf{W}^H$) can be done quite efficiently. For example, if the columns of $\mathbf{W}$ contain a wavelet basis or a tight wavelet frame, any multiplication of the form $\mathbf{W}\mathbf{v}$ or $\mathbf{W}^H\mathbf{v}$ can be performed by a fast wavelet transform algorithm [34]. Similarly, if $\mathbf{B}$ represents a circular convolution, products of the form $\mathbf{B}\mathbf{v}$ or $\mathbf{B}^H\mathbf{v}$ can be performed with the help of the fast Fourier transform (FFT) algorithm. These facts have stimulated the development of special purpose methods, in which the only operations involving $\mathbf{B}$ or $\mathbf{W}$ (or their conjugate transposes) are matrix-vector products.

To present a unified view of algorithms for handling (1) and (2), we write them in a common form

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \tau\,\phi(\mathbf{x}) \qquad (5)$$

where $\mathbf{A} = \mathbf{B}\mathbf{W}$, in the case of (1), while $\mathbf{A} = \mathbf{B}$, for (2).

Arguably, the standard algorithm for solving problems of the form (5) is the so-called *iterative shrinkage/thresholding* (IST) algorithm. IST can be derived as an expectation-maximization (EM) algorithm [22], as a *majorization-minimization* (MM, [29]) method [15], [23], or as a forward-backward splitting technique [13], [27]. A key ingredient of IST algorithms is the so-called shrinkage/thresholding function, also known as the Moreau proximity mapping [13] or the denoising function, associated with the regularizer ϕ, which provides the solution of the corresponding pure denoising problem. Formally, this function is denoted as $\boldsymbol{\Psi}_{\tau\phi} : \mathbb{R}^m \longrightarrow \mathbb{R}^m$ and defined as

$$\boldsymbol{\Psi}_{\tau\phi}(\mathbf{y}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \tau\phi(\mathbf{x}). \qquad (6)$$

Notice that if ϕ is proper and convex, the function being minimized is proper and strictly convex, thus, the minimizer exists and is unique making the function well defined [13].

For some choices of ϕ, the corresponding denoising functions $\boldsymbol{\Psi}_{\tau\phi}$ have well-known closed forms. For example, choosing $\phi(\mathbf{x}) = \|\mathbf{x}\|_1 = \sum_i |x_i|$ (the ℓ_1 norm) leads to $\boldsymbol{\Psi}_{\tau\ell_1}(\mathbf{y}) = \text{soft}(\mathbf{y}, \tau)$, where $\text{soft}(\cdot, \tau)$ denotes the component-wise application of the function $y \mapsto \text{sign}(y)\max\{|y| - \tau, 0\}$.

If $\phi(\mathbf{x}) = \|\mathbf{x}\|_0 = |\{i : x_i \neq 0\}|$, usually referred to as the ℓ_0 "norm" (although it is not a norm), despite the fact that this regularizer is not convex, the corresponding shrinkage/thresholding function also has a simple closed form: the so-called hard-threshold function, $\boldsymbol{\Psi}_{\tau\ell_0}(\mathbf{y}) = \text{hard}(\mathbf{y}, \sqrt{2\tau})$, where $\text{hard}(\cdot, a)$ denotes the component-wise application of the function $y \mapsto y 1_{|y|\geq a}$. A comprehensive coverage of *Moreau proximal maps* can be found in [13].

Each IST iteration for solving (5) is given by

$$\mathbf{x}_{k+1} = \boldsymbol{\Psi}_{\tau\phi}\left(\mathbf{x}_t - \frac{1}{\gamma}\,\mathbf{A}^H\left(\mathbf{A}\mathbf{x}_k - \mathbf{y}\right)\right) \qquad (7)$$

where $1/\gamma$ is a step size. Notice that $\mathbf{A}^H(\mathbf{A}\mathbf{x}_k - \mathbf{y})$ is the gradient of the data-fidelity term $(1/2)\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$, computed at $\mathbf{x}_k$. Thus, each IST iteration takes a step of length $1/\gamma$ in the direction of the negative gradient of the data-fidelity term, followed by the application of the shrinkage/thresholding function associated with the regularizer ϕ.

It has been shown that if $\gamma > \|\mathbf{A}\|_2^2/2$ and ϕ is convex, the algorithm converges to a solution of (5) [13]. However, it is known that IST may be quite slow, specially when τ is very small and/or the matrix $\mathbf{A}$ is very ill-conditioned [4], [5], [21], [27]. This observation has stimulated work on faster variants of IST, which we will briefly review in the next paragraphs.

In the *two-step IST* (TwIST) algorithm [5], each iterate depends upon the two previous iterates, rather than only on the previous one (as in IST). This algorithm may be seen as a nonlinear version of the so-called two-step methods for linear problems [2]. TwIST was shown to be considerably faster than IST on a variety of wavelet-based and TV-based image restoration problems; the speed gains can reach up to two orders of magnitude in typical benchmark problems.

Another two-step variant of IST, named *fast IST algorithm* (FISTA), was recently proposed and also shown to clearly outperform IST in terms of speed [4]. FISTA is a nonsmooth variant of Nesterov's optimal gradient-based algorithm for smooth convex problems [35], [36].

A strategy recently proposed to obtain faster variants of IST consists in relaxing the condition $\gamma > \gamma_{\min} \equiv \|\mathbf{A}\|_2^2/2$. In the SpaRSA (standing for *sparse reconstruction by separable approximation*) framework [44], [45], a different γ_t is used in each iteration (which may be smaller than $\gamma_{\min}$, meaning larger step sizes). It was shown experimentally that SpaRSA clearly outperforms standard IST. A convergence result for SpaRSA was also given in [45].

Finally, when the slowness is caused by the use of a small value of the regularization parameter, *continuation* schemes have been found quite effective in speeding up the IST algorithm. The key observation is that IST benefits significantly from *warm-starting*, i.e., from being initialized near a minimum of the objective function. This suggests that we can use the solution of (5), for a given value of τ, to initialize IST in solving

the same problem for a nearby value of τ. This *warm-starting* property underlies *continuation* schemes [24], [27], [45]. The idea is to use IST to solve (1) for a larger value of τ (which is usually fast), then decrease τ in steps toward its desired value, running IST with warm-start for each successive value of τ.

C. Proposed Approach

The approach proposed in this paper is based on the technique known as variable splitting, which goes back at least to Courant in the 40s [14], [43]. Since the objective function (5) to be minimized is the sum of two functions, the idea is to split the variable $\mathbf{x}$ into a pair of variables, say $\mathbf{x}$ and $\mathbf{v}$, each to serve as the argument of each of the two functions, and then minimize the sum of the two functions under the constraint that the two variables have to be equal, so that the problems are equivalent. Although variable splitting is also the rationale behind the recently proposed split-Bregman method [25], in this paper, we exploit a different type of splitting to attack problem (5). In the following, we will explain this difference in detail.

The constrained optimization problem resulting from variable splitting is then dealt with using an augmented Lagrangian (AL) scheme [37], which is known to be equivalent to the Bregman iterative methods recently proposed to handle imaging inverse problems (see [46] and references therein). We prefer the AL perspective, rather than the Bregman iterative view, as it is a standard and more elementary optimization tool (covered in most textbooks on optimization). In particular, we solve the constrained problem resulting from the variable splitting using an algorithm known as alternating direction method of multipliers (ADMM) [17].

The application of ADMM to our particular problem involves solving a linear system with the size of the unknown image (in the case of problem (2)) or with the size of its representation (in the case of problem (1)). Although this seems like an unsurmountable obstacle, we show that it is not the case. In many problems of the form (2), such as (circular) deconvolution, recovery of missing samples, or reconstruction from partial Fourier observations, this system can be solved very quickly in closed form (with $O(n)$ or $O(n \log n)$ cost). For problems of the form (1), we show how exploiting the fact that $\mathbf{W}$ is a tight Parseval frame, this system can still be solved efficiently (typically with $O(n \log n)$ cost).

We report results of a comprehensive set of experiments, on a set of benchmark problems, including image deconvolution, recovery of missing pixels, and reconstruction from partial Fourier transform, using both frame-based and TV-based regularization. In all the experiments, the resulting algorithm is consistently and considerably faster than the previous state of the art methods FISTA [4], TwIST [5], and SpaRSA [45].

Arguably, the speed of the proposed algorithm, which we term SALSA (*split augmented Lagrangian shrinkage algorithm*), comes from the fact that it uses (a regularized version of) the Hessian of the data fidelity term of (5), that is, $\mathbf{A}^H \mathbf{A}$, while the previously mentioned algorithms essentially only use gradient information.

D. Organization of the Paper

Section II describes the basic ingredients of SALSA: variable splitting, augmented Lagrangians, and ADMM. In Section III,

we show how these ingredients are combined to obtain the proposed SALSA. Section IV reports experimental results, and Section V ends the paper with a few remarks and pointers to future work.

II. Basic Ingredients

A. Variable Splitting

Consider an unconstrained optimization problem in which the objective function is the sum of two functions, one of which is written as the composition of two functions

$$\min_{\mathbf{u} \in \mathbb{R}^n} f_1(\mathbf{u}) + f_2(g(\mathbf{u})) \qquad (8)$$

where $g : \mathbb{R}^n \to \mathbb{R}^d$. Variable splitting is a very simple procedure that consists in creating a new variable, say $\mathbf{v}$, to serve as the argument of f_2, under the constraint that $g(\mathbf{u}) = \mathbf{v}$. This leads to the constrained problem

$$\min_{\mathbf{u} \in \mathbb{R}^n, \, \mathbf{v} \in \mathbb{R}^d} \quad f_1(\mathbf{u}) + f_2(\mathbf{v})$$
$$\text{subject to} \quad g(\mathbf{u}) = \mathbf{v} \qquad (9)$$

which is clearly equivalent to unconstrained problem (8): in the feasible set $\{(\mathbf{u}, \mathbf{v}) : g(\mathbf{u}) = \mathbf{v}\}$, the objective function in (9) coincides with that in (8). The rationale behind variable splitting methods is that it may be easier to solve the constrained problem (9) than it is to solve its unconstrained counterpart (8).

The splitting idea has been recently used in several image processing applications. A variable splitting method was used in [43] to obtain a fast algorithm for TV-based image restoration. Variable splitting was also used in [6] to handle problems involving compound regularizers; i.e., where instead of a single regularizer $\tau \phi(\mathbf{x})$ in (5), one has a linear combination of two (or more) regularizers $\tau_1 \phi_1(\mathbf{x}) + \tau_2 \phi_2(\mathbf{x})$. In [6] and [43], the constrained problem (9) is attacked by a quadratic penalty approach, i.e., by solving

$$\min_{\mathbf{u} \in \mathbb{R}^n, \mathbf{v} \in \mathbb{R}^d} \quad f_1(\mathbf{u}) + f_2(\mathbf{v}) + \frac{\alpha}{2} \|g(\mathbf{u}) - \mathbf{v}\|_2^2 \qquad (10)$$

by alternating minimization with respect to $\mathbf{u}$ and $\mathbf{v}$, while slowly taking α to very large values (a *continuation* process), to force the solution of (10) to approach that of (9), which in turn is equivalent to (8). The rationale behind these methods is that each step of this alternating minimization may be much easier than the original unconstrained problem (8). The drawback is that as α becomes very large, the intermediate minimization problems become increasingly ill-conditioned, thus, causing numerical problems (see [37, Ch. 17]).

A similar variable splitting approach underlies the recently proposed split-Bregman methods [25]; however, instead of using a quadratic penalty technique, those methods attack the constrained problem directly using a Bregman iterative algorithm [46]. It has been shown that, when g is a linear function, i.e., $g(\mathbf{u}) = \mathbf{G}\mathbf{u}$, the Bregman iterative algorithm is equivalent to the augmented Lagrangian method [46], which is briefly reviewed in the following subsection.

B. Augmented Lagrangian

Consider the constrained optimization problem

$$\min_{\mathbf{z} \in \mathbb{R}^n} \quad E(\mathbf{z})$$
$$\text{s.t.} \quad \mathbf{Hz} - \mathbf{b} = \mathbf{0} \qquad (11)$$

where $\mathbf{b} \in \mathbb{R}^p$ and $\mathbf{H} \in \mathbb{R}^{p \times n}$, i.e., there are p linear equality constraints. The *augmented Lagrangian* function for this problem is defined as

$$\mathcal{L}_A(\mathbf{z}, \boldsymbol{\lambda}, \mu) = E(\mathbf{z}) + \boldsymbol{\lambda}^T (\mathbf{b} - \mathbf{Hz}) + \frac{\mu}{2} \|\mathbf{Hz} - \mathbf{b}\|_2^2 \quad (12)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^p$ is a vector of Lagrange multipliers and $\mu \geq 0$ is called the penalty parameter [37].

The so-called *augmented Lagrangian method* (ALM) [37], also known as the *method of multipliers* (MM) [28], [38], consists in minimizing $\mathcal{L}_A(\mathbf{z}, \boldsymbol{\lambda}, \mu)$ with respect to $\mathbf{z}$, keeping $\boldsymbol{\lambda}$ fixed, then updating $\boldsymbol{\lambda}$, and repeating these two steps until some convergence criterion is satisfied. Formally, the ALM/MM works as follows:

Algorithm ALM/MM

---

1. Set $k = 0$, choose $\mu > 0$, and $\boldsymbol{\lambda}_0$.

2. **repeat**

3. $\mathbf{z}_{k+1} \in \arg\min_{\mathbf{z}} \mathcal{L}_A(\mathbf{z}, \boldsymbol{\lambda}_k, \mu)$

4. $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mu(\mathbf{Hz}_{k+1} - \mathbf{b})$

5. $k \leftarrow k + 1$

6. **until** stopping criterion is satisfied.

It is also possible (and even recommended) to update the value of μ in each iteration [3], [37, Chap. 9]. However, unlike in the quadratic penalty approach, the ALM/MM does not require μ to be taken to infinity to guarantee convergence to the solution of the constrained problem (11).

Notice that (after a straightforward complete-the-squares procedure) the terms added to $E(\mathbf{z})$ in the definition of the augmented Lagrangian $\mathcal{L}_A(\mathbf{z}, \boldsymbol{\lambda}_k, \mu)$ in (12) can be written as a single quadratic term (plus a constant independent of $\mathbf{z}$, thus, irrelevant for the ALM/MM), leading to the following alternative form of the algorithm (which makes clear its equivalence with the Bregman iterative method [46]):

Algorithm ALM/MM (Version II)

---

1. Set $k = 0$, choose $\mu > 0$ and $\mathbf{d}_0$.

2. **repeat**

3. $\mathbf{z}_{k+1} \in \arg\min_{\mathbf{z}} E(\mathbf{z}) + (\mu/2) \|\mathbf{Hz} - \mathbf{d}_k\|_2^2$

4. $\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{Hz}_{k+1} - \mathbf{b})$

5. $k \leftarrow k + 1$

6. **until** stopping criterion is satisfied.

It has been shown that, with adequate initializations, the ALM/MM generates the same sequence as a *proximal point algorithm* applied to the Lagrange dual of problem (11) [30]. Moreover, the sequence $\{\mathbf{d}_k\}$ converges to a solution of this dual problem and all cluster points of the sequence $\{\mathbf{z}_k\}$ are solutions of the (primal) problem (11) [30].

C. ALM/MM for Variable Splitting

We now review how the ALM/MM can be used to address problem (9), in the particular case where $g(\mathbf{u}) = \mathbf{Gu}$, i.e.,

$$\min_{\mathbf{u} \in \mathbb{R}^n, \, \mathbf{v} \in \mathbb{R}^d} \quad f_1(\mathbf{u}) + f_2(\mathbf{v})$$
$$\text{subject to} \quad \mathbf{Gu} = \mathbf{v} \qquad (13)$$

where $\mathbf{G} \in \mathbb{R}^{d \times n}$. Problem (13) can be written in the form (11) using the following definitions:

$$\mathbf{z} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}, \quad \mathbf{b} = \mathbf{0}, \quad \mathbf{H} = [\mathbf{G} \quad -\mathbf{I}] \qquad (14)$$

and

$$E(\mathbf{z}) = f_1(\mathbf{u}) + f_2(\mathbf{v}). \qquad (15)$$

With these definitions in place, Steps 3 and 4 of the ALM/MM (version II) can be written as follows:

$$(\mathbf{u}_{k+1}, \mathbf{v}_{k+1}) \in \arg\min_{\mathbf{u}, \mathbf{v}} f_1(\mathbf{u}) + f_2(\mathbf{v})$$
$$+ \frac{\mu}{2} \|\mathbf{Gu} - \mathbf{v} - \mathbf{d}_k\|_2^2 \qquad (16)$$
$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{Gu}_{k+1} - \mathbf{v}_{k+1}). \qquad (17)$$

The minimization problem (16) is not trivial since, in general, it involves nonseparable quadratic, and possibly nonsmooth, terms. A natural way to address (16) is to use a nonlinear block-Gauss-Seidel (NLBGS) technique, in which (16) is solved by alternatingly minimizing it with respect to $\mathbf{u}$ and $\mathbf{v}$, while keeping the other variable fixed. Of course this raises several questions: for a given $\mathbf{d}_k$, how much computational effort should be spent in approximating the solution of (16)? Does this NLBGS procedure converge? Experimental evidence in [25] suggests that an efficient algorithm is obtained by running just one NLBGS step. It turns out that the resulting algorithm is the so-called *alternating direction method of multipliers* (ADMM) [17], which works as follows:

Algorithm ADMM

---

1. Set $k = 0$, choose $\mu > 0$, $\mathbf{v}_0$, and $\mathbf{d}_0$.

2. **repeat**

3. $\mathbf{u}_{k+1} \in \arg\min_{\mathbf{u}} f_1(\mathbf{u}) + (\mu/2) \|\mathbf{Gu} - \mathbf{v}_k - \mathbf{d}_k\|_2^2$

4. $\mathbf{v}_{k+1} \in \arg\min_{\mathbf{v}} f_2(\mathbf{v}) + (\mu/2) \|\mathbf{Gu}_{k+1} - \mathbf{v} - \mathbf{d}_k\|_2^2$

5. $\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{Gu}_{k+1} - \mathbf{v}_{k+1})$

6. $k \leftarrow k + 1$

7. **until** stopping criterion is satisfied.

For later reference, we now recall the theorem by Eckstein and Bertsekas, in which convergence of (a generalized version of) ADMM is shown. This theorem applies to problems of the form (8) with $g(\mathbf{u}) = \mathbf{G}\mathbf{u}$, i.e.,

$$\min_{\mathbf{u}\in\mathbb{R}^n} f_1(\mathbf{u}) + f_2(\mathbf{G}\mathbf{u}) \qquad (18)$$

of which (13) is the constrained optimization reformulation.

Theorem 1 (Eckstein-Bertsekas, [17]): Consider problem (18), where f_1 and f_2 are closed, proper, convex functions, and $\mathbf{G} \in \mathbb{R}^{d\times n}$ has full column rank. Consider arbitrary $\mu > 0$ and $\mathbf{v}_0,\ \mathbf{d}_0 \in \mathbb{R}^d$. Let $\{\eta_k \geq 0,\ k = 0,1,\dots\}$ and $\{\nu_k \geq 0,\ k = 0,1,\dots\}$ be two sequences such that

$$\sum_{k=0}^{\infty} \eta_k < \infty \quad \text{and} \quad \sum_{k=0}^{\infty} \nu_k < \infty.$$

Consider three sequences $\{\mathbf{u}_k \in \mathbb{R}^n,\ k = 0,1,\dots\}$, $\{\mathbf{v}_k \in \mathbb{R}^d,\ k = 0,1,\dots\}$, and $\{\mathbf{d}_k \in \mathbb{R}^d,\ k = 0,1,\dots\}$ that satisfy

$$\eta_k \geq \left\| \mathbf{u}_{k+1} - \arg\min_{\mathbf{u}} \left\{ f_1(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{u} - \mathbf{v}_k - \mathbf{d}_k\|_2^2 \right\} \right\|$$

$$\nu_k \geq \left\| \mathbf{v}_{k+1} - \arg\min_{\mathbf{v}} \left\{ f_2(\mathbf{v}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{u}_{k+1} - \mathbf{v} - \mathbf{d}_k\|_2^2 \right\} \right\|$$

$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\mathbf{u}_{k+1} - \mathbf{v}_{k+1}).$$

Then, if (18) has a solution, the sequence $\{\mathbf{u}_k\}$ converges, $\mathbf{u}_k \to \mathbf{u}^*$, where $\mathbf{u}^*$ is a solution of (18). If (18) does not have a solution, then at least one of the sequences $\{\mathbf{v}_k\}$ or $\{\mathbf{d}_k\}$ diverges.

Notice that the ADMM algorithm defined previously generates sequences $\{\mathbf{u}_k\}$, $\{\mathbf{v}_k\}$, and $\{\mathbf{d}_k\}$ which satisfy the conditions in Theorem 1 in a strict sense (i.e., with $\eta_k = \mu_k = 0$). One of the important consequences of this theorem is that it shows that it is not necessary to exactly solve the minimizations in lines 3 and 4 of ADMM; as long as sequence of errors is absolutely summable, convergence is still guaranteed.

The proof of Theorem 1 is based on the equivalence between ADMM and the so-called Douglas-Rachford splitting method (DRSM) applied to the dual of problem (18). The DRSM was recently used for image recovery problems in [12]. For recent and comprehensive reviews of ALM/MM, ADMM, DRSM, and their relationship with Bregman and split-Bregman methods, see [26], [42].

III. PROPOSED METHOD

A. Constrained Optimization Formulation of Image Recovery

We now return to the unconstrained optimization formulation of regularized image recovery, as defined in (5). This problem can be written in the form (18), with

$$f_1(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \qquad (19)$$
$$f_2(\mathbf{x}) = \tau\phi(\mathbf{x}) \qquad (20)$$
$$\mathbf{G} = \mathbf{I}. \qquad (21)$$

The constrained optimization formulation is, thus

$$\min_{\mathbf{x},\mathbf{v}\in\mathbb{R}^n} \quad \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \tau\phi(\mathbf{v})$$
$$\text{subject to} \quad \mathbf{x} = \mathbf{v}. \qquad (22)$$

At this point, we are in a position to clearly explain the difference between this formulation and the splitting exploited in split-Bregman methods (SBM) for image recovery [25]. In those methods, the focus of attention is a nonseparable regularizer that can be written as $\phi(\mathbf{x}) = \varphi(\mathbf{D}\mathbf{x})$, as is the case of the TV norm. The variable splitting used in SBM addresses this nonseparability by defining the following constrained optimization formulation:

$$\min_{\mathbf{x},\mathbf{v}\in\mathbb{R}^n} \quad \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \tau\varphi(\mathbf{v})$$
$$\text{subject to} \quad \mathbf{D}\mathbf{x} = \mathbf{v}. \qquad (23)$$

In contrast, we assume that the Moreau proximal mapping associated to the regularizer ϕ, i.e., the function $\mathbf{\Psi}_{\tau\phi}(\cdot)$ defined in (6), can be computed efficiently. The goal of our splitting is not to address the difficulty raised by a nonseparable and non-quadratic regularizer, but to exploit second-order (Hessian) information of the function f_1, as will be shown in the following.

Algorithm and Its Convergence

Inserting the definitions given in (19)–(21) in the ADMM presented in the previous section yields the proposed SALSA (*split augmented Lagrangian shrinkage algorithm*).

Algorithm SALSA

---

1. Set $k = 0$, choose $\mu > 0$, $\mathbf{v}_0$, and $\mathbf{d}_0$.

2. **repeat**

3. $\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \mu\|\mathbf{x} - \mathbf{v}_k - \mathbf{d}_k\|_2^2$

4. $\mathbf{v}_{k+1} = \arg\min_{\mathbf{v}} \tau\phi(\mathbf{v}) + (\mu/2)\|\mathbf{x}_{k+1} - \mathbf{v} - \mathbf{d}_k\|_2^2$

5. $\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{x}_{k+1} - \mathbf{v}_{k+1})$

6. $k \leftarrow k + 1$

7. **until** stopping criterion is satisfied.

Since SALSA is an instance of ADMM with $\mathbf{G} = \mathbf{I}$, the full column rank condition in Theorem 1 is satisfied. If the minimizations in lines 3 and 4 are solved exactly, we can then invoke Theorem 1 to guarantee the convergence of SALSA.

In line 3 of SALSA, a strictly convex quadratic function has to be minimized, which leads to the following linear system:

$$\mathbf{x}_{k+1} = \left(\mathbf{A}^H\mathbf{A} + \mu\mathbf{I}\right)^{-1}\left(\mathbf{A}^H\mathbf{y} + \mu\mathbf{x}'_k\right) \qquad (24)$$

where $\mathbf{x}'_k = \mathbf{v}_k + \mathbf{d}_k$. As shown in the next subsection, this linear system can be solved exactly (naturally, up to numerical precision), i.e., noniteratively, for a comprehensive set of situations of interest. The matrix $\mathbf{A}^H\mathbf{A} + \mu\mathbf{I}$ can be seen as a regularized (by the addition of $\mu\mathbf{I}$) version of the Hessian of $f_1(\mathbf{x}) = (1/2)\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$, thus, SALSA does use second-order

information of this function. Notice also that (24) is formally similar to the *maximum a posteriori* (MAP) estimate of $\mathbf{x}$, from observations $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$ (where $\mathbf{n}$ is white Gaussian noise of variance $1/\mu$) under a Gaussian prior of mean $\mathbf{x}'_k$ and covariance $\mathbf{I}$.

The problem in line 4 is, by definition, the Moreau proximity mapping of ϕ applied to $\mathbf{v}'_k = \mathbf{x}_{k+1} - \mathbf{d}_k$, thus, its solution can be written as

$$\mathbf{v}_{k+1} = \mathbf{\Psi}_{\tau\phi/\mu}(\mathbf{v}'_k). \tag{25}$$

If this mapping can be computed exactly in closed form (for example, if $\phi(\mathbf{x}) = \|\mathbf{x}\|_1$, thus, $\mathbf{\Psi}$ is simply a soft threshold), then, by Theorem 1, SALSA is guaranteed to converge. If $\mathbf{\Psi}$ does not have a closed form solution and requires itself an iterative algorithm (e.g., if ϕ is the TV norm), then convergence of SALSA still holds if one can guarantee that the error sequence ν_k (see Theorem 1) is summable. In principle, this can be achieved if the iterative algorithm used to approximate $\mathbf{\Psi}$ is initialized with the result of the previous outer iteration, and a decreasing stopping threshold is used; this idea will be exploited in a future paper. In our experiments with TV regularization reported in this paper, we use a fixed number of iterations of Chambolle's algorithm to approximate $\mathbf{\Psi}$; this was empirically found not to compromise the convergence of SALSA.

B. Computing $\mathbf{x}_{k+1}$

As previously stated, we are interested in problems where it is not feasible to explicitly form matrix $\mathbf{A}$; this might suggest that it is not easy, or even feasible, to compute the inverse in (24). However, as shown next, in a number of problems of interest, this inverse can be computed very efficiently. Problems such as noncyclic deconvolution, for which the term $\left(\mathbf{A}^H\mathbf{A} + \mu\,\mathbf{I}\right)$ is not invertible will be addressed in a future paper.

1) Deconvolution With Analysis Prior: In this case, we have $\mathbf{A} = \mathbf{B}$ (see (1), (2), and (5)), where $\mathbf{B}$ is the matrix representation of a circular convolution. This is the simplest case, since the inverse $\left(\mathbf{B}^H\mathbf{B} + \mu\,\mathbf{I}\right)^{-1}$ can be computed in the Fourier domain. Although this is an elementary and well-known fact, we include the derivation for the sake of completeness. Assuming that the convolution is periodic (other boundary conditions can be addressed with minor changes), $\mathbf{B}$ is a block-circulant matrix with circulant blocks which can be factorized as

$$\mathbf{B} = \mathbf{U}^H\mathbf{D}\mathbf{U} \tag{26}$$

where $\mathbf{U}$ is the matrix that represents the 2-D discrete Fourier transform (DFT), $\mathbf{U}^H = \mathbf{U}^{-1}$ is its inverse ($\mathbf{U}$ is unitary, i.e., $\mathbf{U}\mathbf{U}^H = \mathbf{U}^H\mathbf{U} = \mathbf{I}$), and $\mathbf{D}$ is a diagonal matrix containing the DFT coefficients of the convolution operator represented by $\mathbf{B}$. Thus

$$\left(\mathbf{B}^H\mathbf{B} + \mu\,\mathbf{I}\right)^{-1} = \left(\mathbf{U}^H\mathbf{D}^*\mathbf{D}\mathbf{U} + \mu\,\mathbf{U}^H\mathbf{U}\right)^{-1} \tag{27}$$

$$= \mathbf{U}^H\left(|\mathbf{D}|^2 + \mu\,\mathbf{I}\right)^{-1}\mathbf{U} \tag{28}$$

where $(\cdot)^*$ denotes complex conjugate and $|\mathbf{D}|^2$ the squared absolute values of the entries of the diagonal matrix $\mathbf{D}$. Since $|\mathbf{D}|^2 + \mu\,\mathbf{I}$ is diagonal, its inversion has linear cost $O(n)$. The

TABLE I
DETAILS OF THE IMAGE DECONVOLUTION EXPERIMENTS

Experiment	blur kernel	σ^2
1	9×9 uniform	0.56^2
2A	Gaussian	2
2B	Gaussian	8
3A	$h_{ij} = 1/(1 + i^2 + j^2)$	2
3B	$h_{ij} = 1/(1 + i^2 + j^2)$	8

products by $\mathbf{U}$ and $\mathbf{U}^H$ can be carried out with $O(n \log n)$ cost using the FFT algorithm. The expression in (28) is a Wiener filter in the frequency domain.

2) Deconvolution With Frame-Based Synthesis Prior: In this case, we have a problem of the form (1), i.e., $\mathbf{A} = \mathbf{B}\mathbf{W}$, thus, the inversion that needs to be performed is $\left(\mathbf{W}^H\mathbf{B}^H\mathbf{B}\mathbf{W} + \mu\,\mathbf{I}\right)^{-1}$. Assuming that $\mathbf{B}$ represents a (periodic) convolution, this inversion may be sidestepped under the assumption that matrix $\mathbf{W}$ corresponds to a normalized tight frame (a Parseval frame), i.e., $\mathbf{W}\,\mathbf{W}^H = \mathbf{I}$. Applying the Sherman–Morrison–Woodbury (SMW) matrix inversion formula yields

$$\left(\mathbf{W}^H\mathbf{B}^H\mathbf{B}\mathbf{W} + \mu\,\mathbf{I}\right)^{-1}$$
$$= \frac{1}{\mu}\,(\mathbf{I} - \mathbf{W}^H\underbrace{\mathbf{B}^H\left(\mathbf{B}\mathbf{B}^H + \mu\,\mathbf{I}\right)^{-1}\mathbf{B}}_{\mathbf{F}}\mathbf{W}).$$

Let us focus on the term $\mathbf{F} \equiv \mathbf{B}^H\left(\mathbf{B}\mathbf{B}^H + \mu\,\mathbf{I}\right)^{-1}\mathbf{B}$; using the factorization (26), we have

$$\mathbf{F} = \mathbf{U}^H\mathbf{D}^*\left(|\mathbf{D}|^2 + \mu\,\mathbf{I}\right)^{-1}\mathbf{D}\mathbf{U}. \tag{29}$$

Since all the matrices in $\mathbf{D}^*\left(|\mathbf{D}|^2 + \mu\,\mathbf{I}\right)^{-1}\mathbf{D}$ are diagonal, this expression can be computed with $O(n)$ cost, while the products by $\mathbf{U}$ and $\mathbf{U}^H$ can be computed with $O(n \log n)$ cost using the FFT. Consequently, products by matrix $\mathbf{F}$ (defined in (29)) have $O(n \log n)$ cost.

Defining $\mathbf{r}_k = \left(\mathbf{A}^H\mathbf{y} + \mu\,\mathbf{x}'_k\right) = \left(\mathbf{W}^H\mathbf{B}^H\mathbf{y} + \mu\,\mathbf{x}'_k\right)$, allows writing (24) compactly as

$$\mathbf{x}_{k+1} = \frac{1}{\mu}\left(\mathbf{r}_k - \mathbf{W}^T\mathbf{F}\mathbf{W}\,\mathbf{r}_k\right). \tag{30}$$

Notice that multiplication by $\mathbf{F}$ corresponds to applying a filter in the Fourier domain. Finally, notice also that the term $\mathbf{B}^H\mathbf{W}^H\mathbf{y}$ can be precomputed, as it does not change during the algorithm.

The leading cost of each application of (30) will be either $O(n \log n)$ or the cost of the products by $\mathbf{W}^H$ and $\mathbf{W}$. For most tight frames used in image processing, these products correspond to direct and inverse transforms for which fast algorithms exist. For example, when $\mathbf{W}^H$ and $\mathbf{W}$ are the inverse and direct translation-invariant wavelet transforms, these products can be computed using the undecimated wavelet transform with $O(n \log n)$ total cost [32]. Curvelets also constitute a Parseval frame for which fast $O(n \log n)$ implementations of the forward and inverse transform exist [7]. Yet another example of a redundant Parseval frame is the complex wavelet transform, which has $O(n)$ computational cost [31], [41]. In conclusion,
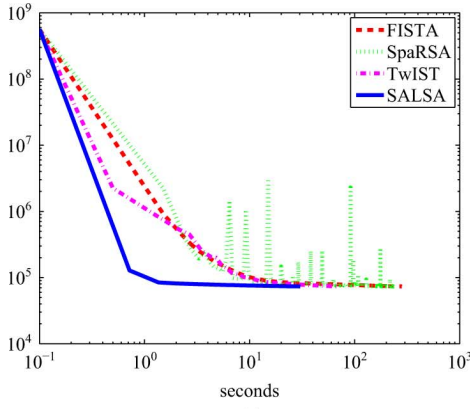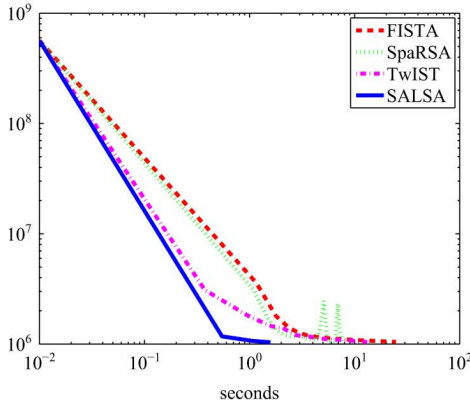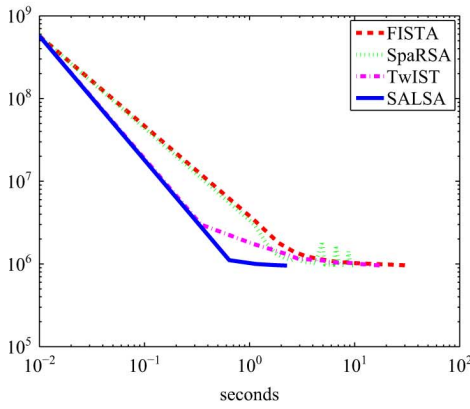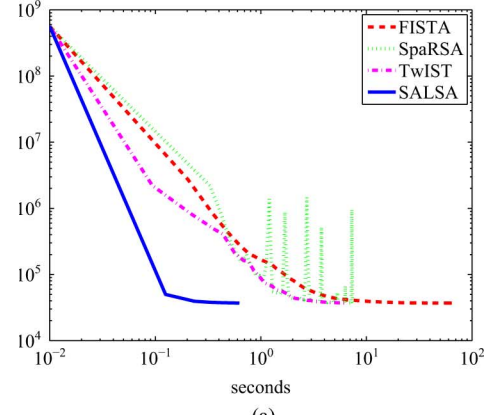
Fig. 1. Objective function evolution (redundant wavelets): (a) experiment 1A; (b) experiment 2B; (c) experiment 3A.

Fig. 2. Objective function evolution (orthogonal wavelets): (a) experiment 1A; (b) experiment 2B; (c) experiment 3A.
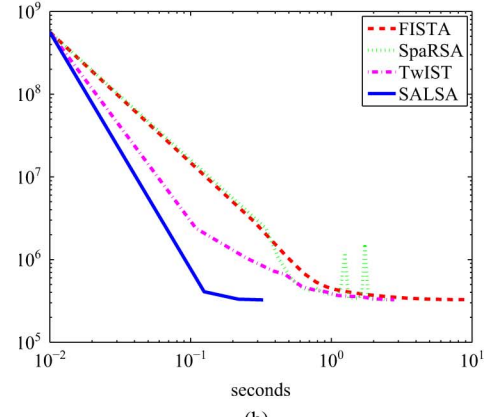
for a large class of choices of $\mathbf{W}$, each iteration of the SALSA algorithm has $O(n \log n)$ cost.

3) Missing Pixels (Image Inpainting): In the analysis prior case (TV-based), we have $\mathbf{A} = \mathbf{B}$, where the observation matrix $\mathbf{B}$ models the loss of some image pixels. Matrix $\mathbf{B}$ is, thus, an $m \times n$ binary matrix, with $m < n$, which can be obtained by taking a subset of rows of an identity matrix. Due to its particular structure, this matrix satisfies $\mathbf{B}\mathbf{B}^T = \mathbf{I}$. Using this fact together with the SMW formula leads to
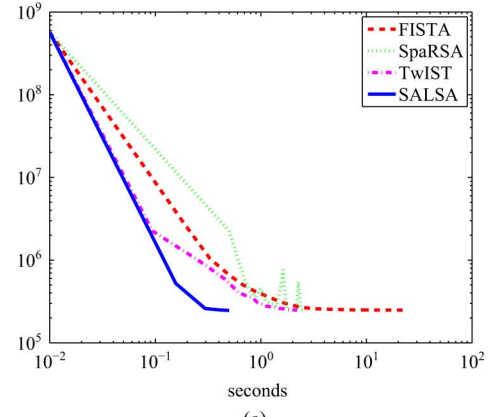
$$\left(\mathbf{B}^T\mathbf{B} + \mu\mathbf{I}\right)^{-1} = \frac{1}{\mu}\left(\mathbf{I} - \frac{1}{1+\mu}\,\mathbf{B}^T\mathbf{B}\right). \quad (31)$$

Since $\mathbf{B}^T\mathbf{B}$ is equal to an identity matrix with some zeros in the diagonal (corresponding to the positions of the missing observations), the matrix in (31) is diagonal with elements either equal to $1/(\mu + 1)$ or $1/\mu$. Consequently, (24) corresponds simply to multiplying $(\mathbf{B}^H\mathbf{y} + \mu\mathbf{x}'_k)$ by this diagonal matrix, which is an $O(n)$ operation.

In the synthesis prior case, we have $\mathbf{A} = \mathbf{B}\mathbf{W}$, where $\mathbf{B}$ is the binary subsampling matrix defined in the previous paragraph. Using the SMW formula yet again, and the fact that $\mathbf{B}\mathbf{B}^T = \mathbf{I}$, we have

$$\left(\mathbf{W}^H\mathbf{B}^H\mathbf{B}\mathbf{W} + \mu\,\mathbf{I}\right)^{-1} = \frac{1}{\mu}\,\mathbf{I} - \frac{\mu}{1+\mu}\,\mathbf{W}^H\mathbf{B}^T\mathbf{B}\mathbf{W}. \quad (32)$$

TABLE II
IMAGE DEBLURRING WITH REDUNDANT WAVELETS: COMPUTATIONAL COSTS AND ISNR VALUES. ALGORITHMS:
F = FISTA, T = TwIST, Sp = SpaRSA, S = SALSA.

Ex	Calls to $\mathbf{B}, \mathbf{B}^H$				Iterations				CPU time (seconds)				ISNR (dB)			
	F	T	Sp	S	F	T	Sp	S	F	T	Sp	S	F	T	Sp	S
1	1208	256	1064	94	402	123	531	46	240.40	56.09	195.50	22.65	5.46	7.66	7.59	7.73
2A	1067	210	1040	50	355	101	519	24	213.70	46.03	192.50	11.96	3.25	4.26	4.43	4.37
2B	134	62	43	8	44	28	20	3	26.80	13.26	8.21	1.57	3.56	3.49	3.34	3.58
3A	161	74	55	10	53	34	26	4	32.24	16.05	10.44	2.08	6.61	6.19	6.09	6.63
3B	134	53	41	12	44	24	19	5	26.82	11.13	7.849	2.56	4.45	4.22	4.49	4.51

TABLE III
IMAGE DEBLURRING WITH ORTHOGONAL WAVELETS: COMPUTATIONAL COSTS AND ISNR VALUES. ALGORITHMS:
F = FISTA, T = TwIST, Sp = SpaRSA, S = SALSA.

Ex	Calls to $\mathbf{B}, \mathbf{B}^H$				Iterations				CPU time (seconds)				ISNR (dB)			
	F	T	Sp	S	F	T	Sp	S	F	T	Sp	S	F	T	Sp	S
1	1367	151	199	14	455	70	98	6	71.10	8.05	10.18	0.74	5.98	6.65	6.53	6.71
2A	1268	98	98	10	422	43	48	4	64.60	5.15	5.02	0.49	0.62	4.08	4.09	4.10
2B	176	66	57	8	58	29	27	3	8.88	3.41	2.96	0.37	1.65	2.37	2.97	2.87
3A	470	47	43	10	156	19	20	4	23.89	2.42	2.28	0.48	4.42	5.13	5.17	5.33
3B	87	35	29	8	29	14	13	3	4.41	1.74	1.57	0.37	3.23	3.68	3.82	3.79

As noted in the previous paragraph, $\mathbf{B}^T\mathbf{B}$ is equal to an identity matrix with zeros in the diagonal (corresponding to the positions of the missing observations), i.e., it is a binary mask. Thus, the multiplication by $\mathbf{W}^H\mathbf{B}^H\mathbf{B}\mathbf{W}$ corresponds to synthesizing the image, multiplying it by this mask, and computing the representation coefficients of the result. In conclusion, the cost of (24) is again that of the products by $\mathbf{W}$ and $\mathbf{W}^H$, usually $O(n \log n)$.

4) Partial Fourier Observations: MRI Reconstruction: The final case considered is that of partial Fourier observations, which is used to model magnetic resonance image (MRI) acquisition [33], and has been the focus of much recent interest due to its connection to compressed sensing [8], [9], [16]. In the TV-regularized case, the observation matrix has the form $\mathbf{A} = \mathbf{B}\mathbf{U}$, where $\mathbf{B}$ is an $m \times n$ binary matrix, with $m < n$, similar to the one in the missing pixels case (it is formed by a subset of rows of an identity matrix), and $\mathbf{U}$ is the DFT matrix. This case is similar to (32), with $\mathbf{U}$ and $\mathbf{U}^H$ instead of $\mathbf{W}$ and $\mathbf{W}^H$, respectively. The cost of (24) is again that of the products by $\mathbf{U}$ and $\mathbf{U}^H$, i.e., $O(n \log n)$ if we use the FFT.

In the synthesis case, the observation matrix has the form $\mathbf{A} = \mathbf{B}\mathbf{U}\mathbf{W}$. Clearly, the case is again similar to (32), but with $\mathbf{U}\mathbf{W}$ and $\mathbf{W}^H\mathbf{U}^H$ instead of $\mathbf{W}$ and $\mathbf{W}^H$, respectively. Again, the cost of (24) is $O(n \log n)$, if the FFT is used to compute the products by $\mathbf{U}$ and $\mathbf{U}^H$ and fast frame transforms are used for the products by $\mathbf{W}$ and $\mathbf{W}^H$.

IV. EXPERIMENTS

In this section, we report results of experiments aimed at comparing the speed of SALSA with that of the current state of the art methods (all of which are freely available online): TwIST[2] [5], SpaRSA[3] [45], and FISTA[4] [4]. We consider three standard and often studied imaging inverse problems: image deconvolution (using both wavelet and TV-based regularization); image restoration from missing samples (inpainting); image reconstruction from partial Fourier observations, which (as mentioned previously) has been the focus of much recent interest due to its connection with compressed sensing and the fact that it models MRI acquisition [33]. All the experiments were performed using MATLAB, on a computer equipped with an Intel Pentium-IV 3.0 GHz processor, with 1.5 GB of RAM, and running Windows XP. To compare the speed of the algorithms, in a way that is as independent as possible from the different stopping criteria, we first run FISTA and then SALSA and the other algorithms until they reach the same value of the objective function. The value of μ for fastest convergence was found to differ (though not very much) in each case, but a good rule of thumb, adopted in all the experiments, is $\mu = \tau/10$. The number of calls to the operators $\mathbf{B}, \mathbf{B}^H$, the number of iterations, computation times, and improvement in SNR (ISNR) tabulated for each experiment are the average values over 10 instances of each experiment. The average ISNR was computed as $10\log_{10}(\sum_k \|\mathbf{x} - \mathbf{y}_k\|^2 / \sum_k \|\mathbf{x} - \hat{\mathbf{x}}_k\|^2)$, where $\mathbf{x}$ is the original image, $\mathbf{y}_k$ is the observed image at the kth iteration, and $\hat{\mathbf{x}}_k$ is the corresponding estimated image. The plots of the objective functions, for each experiment, were with a logarithmic x-axis which was started at 0.01 s, for the purpose of display.

A. Image Deblurring With Wavelets

We consider five benchmark deblurring problems [22], summarized in Table I, all on the well-known Cameraman image, with size 256×256 pixels. The regularizer is $\phi(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$, thus, $\boldsymbol{\Psi}_{\tau\phi}$ is an element-wise soft threshold. The blur operator $\mathbf{B}$ is applied via the FFT. The regularization parameter τ is hand tuned in each case for best improvement in SNR, so that the comparison is carried out in the regime that is relevant in practice.

In the first set of experiments, $\mathbf{W}$ is a redundant Haar wavelet frame with four levels. The average number of calls to the operators $\mathbf{B}, \mathbf{B}^H$, the number of iterations, the computation times, and the ISNR achieved by each of the algorithms are presented in Table II. In the second set of experiments, $\mathbf{W}$ is an orthogonal Haar wavelet basis; the results are reported in Table III. To visually illustrate the relative speed of the algorithms, Figs. 1 and 2 plot the evolution of the objective function [see (1)], versus time,

[2] Available at http://www.lx.it.pt/~bioucas/code/TwIST_v1.zip

[3] Available at http://www.lx.it.pt/~mtf/SpaRSA/

[4] Available at http://iew3.technion.ac.il/~becka/papers/wavelet_FISTA.zip

TABLE IV
TV-BASED IMAGE DEBLURRING: COMPUTATIONAL COSTS AND ISNR VALUES. ALGORITHMS: $F = FISTA, T = TwIST, Sp = SpaRSA, S = SALSA$

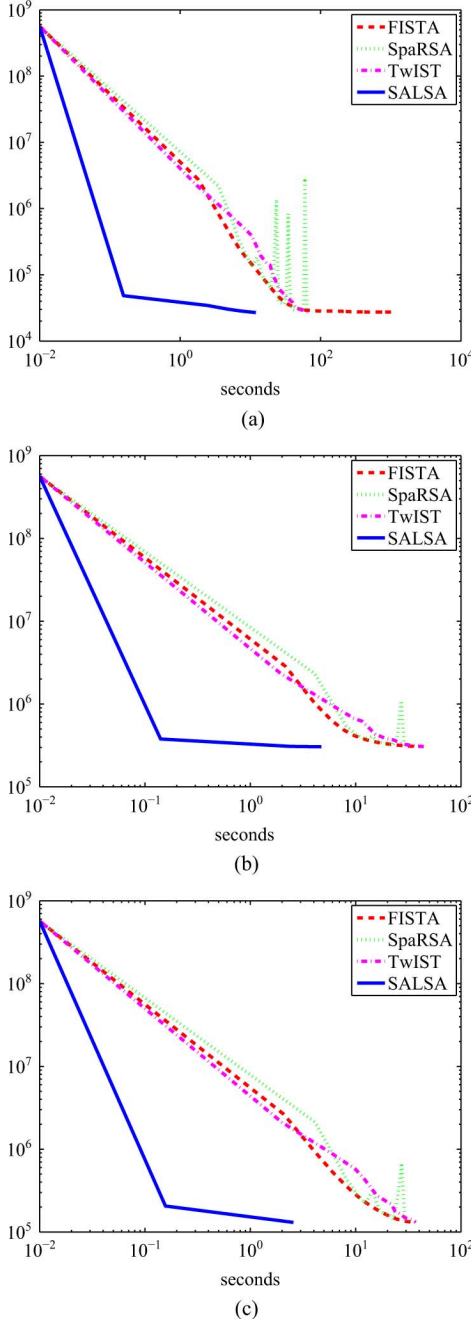| Ex	Calls to $\mathbf{B}, \mathbf{B}^H$				Iterations				CPU time (seconds)				ISNR (dB)			
	F	T	Sp	S	F	T	Sp	S	F	T	Sp	S	F	T	Sp	S
1	869	69	73	13	289	29	35	6	532.70	61.41	64.95	10.96	7.42	5.16	5.73	8.34
2A	104	51	44	6	34	21	21	2	31.90	26.56	21.73	1.35	3.60	3.68	3.61	4.08
2B	73	41	29	8	24	16	13	3	23.09	23.12	15.19	2.52	2.77	2.66	2.59	3.21
3A	123	32	27	6	41	12	12	2	40.55	17.85	14.01	1.37	5.02	4.57	4.55	6.00
3B	222	18	13	6	74	6	5	2	80.34	10.85	6.83	1.36	1.69	1.57	1.70	3.93

Fig. 3. Image deblurring with TV regularization—Objective function evolution: (a) 9×9 uniform blur, $\sigma = 0.56$; (b) Gaussian blur, $\sigma^2 = 8$; (c) $h_{ij} = 1/(1 + i^2 + j^2)$ blur, $\sigma^2 = 2$.

in experiments 1, 2B, and 3A, for redundant and orthogonal wavelets, respectively.

B. Image Deblurring With Total Variation

The same five image deconvolution problems listed in Table I were also addressed using total variation (TV) regularization (more specifically, the isotropic discrete total variation, as defined in [10]). The corresponding Moreau proximal mapping is computed using 5 iterations of Chambolle's algorithm [10].

The average number of calls to the operators $\mathbf{B}, \mathbf{B}^H$, the number of iterations, computation times, and the ISNR values obtained by SALSA, TwIST, SpaRSA, and FISTA are listed in Table IV. The evolutions of the objective functions (for experiments 1, 2B, and 3A) are plotted in Fig. 3.

We can conclude from Tables II–IV that, in image deconvolution problems, both with wavelet-based and TV-based regularization, SALSA is clearly faster by at least an order of magnitude of the computation time, under our experimental conditions. If an approximate solution is needed, in some problems such as experiment 3A for deconvolution with orthogonal wavelets, it may be possible to get a solution using TwIST or FISTA, quicker than SALSA.

C. MRI Image Reconstruction

We consider the problem of reconstructing the 128×128 Shepp-Logan phantom [shown in Fig. 4(a)] from a limited number of radial lines [22, in our experiments, as shown in Fig. 4(b)] of its 2-D discrete Fourier transform. The projections are also corrupted with circular complex Gaussian noise, with variance $\sigma^2 = 0.5 \times 10^{-3}$. We use TV regularization (as described in Section IV-B), with the corresponding Moreau proximal mapping implemented by 40 iterations of Chambolle's algorithm [10].

Table V shows the CPU times, numbers of products by $\mathbf{B}$ or $\mathbf{B}^H$, numbers of iterations, and MSE values, while Fig. 5 plots the evolution of the objective function over time. Fig. 4(c) shows the estimate obtained using SALSA (the others are, naturally, visually indistinguishable). As in the case of some of the image deconvolution problems, if an approximate solution is needed, it may be possible to get a solution using TwIST or FISTA, quicker than SALSA. SALSA is faster by almost an order of magnitude of the computation time, under our experimental conditions.

D. Image Inpainting

Finally, we consider an image inpainting problem, as explained in Section III-C. The original image is again the 256×256 Cameraman, and the observation consists in losing 40% of its pixels, as shown in Fig. 6. The observations are also corrupted with Gaussian noise (with an SNR of 40 dB).
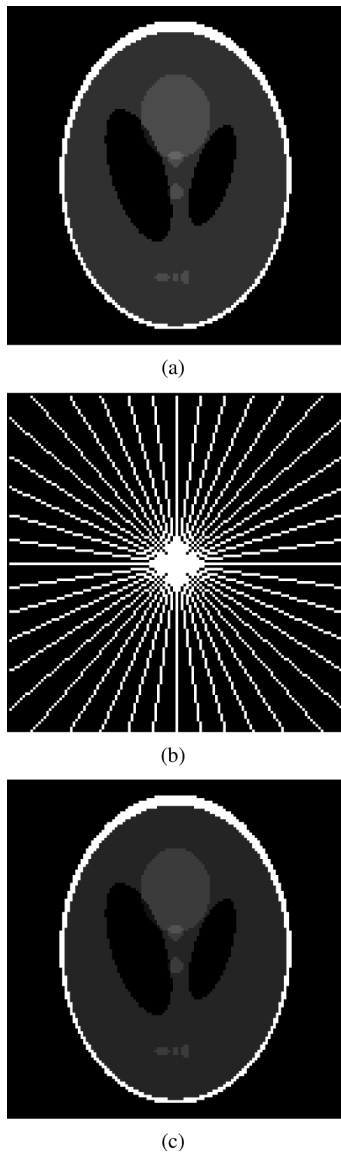
(a)

(b)

(c)

Fig. 4. MRI reconstruction: (a)128 × 128 Shepp Logan phantom; (b) mask with 22 radial lines; (c) image estimated using SALSA.
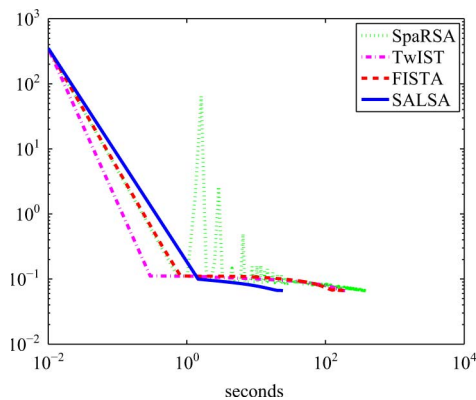

Fig. 5. MRI reconstruction: evolution of the objective function over time.

The regularizer is again TV implemented by 20 iterations of Chambolle's algorithm.

TABLE V
MRI RECONSTRUCTION: COMPARISON OF THE VARIOUS ALGORITHMS

Alg.	Calls to $\mathbf{B}, \mathbf{B}^H$	Iters.	CPU time (sec.)	MSE
FISTA	1520	506	147.1	3.14e-6
TwIST	1073	477	171.3	9.65e-6
SpaRSA	2157	1077	312.4	1.19e-5
SALSA	101	17	23.38	2.45e-6

(a)

(b)

(c)

Fig. 6. Image inpainting with TV regularization: (a) original cameraman image; (b) image with 40% pixels missing; (c) estimated using SALSA.

The image estimate obtained by SALSA is shown in Fig. 6, with the original also shown for comparison. The estimates obtained using TwIST and FISTA were visually very similar. For this experiment, SpARSA was unable to reach the value of the objective function reached by the others, even after 5000 iterations and, thus, will not be compared here. Table VI compares the performance of SALSA with that of TwIST and FISTA and Fig. 7 shows the evolution of the objective function for each of the algorithms. For our experimental conditions, SALSA is considerably faster than the alternative algorithms.
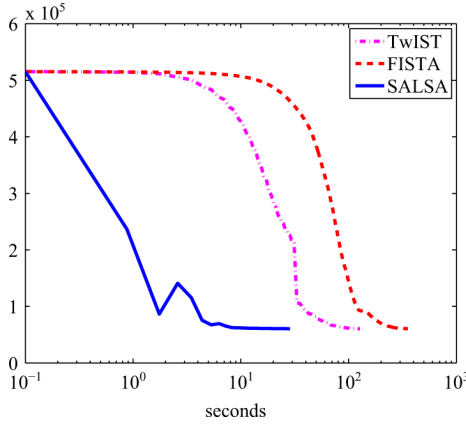
Fig. 7. Image inpainting: evolution of the objective function over time.

TABLE VI
IMAGE INPAINTING: COMPARISON OF THE VARIOUS ALGORITHMS

Alg.	Calls to $\mathbf{B}, \mathbf{B}^H$	Iter.	CPU time (sec.)	MSE MSE	ISNR (dB)
FISTA	1022	340	263.8	92.01	18.96
TwIST	271	124	112.7	100.92	18.54
SALSA	84	28	20.88	77.61	19.68

V. CONCLUSIONS

We have presented a new algorithm for solving the unconstrained optimization formulation of regularized image reconstruction/restoration. The approach, which can be used with different types of regularization (wavelet-based, total variation), is based on a variable splitting technique which yields an equivalent constrained problem. This constrained problem is then addressed using an augmented Lagrangian method, more specifically, the alternating direction method of multipliers (ADMM). The algorithm uses a regularized version of the Hessian of the ℓ_2 data-fidelity term, which can be computed efficiently for several classes of problems. Experiments on a set of standard image recovery problems (deconvolution, MRI reconstruction, inpainting) have shown that the proposed algorithm (termed SALSA, for *split augmented Lagrangian shrinkage algorithm*) is faster than previous state-of-the-art methods. Current and future work involves using a similar approach to solve constrained formulations of the forms (3) and (4), addressing the case where the term $\left(\mathbf{A}^H \mathbf{A} + \mu\, \mathbf{I}\right)$ in (24) is not invertible, and using as the MPM for TV regularization, an iterative algorithm initialized with the result of the previous outer iteration, and a decreasing stopping threshold.

REFERENCES

[1] H. Andrews and B. Hunt, *Digital Image Restoration*. Upper Saddle River, NJ: Prentice-Hall, 1977.

[2] O. Axelsson, *Iterative Solution Methods*. Cambridge, U.K.: Cambridge Univ. Press, 1996.

[3] M. Bazaraa, H. Sherali, and C. Shetty, *Nonlinear Programming: Theory and Algorithms*. Hoboken, NJ: Wiley, 1993.

[4] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, pp. 183–202, 2009.

[5] J. Bioucas-Dias and M. Figueiredo, "A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[6] J. Bioucas-Dias and M. Figueiredo, "An iterative algorithm for linear inverse problems with compound regularizers," in *Proc. IEEE Int. Conf. Image Processing*, San Diego, CA, 2008, pp. 685–688.

[7] E. Candès, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," *Multiscale Model. Sim.*, vol. 5, pp. 861–899, 2005.

[8] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate information," *Commun. Pure Appl. Math.*, vol. 59, pp. 1207–1233, 2005.

[9] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[10] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, no. 1–2, pp. 89–97, 2004.

[11] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Recent developments in total variation image restoration," in *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds. New York: Springer-Verlag, 2005.

[12] P. Combettes and J.-C. Pesquet, "A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 564–574, Dec. 2007.

[13] P. Combettes and V. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM J. Multiscale Model. Sim.*, vol. 4, pp. 1168–1200, 2005.

[14] R. Courant, "Variational methods for the solution of problems with equilibrium and vibration," *Bull. Amer. Math. Soc.*, vol. 49, pp. 1–23, 1943.

[15] I. Daubechies, M. D. Friese, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2004.

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

[17] J. Eckstein and D. Bertsekas, "On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 5, pp. 293–318, 1992.

[18] M. Elad, B. Matalon, and M. Zibulevsky, "Image denoising with shrinkage and redundant representations," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, New York, 2006, pp. 1924–1931.

[19] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inv. Probl.*, vol. 23, pp. 947–968, 2007.

[20] M. Figueiredo, J. Bioucas-Dias, and M. Afonso, "Fast frame-based image deconvolution using variable splitting and constrained optimization," in *Proc. IEEE Workshop on Statistical Signal Processing*, Cardiff, 2009, pp. 109–112.

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

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

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

[24] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–598, Dec. 2007.

[25] T. Goldstein and S. Osher, "The split Bregman algorithm for L_1 regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[26] E. Esser, Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split-Bregman Computat. Appl. Math., Univ. California, Los Angeles, 2009, Tech. Rep. 09-31.

[27] T. Hale, W. Yin, and Y. Zhang, *A Fixed-Point Continuation Method for ℓ_1-Regularized Minimization With Applications to Compressed Sensing. TR07-07.* Houston, TX: Dept. Computat. Appl. Math., Rice Univ., 2007.

[28] M. Hestenes, "Multiplier and gradient methods," *J. Opt. Theory Appl.*, vol. 4, pp. 303–320, 1969.

[29] D. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statist.*, vol. 58, pp. 30–37, 2004.

[30] A. Iusem, "Augmented Lagrangian methods and proximal point methods for convex optimization," *Investigación Operativa*, vol. 8, pp. 11–49, 1999.

[31] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *J. Appl. Computat. Harm. Anal.*, vol. 10, pp. 234–253, 2001.

[32] M. Lang, H. Guo, J. Odegard, C. Burrus, and R. Wells, "Noise reduction using an undecimated discrete wavelet transform," *IEEE Signal Process. Lett.*, vol. 3, no. 1, pp. 10–12, Jan. 1996.

[33] M. Lustig, D. Donoho, and J. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic Resonance Med.*, vol. 58, pp. 1182–1195, 2007.

[34] S. Mallat, *A Wavelet Tour of Signal Processing*. San Diego, CA: Academic, 1998.

[35] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Soviet Math. Doklady*, vol. 269, pp. 543–547, 1983.

[36] Y. Nesterov, *Introductory Lectures on Optimization*. Norwell, MA: Kluwer, 2004.

[37] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer-Verlag, 2006.

[38] M. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization*, R. Fletcher, Ed. New York: Academic, 1969, pp. 283–298.

[39] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton Univ. Press, 1970.

[40] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[41] I. Selesnick, "Hilbert transform pairs of wavelet bases," *IEEE Signal Process. Lett.*, vol. 8, no. 6, pp. 170–173, Jun. 2001.

[42] S. Setzer, "Split Bregman algorithm, Douglas-Rachford splitting, and frame shrinkage," in *Proc. 2nd Int. Conf. Scale Space Methods and Variational Methods in Computer Vision*, 2009, pp. 464–476.

[43] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imag. Sci.*, vol. 1, pp. 248–272, 2008.

[44] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Las Vegas, NV, 2008, pp. 3373–3376.

[45] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[46] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for ℓ_1 minimization with applications to compressed sensing," *SIAM J. Imag. Sci.*, vol. 1, pp. 143–168, 2008.

Manya Afonso received the B.E. degree in electronics and telecommunication engineering from Goa University, Goa, India, in 2003, the M.Tech. degree in communication engineering from the Indian Institute of Technology, Delhi in 2005, and is currently pursuing the Ph.D. degree from the Instituto Superior Técnico (IST), Technical University of Lisbon (TULisbon), Portugal.

Since 2007, he has been a Researcher at Instituto de Telecomunicações, Lisboa, Portugal, a private not-for-profit research institution. He is a Marie Curie Actions research fellow under the SIGNAL programme. His current research interests include image processing and analysis, inverse problems, statistical inference, and optimization. He had previously worked in industry as a software developer in the field of network management and monitoring.

José M. Bioucas-Dias (S'87–M'95) received the E.E., M.Sc., Ph.D., and "Agregado" degrees, all in electrical and computer engineering, from Instituto Superior Técnico (IST), the Engineering School of the Technical University of Lisbon (TULIsbon), Portugal, in 1985, 1991, 1995, and 2007, respectively.

Since 1995, he has been with the Department of Electrical and Computer Engineering, IST. He is also a Senior Researcher with the Communication Theory and Pattern Recognition Group of the Institute of Telecommunications, a private not-for-profit research institution. His research interests include signal and image processing, pattern recognition, optimization, and remote sensing.

Dr. Bioucas-Dias is involved in several national and international research projects and networks, including the Marie Curie Actions "Hyperspectral Imaging Network (HYPER-I-NET)" and the "European Doctoral Program in Signal Processing (SIGNAL)." He is an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS and SYSTEMS, and a guest editor of a special issue of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He has been a member of program/technical committees of several international conferences, including CVPR, ICPR, ICIAR, IGARSS, ICIP, SPIE, EMMCVPR, ISVC, and WHISPERS.

Mário A. T. Figueiredo (S'87–M'95–SM'00–F'10) received EE, MSc, Ph.D., and "Agregado" degrees in electrical and computer engineering, all from Instituto Superior Técnico (IST), the engineering school of the Technical University of Lisbon (TULisbon), Portugal, in 1985, 1990, 1994, and 2004, respectively.

Since 1994, he has been with the faculty of the Department of Electrical and Computer Engineering, IST. He is also area coordinator at Instituto de Telecomunicações, a private not-for-profit research institution. He has spent sabbatical leaves at the Department of Computer Science and Engineering, Michigan State University, and the Department of Electrical and Computer Engineering, University of Wisconsin-Madison, in 1998 and 2005, respectively. His research interests include image processing and analysis, pattern recognition, statistical learning, and optimization.

Dr. Figueiredo is a Fellow of the IAPR (International Association for Pattern Recognition) and a member of the Image, Video, and Multidimensional Signal Processing Technical Committee of the IEEE. He received the 1995 Portuguese IBM Scientific Prize and the 2008 UTL/Santander-Totta Scientific Prize. He is/was associate editor of the following journals: IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON PATTERN ANALYSIS and MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MOBILE COMPUTING, *Pattern Recognition Letters*, and *Signal Processing*. He was a co-chair of the 2001 and 2003 Workshops on Energy Minimization Methods in Computer Vision and Pattern Recognition, and program/technical committee member of many international conferences.