An Augmented Lagrangian Approach to the Constrained Optimization Formulation of Imaging Inverse Problems

Manya V. Afonso, José M. Bioucas-Dias, and Mário A. T. Figueiredo

***Abstract*—We propose a new fast algorithm for solving one of the standard approaches to ill-posed linear inverse problems (IPLIP), where a (possibly nonsmooth) regularizer is minimized under the constraint that the solution explains the observations sufficiently well. Although the regularizer and constraint are usually convex, several particular features of these problems (huge dimensionality, nonsmoothness) preclude the use of off-the-shelf optimization tools and have stimulated a considerable amount of research. In this paper, we propose a new efficient algorithm to handle one class of constrained problems (often known as basis pursuit denoising) tailored to image recovery applications. The proposed algorithm, which belongs to the family of augmented Lagrangian methods, can be used to deal with a variety of imaging IPLIP, including deconvolution and reconstruction from compressive observations (such as MRI), using either total-variation or wavelet-based (or, more generally, frame-based) regularization. The proposed algorithm is an instance of the so-called alternating direction method of multipliers, for which convergence sufficient conditions are known; we show that these conditions are satisfied by the proposed algorithm. Experiments on a set of image restoration and reconstruction benchmark problems show that the proposed algorithm is a strong contender for the state-of-the-art.**

***Index Terms*—Convex optimization, frames, image reconstruction, image restoration, inpainting, total-variation.**

I. Introduction

A. Problem Formulation

IMAGE restoration/reconstruction is one of the earliest and most classical linear inverse problems in imaging, dating back to the 1960s [3]. 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 adopt the vector notation for images, where the pixels on an $M \times N$ image are stacked into $\mathbf{x}$ an (NM)-vector in, e.g., lexicographic order. In the sequel, we denote by $n = MN$ 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 be different).

In the particular case of image deconvolution, $\mathbf{B}$ is the matrix representation of a convolution operator. This type of 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 [7].

In more general image reconstruction problems, $\mathbf{B}$ represents some linear direct operator, such as tomographic projections (Radon transform), a partially observed (e.g., Fourier) transform, or the loss of part of the image pixels.

The problem of estimating $\mathbf{x}$ from $\mathbf{y}$ is called a linear inverse problem (LIP); for most scenarios of practical interest, this is an ill-posed LIP (IPLIP), i.e., matrix $\mathbf{B}$ is singular and/or very ill-conditioned. Consequently, this IPLIP requires some sort of regularization (or prior information, in Bayesian inference terms). One way to regularize the problem of estimating $\mathbf{x}$, given $\mathbf{y}$, consists in a constrained optimization problem of the form

$$\min_{\mathbf{x}} \phi(\mathbf{x}) \qquad \text{subject to} \qquad \|\mathbf{B}\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon \tag{1}$$

where $\phi : \mathbb{R}^n \to \bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ is the regularizer or regularization function, and $\varepsilon \geq 0$ a parameter which depends upon the noise variance. In the case where $\phi(\mathbf{x}) = \|\mathbf{x}\|_1$, the previous problem is usually known as basis pursuit denoising (BPD) [16]. The so-called basis pursuit (BP) problem is the particular case of (1) for $\varepsilon = 0$.

In recent years, an explosion of interest in problems of the form (1) was sparked by the emergence of compressive sensing (CS) [13], [23]. The theory of CS provides conditions (on matrix $\mathbf{B}$ and the degree of sparseness of the original $\mathbf{x}$) under which a solution of (1), for $\phi(\mathbf{x}) = \|\mathbf{x}\|_1$, is an optimal (in some sense) approximation to the "true" $\mathbf{x}$.

In most signal/image recovery and CS problems, nonsmooth regularizers such as the total variation (TV) [13], [48] and $\ell_1(\phi(\mathbf{x}) = \|\mathbf{x}\|_1)$ norms are popular and powerful choices.

B. Analysis and Synthesis Formulations

In a frame-based representation, the unknown image $\mathbf{x}$ can 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

Manuscript received December 17, 2009; revised June 05, 2010; accepted August 27, 2010. Date of publication September 13, 2010; date of current version February 18, 2011. The work of M. Afonso was supported by a EU Marie-Curie Fellowship under Contract MEST-CT-2005-021175. This work was supported in part by Fundação para a Ciência e Tecnologia (FCT), Portuguese Ministry of Science and Higher Education, under project PTDC/EEA-TEL/104515/2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Oscar C. Au.

The authors are with the Instituto de Telecomunicações and Department of Electrical and Computer Engineering, Instituto Superior Técnico, 1049-001 Lisboa, Portugal (e-mail: mafonso@lx.it.pt; bioucas@lx.it.pt; mtf@lx.it.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2076294

the $n \times d$ matrix $\mathbf{W}$ are the elements of a wavelet[1] frame (an orthogonal basis or an overcomplete dictionary). The coefficients of this representation are then estimated from the noisy image, under one of the well-known sparsity inducing regularizers, such as the ℓ_1 norm (see [8], [21], [25], [27], [30], and further references therein). Formally, this leads to the following constrained optimization problem:

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

This formulation is referred to as the synthesis approach[26], [50], since it is based on a synthesis equation: $\mathbf{x}$ is synthesized from its representation coefficients ($\mathbf{x} = \mathbf{W}\boldsymbol{\beta}$) which are the object of the estimation criterion. Naturally, the estimate of $\mathbf{x}$ is $\widehat{\mathbf{x}} = \mathbf{W}\widehat{\boldsymbol{\beta}}$. Of course, (2) has the form (1) with $\mathbf{BW}$ replacing $\mathbf{B}$.

An alternative formulation applies a regularizer directly to the unknown image, leading to criteria of the form (1), usually called analysis approaches, since they are based on a regularizer that analyzes the image itself, rather than the coefficients of a representation thereof. Arguably, the best known and most often used regularizer in analysis approaches to image restoration is the TV norm [15], [48].

Wavelet-based analysis approaches are also possible and have the form

$$\min_{\mathbf{x}} \phi(\mathbf{P}\mathbf{x}) \qquad \text{subject to} \qquad \|\mathbf{B}\,\mathbf{x} - \mathbf{y}\|_2^2 \leq \varepsilon \qquad (3)$$

where $\mathbf{P}$ is a linear operator (a matrix) corresponding to a wavelet transform [26]. In this paper, we always assume that $\mathbf{P}$ is the analysis operator associated with 1-tight (Parseval) frame, thus, $\mathbf{P}^H\mathbf{P} = \mathbf{I}$ [42].

If the regularizers are convex, problems (1)–(3) are convex, but the very high dimensions (at least $\geq 10^4$, often $\gg 10^5$) of $\mathbf{x}$ and possibly $\mathbf{y}$ precludes the direct application of off-the-shelf optimization algorithms. This difficulty is further amplified by the following fact: for any problem of nontrivial dimension, matrices $\mathbf{B}$, $\mathbf{W}$, or $\mathbf{P}$ 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 these matrices (or their conjugate transposes, denote by $(\cdot)^H$) can be computed 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 transform algorithm [42]. Similarly, if $\mathbf{B}$ represents a convolution, products by $\mathbf{B}$ or $\mathbf{B}^H$ can be performed with the help of the fast Fourier transform (FFT). These facts have stimulated the development of special purpose methods, in which the only operations involving matrices are matrix-vector products.

C. Previous Algorithms: Unconstrained Formulations

Most state-of-the-art methods for dealing with linear inverse problems, under convex, nonsmooth regularizers (namely, TV and ℓ_1), consider, rather than (1), the unconstrained problem

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{B}\mathbf{x} - \mathbf{y}\|_2^2 + \tau\phi(\mathbf{x}) \qquad (4)$$

where $\tau \in \mathbb{R}_+$ is the so-called regularization parameter. Of course, problems (1) and (4) are equivalent, in the following sense: for any $\varepsilon{>}0$ such that problem (1) is feasible, a solution of (1) is either the null vector, or else is a solution of (4), for some $\tau{>}0$ [31], [47]. For solving problems of the form (4), some of the state-of-the-art algorithms belong to the iterative shrinkage/thresholding (IST) family [18], [21], [30], [36], and its two-step (TwIST [9] and FISTA [5]) and accelerated (SpaRSA [54]) variants. These methods were shown to be considerably faster than earlier methods, including *l1_ls* [38] and the codes in the ℓ_1-magic [2] and the *SparseLab* [3] toolboxes.

A key ingredient of most of these algorithms is the so-called shrinkage/thresholding/denoising function, which is the Moreau proximal mapping of the regularizer ϕ [18]. Formally, this function $\boldsymbol{\Psi}_{\tau\phi} : \mathbb{R}^n \to \mathbb{R}^n$ is defined as

$$\boldsymbol{\Psi}_{\tau\phi}(\mathbf{y}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \tau\phi(\mathbf{x}). \qquad (5)$$

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 [18]. For some choices of ϕ, the corresponding $\boldsymbol{\Psi}_{\tau\phi}$ have well known closed forms. For example, if $\phi(\mathbf{x}) \equiv \|\mathbf{x}\|_1$, then $\boldsymbol{\Psi}_{\tau\phi}(\mathbf{y}) = \text{soft}(\mathbf{y}, \tau)$, where $\text{soft}(\cdot, \tau)$ denotes the component-wise application of the soft-threshold function $y \mapsto \text{sign}(y)\max\{|y| - \tau, 0\}$.

In [2] and [28], we proposed a new algorithm called split augmented Lagrangian shrinkage algorithm (SALSA), to solve unconstrained optimization problems of the form (4) based on variable splitting [19], [53]. The idea is to transform the unconstrained problem (4) into a constrained one via a variable splitting "trick," and then attack this constrained problem using an augmented Lagrangian (AL) method [45], specifically the alternating direction method of multipliers (ADMM) [24], [32], [33]. Although AL is known to be equivalent to the Bregman iterations recently proposed to handle imaging inverse problems (see [57] and references therein), we prefer the AL perspective, rather than the Bregman iterative view, as it is a more standard/elementary tool (covered in most optimization textbooks). On several benchmark experiments (namely image deconvolution, recovery of missing pixels, and reconstruction from partial Fourier observations) using either frame-based or TV-based regularization, SALSA was found to be faster than the previous state-of-the-art methods FISTA [5], TwIST [9], and SpaRSA [54].

Variable splitting and ADMM were also recently used in [53] to obtain a fast algorithm for TV-based image deblurring; that work addresses a TV-ℓ_2 unconstrained formulation (i.e., problem (4) with ϕ a TV norm) and is termed fast total variation deconvolution (FTVd). Notice that the splitting underlying FTVd is radically different from the one used in [2], [28]. Finally, the reconstruction from partial Fourier (RecPF) algorithm was designed for signal reconstruction from partial Fourier observations and also uses ADMM; the optimization problem addressed by RecPF also has the form (4), but with a regularizer that is the sum of two terms: a TV norm and an ℓ_1

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

[2]Available at http://www.l1-magic.org.

[3]Available at http://sparselab.stanford.edu.

norm [56]. Notice that neither FTVd nor RecPF can be used to address the constrained form (1).

D. Previous Algorithms: Constrained Formulations

Although it is usually simpler to solve an unconstrained problem than a constrained one, formulations (1)–(3) have an important advantage: parameter ε has a clear meaning (it is proportional to the noise standard deviation) and is much easier to set than parameter τ in (4). Of course, one may solve (1) by solving (4) and searching for the "correct" value of τ that makes (4) equivalent to (1). Clearly, this is not efficient, as it involves solving many instances of (4). Consequently, obtaining fast algorithms for directly solving (1) (or (2) or (3)) is, thus, an important research front.

There are few efficient algorithms to solve problems of the form (1) in an image recovery context: $\mathbf{x}$ and $\mathbf{y}$ of high dimensionality ($\geq 10^4$, often $\geq 10^6$), $\mathbf{B}$ representing an operator, and ϕ a convex, nonsmooth function. A notable exception is the recent SPGL1 [52], which (as its name implies) is specifically designed for ℓ_1 regularization. As shown in [52], the methods for solving (1) available in the ℓ_1-magic package are quite inefficient for large problems. General purpose methods, such as the SeDuMi package[4], are simply not applicable when $\mathbf{B}$ is not an explicitly stored matrix, but an operator.

Another efficient algorithm for solving problems of the form (1) is the very recently proposed NESTA [6], which is based on Nesterov's first-order methods wherein an optimal gradient method is applied on a smooth approximation of the nonsmooth objective function [43], [44]. NESTA allows for minimizing either the ℓ_1 or TV norm, and also allows using synthesis or analysis formulations. Nesterov's first-order method was also recently adopted in [20] to perform TV-regularized image denoising, deblurring, and inpainting.

Your augmented Lagrangian for ℓ_1 (YALL1) is another recently proposed algorithm which addressed both unconstrained and constrained ℓ_1-regularized formulations using variable splitting and ADMM. In the splitting adopted in [55], one of the steps of ADMM turns out to be of the form (4), which is as difficult as the original problem. To sidestep that difficulty, the authors replace the exact ADMM by an inexact version, in which the nonseparable quadratic term is replaced by a separable approximation (much as in an IST scheme). As shown in the following, we exploit, in this paper, a different splitting which will not lead to a problem of the form (4). Instead, the (exact) ADMM resulting from our splitting involves a quadratic problem (which can be solved exactly in closed form for a large class of problems of interest), a shrinkage operation, and an orthogonal projection on a ball.

Finally, we should mention the Bregman iterative algorithm (BIA), recently proposed to solve (1) with $\varepsilon = 0$, that is, to minimize ϕ under the equality constraint $\mathbf{Bx} = \mathbf{y}$ (see [11], [57] and references therein). To address problems of the form (1) with $\varepsilon > 0$ (inequality constraint), the following approach has been proposed (citing verbatim from [11], changing the equation numbers and the notation to match the corresponding equations in this paper): "To solve(1), one can use the Bregman iteration for the equality constrained minimization problem[5] with an early stopping criterion

$$\|\mathbf{Bx} - \mathbf{y}\|_2 \leq \varepsilon$$

to find a good approximate solution of (1)" In other words, the BIA may only be expected to approximately solve (1), not exactly. Naturally, using an early stopping criterion (based upon the feasible set of (1) with $\varepsilon > 0$) in an algorithm designed for (and which converges to) the solution of the equality constrained problem ((1) with $\varepsilon = 0$) is not guaranteed to yield an optimal solution of the inequality constrained problem.

E. Proposed Approach

In this paper, we introduce an algorithm for solving optimization problems of the form (1). The original constrained problem (1) is transformed into an unconstrained one by adding the indicator function of the feasible set, the ellipsoid $\{\mathbf{x} : \|\mathbf{Bx} - \mathbf{y}\| \leq \varepsilon\}$, to the objective in (1). The resulting unconstrained problem is then transformed into a different constrained problem, by the application of a variable splitting operation; finally, the obtained constrained problem is dealt with using the ADMM [24], [32], [33], which belongs to the family of AL techniques [45]. Since (as SALSA), the proposed method uses variable splitting and AL optimization, we call it C-SALSA (for constrained-SALSA).

The resulting algorithm is more general than SPGL1, in the sense that it can be used with any convex regularizer ϕ for which the corresponding Moreau proximity operator (see [18]), has closed form or can be efficiently computed. In this paper, we will show examples of C-SALSA where $\mathbf{x}$ is an image, ϕ is the TV norm [48], and $\mathbf{\Psi}_{\tau\phi}$ is computed using Chambolle's algorithm [14]. Another classical choice which we will demonstrate is the ℓ_1 norm, which leads to $\mathbf{\Psi}_{\tau\phi}(\mathbf{y}) = \mathrm{soft}(\mathbf{y}, \tau)$.

C-SALSA is experimentally shown to efficiently solve image recovery problems, such as MRI reconstruction from CS-type partial Fourier observations using TV regularization, and image deblurring using wavelet-based or TV regularization, faster than SPGL1 and NESTA.

F. Organization of the Paper

The paper is organized as follows. Section II describes the basic ingredients of C-SALSA: variable splitting, ALs, and the ADMM. Section III contains the derivation leading to C-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

$$\min_{\mathbf{u} \in \mathbb{R}^n} f_1(\mathbf{u}) + f_2(\mathbf{Gu}) \tag{6}$$

where $\mathbf{G} \in \mathbb{R}^{d \times n}$, $f_1 : \mathbb{R}^n \to \bar{\mathbb{R}}$, and $f_2 : \mathbb{R}^d \to \bar{\mathbb{R}}$. Variable splitting (VS) is a simple procedure that consists in creating a

[4]Available at http://sedumi.ie.lehigh.edu.

[5]That is, problem (1) with $\varepsilon = 0$.

new variable, say $\mathbf{v}$, to serve as the argument of f_2, under the constraints that $\mathbf{v} = \mathbf{G}\mathbf{u}$, i.e.,

$$\min_{\mathbf{u}\in\mathbb{R}^n, \mathbf{v}\in\mathbb{R}^d} f_1(\mathbf{u}) + f_2(\mathbf{v}), \quad \text{subject to} \quad \mathbf{v} = \mathbf{G}\mathbf{u}. \tag{7}$$

The rationale behind VS is that it may be easier to solve the constrained problem (7) than it is to solve its equivalent unconstrained counterpart (6).

VS has been recently used in several image processing applications. A VS method was used in [53] to obtain a fast algorithm for TV-based restoration. Variable splitting was also used in [10] to handle problems involving compound regularizers. In [10], [53], the constrained problem (7) is attacked using a quadratic penalty approach, i.e., by solving

$$\min_{\mathbf{u}\in\mathbb{R}^n, \mathbf{v}\in\mathbb{R}^d} f_1(\mathbf{u}) + f_2(\mathbf{v}) + \frac{\alpha}{2}\|\mathbf{G}\mathbf{u} - \mathbf{v}\|_2^2 \tag{8}$$

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 (8) to approach that of (7), which in turn is equivalent to (6). The rationale of these methods is that each step of this alternating minimization may be much easier than the original unconstrained problem (6).

A similar VS approach underlies the recently proposed split-Bregman methods [34]. Instead of using a quadratic penalty technique, those methods attack the constrained problem directly using a Bregman iterative algorithm [57]. Moreover, the split-Bregman method with a single inner iteration is known to be equivalent to the AL method [35], [51], [57].

B. AL

Consider the constrained optimization problem with linear equality constraints

$$\min_{\mathbf{z}\in\mathbb{R}^n} E(\mathbf{z}) \quad \text{subject to} \quad \mathbf{A}\mathbf{z} - \mathbf{b} = \mathbf{0} \tag{9}$$

where $\mathbf{b} \in \mathbb{R}^p$ and $\mathbf{A} \in \mathbb{R}^{p\times n}$, i.e., there are p linear equality constraints. The so-called AL for this problem is defined as

$$\mathcal{L}_A(\mathbf{z}, \boldsymbol{\lambda}, \mu) = E(\mathbf{z}) + \boldsymbol{\lambda}^T(\mathbf{b} - \mathbf{A}\mathbf{z}) + \frac{\mu}{2}\|\mathbf{A}\mathbf{z} - \mathbf{b}\|_2^2 \tag{10}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^p$ is a vector of Lagrange multipliers and $\mu \geq 0$ is called the AL penalty parameter [45]. The so-called augmented Lagrangian method (ALM) [45], also known as the method of multipliers (MM) [37], [46], iterates between minimizing $\mathcal{L}_A(\mathbf{z}, \boldsymbol{\lambda}, \mu)$ with respect to $\mathbf{z}$, keeping $\boldsymbol{\lambda}$ fixed, and updating $\boldsymbol{\lambda}$, until some convergence criterion is satisfied.

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. $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + \mu(\mathbf{A}\mathbf{z}_{k+1} - \mathbf{b})$
5. $k \leftarrow k + 1$
6. **until** some stopping criterion is satisfied.

It is also possible (and even recommended) to update the value of μ in each iteration [4], [45]. 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 (9).

After a straightforward complete-the-squares procedure, the terms added to $E(\mathbf{z})$ in the AL $\mathcal{L}_A(\mathbf{z}, \boldsymbol{\lambda}_k, \mu)$ can be written as a single quadratic term (plus a constant independent of $\mathbf{z}$, thus, irrelevant to the ALM/MM), leading to the following alternative form of the algorithm (which makes clear its equivalence with the Bregman iterative method [57]):

Algorithm *ALM/MM (version II)*

1. Set $k = 0$, choose $\mu > 0$ and $\mathbf{d}_0$
2. **repeat**
3. $\mathbf{z}_{k+1} \in \arg\min_{\mathbf{z}} E(\mathbf{z}) + (\mu/2)\|\mathbf{A}\mathbf{z} - \mathbf{d}_k\|_2^2$
4. $\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{A}\mathbf{z}_{k+1} - \mathbf{b})$
5. $k \leftarrow k + 1$
6. **until** some stopping criterion is satisfied.

C. ALM/MM for Variable Splitting and ADMM

The constrained problem (7) can be written as (9) by defining $E(\mathbf{z}) \equiv f_1(\mathbf{u}) + f_2(\mathbf{v})$ and setting

$$\begin{aligned} \mathbf{z} &\equiv \left[\mathbf{u}^T \mathbf{v}^T\right]^T \\ \mathbf{b} &= \mathbf{0} \\ \mathbf{A} &= [\mathbf{G} - \mathbf{I}]. \end{aligned} \tag{11}$$

With these definitions in place, Steps 3 and 4 of the ALM/MM (version II) become

$$\begin{aligned} (\mathbf{u}_{k+1}, \mathbf{v}_{k+1}) &\in \arg\min_{\mathbf{u},\mathbf{v}} f_1(\mathbf{u}) \\ &\qquad + f_2(\mathbf{v}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{u} - \mathbf{v} - \mathbf{d}_\mathbf{k}\|_2^2 \\ \mathbf{d}_{k+1} &= \mathbf{d}_k - (\mathbf{G}\mathbf{u}_{k+1} - \mathbf{v}_{k+1}). \end{aligned} \tag{12}$$

The minimization problem yielding $(\mathbf{u}_{k+1}, \mathbf{v}_{k+1})$ is not trivial since, in general, it involves a nonseparable quadratic term and possibly nonsmooth terms. A natural approach is to use a nonlinear block-Gauss-Seidel (NLBGS) technique which alternates between minimizing with respect to $\mathbf{u}$ and $\mathbf{v}$ while keeping the other fixed. Of course this raises several questions: for a given $\mathbf{d}_k$, how much computational effort should be spent in this problem? Does the NLBGS procedure converge? The simplest answer to these questions is given in the form of the so-called ADMM [24], [32], [33], which is simply an ALM/MM in which only one NLBGS step is performed in each outer iteration.

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

Algorithm *ADMM*

1. Set $k = 0$, choose μ>0, $\mathbf{v}_0$, $\mathbf{d}_0$.
2. **repeat**
3. $\mathbf{u}_{k+1} \in \arg\min_{\mathbf{u}} f_1(\mathbf{u}) + (\mu/2)\|\mathbf{G}\mathbf{u} - \mathbf{v}_k - \mathbf{d}_k\|_2^2$
4. $\mathbf{v}_{k+1} \in \arg\min_{\mathbf{v}} f_2(\mathbf{v}) + (\mu/2)\|\mathbf{G}\mathbf{u}_{k+1} - \mathbf{v} - \mathbf{d}_k\|_2^2$
5. $\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\mathbf{u}_{k+1} - \mathbf{v}_{k+1})$
6. $k \leftarrow k+1$
7. **until** some stopping criterion is satisfied.

For later reference, we now recall the theorem by Eckstein and Bertsekas [24, Theorem 8] in which convergence of (a generalized version of) ADMM is shown.

Theorem 1 (Eckstein-Bertsekas, [24]): Consider problem (6), where $\mathbf{G}$ has full column rank, and f_1 and f_2 are closed, proper, convex functions. Consider arbitrary $\mu > 0$ and $\mathbf{u}_0$, $\mathbf{d}_0$, $\mathbf{v}_0 \in \mathbb{R}^p$. Let $\{\eta_k \geq 0, k = 0, 1, \ldots\}$ and $\{\nu_k \geq 0, k = 0, 1, \ldots\}$ be two sequences such that

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

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

$$\left\|\mathbf{u}_{k+1} - \arg\min_{\mathbf{u}} f_1(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{u} - \mathbf{v}_k - \mathbf{d}_k\|_2^2\right\| \leq \eta_k$$
$$\left\|\mathbf{v}_{k+1} - \arg\min_{\mathbf{v}} f_2(\mathbf{v}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{u}_{k+1} - \mathbf{v} - \mathbf{d}_k\|_2^2\right\| \leq \nu_k$$
$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\mathbf{u}_{k+1} - \mathbf{v}_{k+1}).$$

Then, if (6) has a solution, say $\mathbf{u}^*$, then the sequence $\{\mathbf{u}_k\}$ converges to $\mathbf{u}^*$. If (6) does not have a solution, then at least one of the sequences $\{\mathbf{u}_k\}$ or $\{\mathbf{d}_k\}$ diverges.

Notice that the ADMM as defined previously (if each step is implemented exactly) generates sequences $\{\mathbf{u}_k\}$, $\{\mathbf{v}_k\}$, and $\{\mathbf{d}_k\}$ that satisfy the conditions in Theorem 1 in a strict sense (i.e., with $\eta_k = \nu_k = 0$). The remaining key condition for convergence is then that $\mathbf{G}$ has full column rank. One of the important corollaries of this theorem is that it is not necessary to exactly solve the minimizations in lines 3 and 4 of ADMM; as long as the sequence of errors are absolutely summable, convergence is not compromised.

The proof of Theorem 1 is based upon the equivalence between ADMM and the Douglas–Rachford splitting (DRS) applied to the dual of problem (6). The DRS was recently used for image recovery problems in [17]. For recent and comprehensive reviews of ALM, ADMM, DRS, and their relationship with Bregman and split-Bregman methods, see [35] and [51].

D. Variant of ADMM

Consider a generalization of problem (6), where instead of two functions, there are J functions, that is

$$\min_{\mathbf{u}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\mathbf{u}) \tag{13}$$

where $g_j : \mathbb{R}^{p_j} \rightarrow \bar{\mathbb{R}}$ are closed, proper, convex functions, and $\mathbf{H}^{(j)} \in \mathbb{R}^{p_j \times d}$ are arbitrary matrices. The minimization problem (13) can be written as (6) using the following correspondences: $f_1 = 0$

$$\mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \in \mathbb{R}^{p \times d} \tag{14}$$

where $p = p_1 + \ldots + p_J$, and $f_2 : \mathbb{R}^{p\times d} \rightarrow \bar{\mathbb{R}}$ given by

$$f_2(\mathbf{v}) = \sum_{j=1}^{J} g_j(\mathbf{v}^{(j)}) \tag{15}$$

where $\mathbf{v}^{(j)} \in \mathbb{R}^{p_j}$ and $\mathbf{v} = [(\mathbf{v}^{(1)})^T, \ldots, (\mathbf{v}^{(J)})^T]^T \in \mathbb{R}^p$. We now simply apply ADMM (as given in the previous subsection), with

$$\mathbf{d}_k = \begin{bmatrix} \mathbf{d}_k^{(1)} \\ \vdots \\ \mathbf{d}_k^{(J)} \end{bmatrix}, \mathbf{v}_k = \begin{bmatrix} \mathbf{v}_k^{(1)} \\ \vdots \\ \mathbf{v}_k^{(J)} \end{bmatrix}.$$

Moreover, the fact that $f_1 = 0$ turns Step 3 of the algorithm into a simple quadratic minimization problem, which has a unique solution if $\mathbf{G}$ has full column rank

$$\begin{aligned} &\arg\min_{\mathbf{u}} \|\mathbf{G}\mathbf{u} - \boldsymbol{\zeta}_k\|_2^2 \\ &= (\mathbf{G}^H\mathbf{G})^{-1}\mathbf{G}^H\boldsymbol{\zeta}_k \\ &= \left[\sum_{j=1}^{J} (\mathbf{H}^{(j)})^H\mathbf{H}^{(j)}\right]^{-1} \sum_{j=1}^{J} (\mathbf{H}^{(j)})^H\boldsymbol{\zeta}_k^{(j)} \end{aligned} \tag{16}$$

where $\boldsymbol{\zeta}_k = \mathbf{v}_k + \mathbf{d}_k$ (and, naturally, $\boldsymbol{\zeta}_k^{(j)} = \mathbf{u}_k^{(j)} + \mathbf{d}_k^{(j)}$) and the second equality results from the particular structure of $\mathbf{G}$ in (14).

Furthermore, our particular way of mapping problem (13) into problem (6) allows decoupling the minimization in Step 4 of ADMM into a set of J independent ones. In fact

$$\mathbf{v}_{k+1} \leftarrow \arg\min_{\mathbf{v}} f_2(\mathbf{v}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{u}_{k+1} - \mathbf{v} - \mathbf{d}_k\|_2^2 \tag{17}$$

TABLE II
IMAGE DEBLURRING USING WAVELETS (REDUNDANT)—COMPUTATIONAL LOAD

Expt.	Avg. calls to $\mathbf{B}, \mathbf{B}^H$ (min/max)			Iterations			CPU time (seconds)		
	SPGL1	NESTA	C-SALSA	SPGL1	NESTA	C-SALSA	SPGL1	NESTA	C-SALSA
1	1029 (659/1290)	3520 (3501/3541)	398 (388/406)	340	880	134	441.16	590.79	100.72
2A	511 (279/663)	4897 (4777/4981)	451 (442/460)	160	1224	136	202.67	798.81	98.85
2B	377 (141/532)	3397 (3345/3473)	362 (355/370)	98	849	109	120.50	557.02	81.69
3A	675 (378/772)	2622 (2589/2661)	172 (166/175)	235	656	58	266.41	423.41	42.56
3B	404 (300/475)	2446 (2401/2485)	134 (130/136)	147	551	41	161.17	354.59	29.57

TABLE III
IMAGE DECONVOLUTION USING WAVELETS (REDUNDANT, ANALYSIS PRIOR)—COMPUTATIONAL LOAD

Expt.	Avg. calls to $\mathbf{B}, \mathbf{B}^H$ (min/max)		Iterations		CPU time (seconds)	
	NESTA	C-SALSA	NESTA	C-SALSA	NESTA	C-SALSA
1	2881 (2861/2889)	413 (404/419)	720	138	353.88	80.32
2A	2451 (2377/2505)	362 (344/371)	613	109	291.14	62.65
2B	2139 (2065/2197)	290 (278/299)	535	87	254.94	50.14
3A	2203 (2181/2217)	137 (134/143)	551	42	261.89	23.83
3B	1967 (1949/1985)	116 (113/119)	492	39	236.45	22.38

can be written as

$$\begin{bmatrix} \mathbf{v}_{k+1}^{(1)} \\ \vdots \\ \mathbf{v}_{k+1}^{(J)} \end{bmatrix} \leftarrow \arg \min_{\mathbf{v}^{(1)},\ldots,\mathbf{v}^{(J)}} g_1(\mathbf{v}^{(1)}) + \cdots + g_J(\mathbf{v}^{(J)}) + \frac{\mu}{2} \left\| \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \mathbf{u}_{k+1} - \begin{bmatrix} \mathbf{v}^{(1)} \\ \vdots \\ \mathbf{v}^{(J)} \end{bmatrix} - \begin{bmatrix} \mathbf{d}_k^{(1)} \\ \vdots \\ \mathbf{d}_k^{(J)} \end{bmatrix} \right\|_2^2 .$$

Clearly, the minimizations with respect to $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(J)}$ are decoupled, thus, can be solved separately, leading to

$$\mathbf{v}_{k+1}^{(j)} \leftarrow \arg \min_{\mathbf{v} \in \mathbb{R}^{p_j}} g_j(\mathbf{v}) + \frac{\mu}{2} \left\| \mathbf{v} - \mathbf{s}_k^{(1)} \right\|_2^2 \qquad (18)$$

for $j = 1, \ldots, J$, where

$$\mathbf{s}_k^{(j)} = \mathbf{H}^{(j)} \mathbf{u}_{k+1} - \mathbf{d}_k^{(j)}.$$

Since this algorithm is exactly an ADMM, and since all the functions g_j, for $j = 1, \ldots, J$, are closed, proper, and convex, convergence is guaranteed if $\mathbf{G}$ has full column rank. Actually, this full column rank condition is also required for the inverse in (16) to exist. Finally, notice that the update equations in (18) can be written as

$$\mathbf{v}_{k+1}^{(j)} = \bar{\mathbf{\Psi}}_{g_j/\mu}(\mathbf{s}_k^{(j)}) \qquad (19)$$

where the $\mathbf{\Psi}_{g_j/\mu}$ are, by definition, the Moreau proximal mappings of $g_1/\mu, \ldots, g_J/\mu$.

In summary, the variant of ADMM (herein referred to as ADMM-2) that results from the formulation just presented is described in the following algorithmic framework.

Algorithm *ADMM-2*

1. Set $k = 0$, choose $\mu > 0$, $\mathbf{v}_0^{(1)}, \ldots, \mathbf{v}_0^{(J)}$, $\mathbf{d}_0^{(1)}, \ldots, \mathbf{d}_0^{(J)}$.
2. **repeat**
3. **for** $i = 1, \ldots, J$
4. **do** $\boldsymbol{\zeta}_k^{(i)} = \mathbf{v}_k^{(i)} + \mathbf{d}_k^{(i)}$
5. $\mathbf{u}_{k+1} = \left[\sum_{j=1}^{J} (\mathbf{H}^{(j)})^H \mathbf{H}^{(j)} \right]^{-1} \sum_{j=1}^{J} \left(\mathbf{H}^{(j)} \right)^H \boldsymbol{\zeta}_k^{(j)}$
6. **for** $i = 1, \ldots, J$
7. **do** $\mathbf{v}_{k+1}^{(i)} = \mathbf{\Psi}_{g_i/\mu} \left(\mathbf{H}^{(i)} \mathbf{u}_{k+1} - \mathbf{d}_k^{(i)} \right)$
8. $\mathbf{d}_{k+1}^{(i)} = \mathbf{d}_k^{(i)} - \mathbf{H}^{(i)} \mathbf{u}_{k+1} + \mathbf{v}_{k+1}^{(i)}$
9. $k \leftarrow k + 1$
10. **until** some stopping criterion is satisfied.

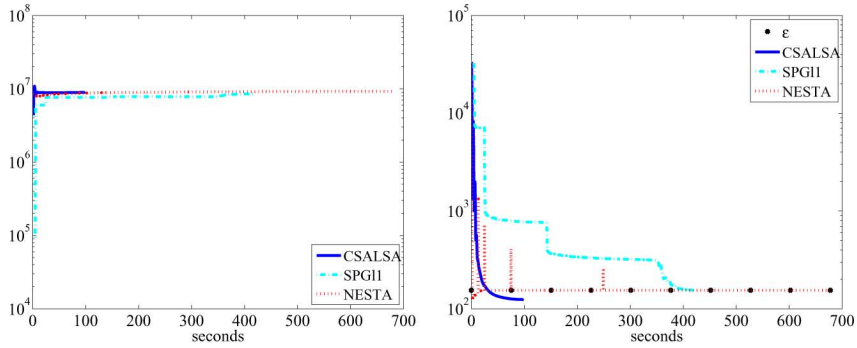
III. PROPOSED METHOD

We now apply the algorithmic framework described in the previous section to the basic problem (1) [which includes (2) as a special case], as well as the analysis formulation (3).

A. Problem (1)

For the constrained optimization problem (1), the feasible set is the ellipsoid

$$E(\varepsilon, \mathbf{B}, \mathbf{y}) = \{ \mathbf{x} \in \mathbb{R}^n : \| \mathbf{B}\mathbf{x} - \mathbf{y} \|_2 \leq \varepsilon \} \qquad (20)$$



(a) (b)

Fig. 1. Image deblurring with wavelets (synthesis prior, redundant Haar wavelets), 9×9 uniform blur, $\sigma = 0.56$. (a) Evolution of the objective function $\|\mathbf{x}\|_1$ over time. (b) Quadratic constraint $\|\mathbf{AWx} - \mathbf{y}\|_2$ over time.

TABLE IV
IMAGE DEBLURRING USING WAVELETS (ORTHOGONAL)—COMPUTATIONAL LOAD

Expt.	Avg. calls to $\mathbf{B}, \mathbf{B}^H$ (min/max)			Iterations			CPU time (seconds)		
	SPGL1	NESTA	C-SALSA	SPGL1	NESTA	C-SALSA	SPGL1	NESTA	C-SALSA
1	730 (382/922)	13901 (13871/13931)	494 (424/748)	298	3475	166	46.64	622.09	23.91
2A	352 (191/480)	1322 (1301/1329)	205 (202/205)	128	331	69	19.21	58.29	10.07
2B	207 (128/254)	1218 (1193/1261)	123 (115/133)	87	305	42	12.23	52.92	6.35
3A	248 (161/320)	1421 (1413/1433)	118 (115/121)	104	355	40	14.98	58.693	5.57
3B	170 (114/220)	4408 (4345/4545)	258 (94/328)	72	1102	87	9.51	181.83	11.93

TABLE V
IMAGE DEBLURRING USING WAVELETS (ORTHOGONAL, ANALYSIS PRIOR)—COMPUTATIONAL LOAD

Expt.	Avg. calls to $\mathbf{B}, \mathbf{B}^H$ (min/max)		Iterations		CPU time (seconds)	
	NESTA	C-SALSA	NESTA	C-SALSA	NESTA	C-SALSA
1	8471 (8413/8553)	387 (380/395)	2118	117	300.60	16.51
2A	2463 (2445/2489)	377 (371/383)	616	126	311.49	77.75
2B	2159 (2097/2253)	300 (290/317)	540	101	280.35	59.75
3A	2203 (2165/2229)	153 (149/155)	551	52	282.12	32.02
3B	4710 (4577/4829)	212 (104/374)	1178	59	167.73	7.89

which is possibly infinite in some directions (since $\mathbf{B}$ may be singular). Problem (1) can be written as an unconstrained problem, with a discontinuous objective

$$\min_{\mathbf{x}} \phi(\mathbf{x}) + \iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}(\mathbf{Bx}) \tag{21}$$

where $\iota_S : \mathbb{R}^m \to \bar{\mathbb{R}}$ denotes the indicator function of set $S \subset \mathbb{R}^m$

$$\iota_S(\mathbf{s}) = \begin{cases} 0, & \text{if } \mathbf{s} \in S \\ +\infty, & \text{if } \mathbf{s} \notin S. \end{cases} \tag{22}$$

Notice that $E(\varepsilon, \mathbf{I}, \mathbf{y})$ is simply a closed ε-radius Euclidean ball centered at $\mathbf{y}$.

Problem (21) has the form (13) with $J = 2$ and

$$g_1 \equiv \phi \tag{23}$$

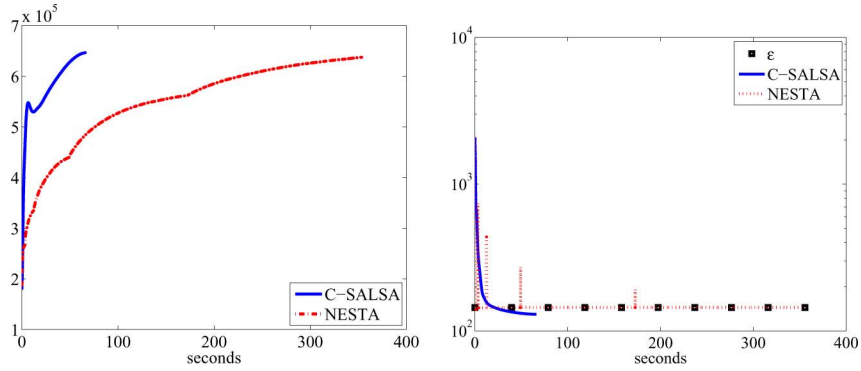
$$g_2 \equiv \iota_{E(\varepsilon,\mathbf{I},\mathbf{y})} \tag{24}$$

$$\mathbf{H}^{(1)} \equiv \mathbf{I} \tag{25}$$

$$\mathbf{H}^{(2)} \equiv \mathbf{B}. \tag{26}$$

Instantiating ADMM-2 to this particular case requires the definition of the Moreau proximal maps associated with $g_1 \equiv \phi$ and $g_2 \equiv \iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}$. Concerning ϕ, the regularizer, we assume that $\mathbf{\Psi}_{\tau\phi}(\cdot)$ (see (5)) can be computed efficiently. This is of course the case of $\phi(\mathbf{x}) \equiv \|\mathbf{x}\|_1$, for which $\mathbf{\Psi}_{\tau\phi}$ is simply a soft threshold. If ϕ is the TV norm, we may use one the fast algorithms available to compute the corresponding denoising function [14], [20]. The Moreau proximal map of $g_2 \equiv \iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}$ is defined as

$$\mathbf{\Psi}_{\iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}/\mu}(\mathbf{s}) = \arg\min_{\mathbf{x}} \frac{\iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}(\mathbf{x})}{\mu} + \frac{1}{2}\|\mathbf{x} - \mathbf{s}\|_2^2 \tag{27}$$



(a) (b)

Fig. 2. Image deblurring (uniform blur) with TV regularization. (a) Evolution of the objective function over time. (b) Evolution of the constraint $\|\mathbf{Bx}-\mathbf{y}\|$ over time.

TABLE VI
IMAGE DEBLURRING USING TV–COMPUTATIONAL LOAD

Expt.	Avg. calls to $\mathbf{B}, \mathbf{B}^H$ (min/max)		Iterations		CPU time (seconds)	
	NESTA	C-SALSA	NESTA	C-SALSA	NESTA	C-SALSA
1	7783 (7767/7795)	695 (680/710)	1945	232	311.98	62.56
2A	7323 (7291/7351)	559 (536/578)	1830	150	279.36	38.63
2B	6828 (6775/6883)	299 (269/329)	1707	100	265.35	25.47
3A	6594 (6513/6661)	176 (98/209)	1649	59	250.37	15.08
3B	5514 (5417/5585)	108 (104/110)	1379	37	210.94	9.23

which is obviously independent of μ and is simply the orthogonal projection of $\mathbf{s}$ on the closed ε-radius ball centered at $\mathbf{y}$

$$\mathbf{\Psi}_{\iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}}(\mathbf{s}) = \mathbf{y} + \begin{cases} \varepsilon\frac{\mathbf{s}-\mathbf{y}}{\|\mathbf{s}-\mathbf{y}\|_2}, & \text{if } \|\mathbf{s}-\mathbf{y}\|_2 > \varepsilon \\ \mathbf{s}-\mathbf{y}, & \text{if } \|\mathbf{s}-\mathbf{y}\|_2 \leq \varepsilon. \end{cases} \tag{28}$$

We are now in a position to instantiate ADMM-2 for solving (21) [equivalently (1)]. The resulting algorithm, which we call C-SALSA-1, is as follows.

Algorithm *C-SALSA-1*

1. Set $k=0$, choose $\mu>0$, $\mathbf{v}_0^{(1)}$, $\mathbf{v}_0^{(2)}$, $\mathbf{d}_0^{(2)}$, $\mathbf{d}_0^{(2)}$.
2. **repeat**
3. $\mathbf{r}_k = \mathbf{v}_0^{(1)} + \mathbf{d}_0^{(1)} + \mathbf{B}^H\left(\mathbf{v}_0^{(2)} + \mathbf{d}_0^{(2)}\right)$
4. $\mathbf{u}_{k+1} = \left(\mathbf{I} + \mathbf{B}^H\mathbf{B}\right)^{-1}\mathbf{r}_k$
5. $\mathbf{v}_{k+1}^{(1)} = \mathbf{\Psi}_{\phi/\mu}\left(\mathbf{u}_{k+1} - \mathbf{d}_k^{(1)}\right)$
6. $\mathbf{v}_{k+1}^{(2)} = \mathbf{\Psi}_{\iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}}\left(\mathbf{B}\mathbf{u}_{k+1} - \mathbf{d}_k^{(2)}\right)$
7. $\mathbf{d}_{k+1}^{(1)} = \mathbf{d}_k^{(1)} - \mathbf{u}_{k+1} + \mathbf{v}_{k+1}^{(1)}$
8. $\mathbf{d}_{k+1}^{(2)} = \mathbf{d}_k^{(2)} - \mathbf{B}\mathbf{u}_{k+1} + \mathbf{v}_{k+1}^{(2)}$
9. $k \leftarrow k+1$
10. **until** some stopping criterion is satisfied.

The issue of how to efficiently solve the linear system of equations in line 4 of C-SALSA-1 will be addressed in Section III-C.

Convergence of C-SALSA-1 is guaranteed by Theorem 1 since it is an instance of ADMM with

$$\mathbf{G} \equiv \begin{bmatrix} \mathbf{I} \\ \mathbf{B} \end{bmatrix} \tag{29}$$

which is a full column rank matrix, and both ϕ and $\iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}$ are closed, proper, convex functions.

Finally, notice that to apply C-SALSA-1 to problem (2) we simply have to replace $\mathbf{B}$ with $\mathbf{BW}$.

B. Problem (3)

Problem (3) can also be written as an unconstrained problem

$$\min_{\mathbf{x}} \phi(\mathbf{Px}) + \iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}(\mathbf{Bx}) \tag{30}$$

which has the form (13) with $J=2$ and

$$g_1 \equiv \phi \tag{31}$$
$$g_2 \equiv \iota_{E(\varepsilon,\mathbf{I},\mathbf{y})} \tag{32}$$
$$\mathbf{H}^{(1)} \equiv \mathbf{P} \tag{33}$$
$$\mathbf{H}^{(2)} \equiv \mathbf{B}. \tag{34}$$

The resulting ADMM algorithm, called C-SALSA-2, is similar to C-SALSA-1, with only a few minor differences.

Algorithm *C-SALSA-2*

1. Set $k = 0$, choose μ>0, $\mathbf{v}_0^{(1)}$, $\mathbf{v}_0^{(2)}$, $\mathbf{d}_0^{(2)}$, $\mathbf{d}_0^{(2)}$.
2. **repeat**
3. $\mathbf{r}_k = \mathbf{P}^H \left(\mathbf{v}_k^{(1)} + \mathbf{d}_k^{(1)}\right) + \mathbf{B}^H \left(\mathbf{v}_k^{(2)} + \mathbf{d}_k^{(2)}\right)$
4. $\mathbf{u}_{k+1} = (\mathbf{P}^H\mathbf{P} + \mathbf{B}^H\mathbf{B})^{-1}\mathbf{r}_k$
5. $\mathbf{v}_{k+1}^{(1)} = \mathbf{\Psi}_{\phi/\mu}\left(\mathbf{P}\mathbf{u}_{k+1} - \mathbf{d}_k^{(1)}\right)$
6. $\mathbf{v}_{k+1}^{(2)} = \mathbf{\Psi}_{\iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}}\left(\mathbf{B}\mathbf{u}_{k+1} - \mathbf{d}_k^{(2)}\right)$
7. $\mathbf{d}_{k+1}^{(1)} = \mathbf{d}_k^{(1)} - \mathbf{P}\mathbf{u}_{k+1} + \mathbf{v}_{k+1}^{(1)}$
8. $\mathbf{d}_{k+1}^{(2)} = \mathbf{d}_k^{(2)} - \mathbf{B}\mathbf{u}_{k+1} + \mathbf{v}_{k+1}^{(2)}$
9. $k \leftarrow k + 1$
10. **until** some stopping criterion is satisfied.

In this paper, we assume that $\mathbf{P}$ is the analysis operator of a 1-tight (Parseval) frame, thus, $\mathbf{P}^H\mathbf{P} = \mathbf{I}$ and line 4 of C-SALSA-2 is similar to line 4 of C-SALSA-1

$$\mathbf{u}_{k+1} = (\mathbf{I} + \mathbf{B}^H\mathbf{B})^{-1}\mathbf{r}_k. \tag{35}$$

The issue of how to efficiently solve this linear system will be addressed in Section III-C.

Since both ϕ and $\iota_{E(\varepsilon,\mathbf{I},\mathbf{y})}$ are closed, proper, convex functions, convergence of C-SALSA-2 holds (by Theorem 1) if

$$\mathbf{G} = \begin{bmatrix} \mathbf{P} \\ \mathbf{B} \end{bmatrix} \tag{36}$$

is a full column rank matrix. This is of course true if $\mathbf{P}$ is itself a full column rank matrix, which is the case if $\mathbf{P}$ is the analysis operator of a tight frame [42].

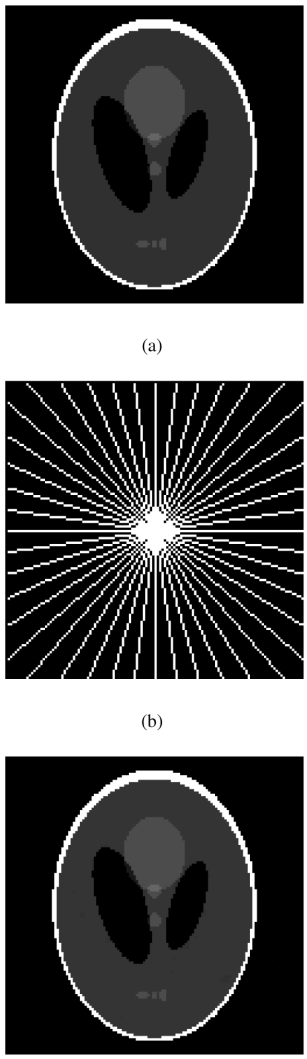
C. Solving (35)

As mentioned in Section I-C, in most imaging problems of interest, it may not be feasible to explicitly form matrix $\mathbf{B}$. This might suggest that it is not easy, or even feasible, to compute the inverse of $\left(\mathbf{I} + \mathbf{B}^H\mathbf{B}\right)$. However, as shown next, in a number of problems of interest, this inverse can be computed very efficiently with $O(n \log n)$ cost, or even $O(n)$ cost in some formulations of inpainting.

1) Deconvolution With Analysis Formulation: Let us first consider analysis formulations of the form (1) or (3) to image deconvolution problems. If matrix $\mathbf{B}$ represents a 2-D cyclic convolution (periodic boundary conditions), it is a block circulant matrix with circulant blocks that can be factorized as $\mathbf{B} = \mathbf{U}^H\mathbf{D}\mathbf{U}$, where $\mathbf{U}$ is the unitary matrix $(\mathbf{U}^H = \mathbf{U}^{-1})$ representing the discrete Fourier transform (DFT) and $\mathbf{D}$ is diagonal. Thus

$$(\mathbf{B}^H\mathbf{B} + \mathbf{I})^{-1} = \mathbf{U}^H(|\mathbf{D}|^2 + \mathbf{I})^{-1}\mathbf{U} \tag{37}$$

where $|\mathbf{D}|^2$ is the matrix with squared absolute values of the entries of $\mathbf{D}$. Since $|\mathbf{D}|^2 + \mathbf{I}$ is diagonal, its inversion cost is $O(n)$. Products by $\mathbf{U}$ and $\mathbf{U}^H$ have $O(n \log n)$ cost, using the FFT algorithm.



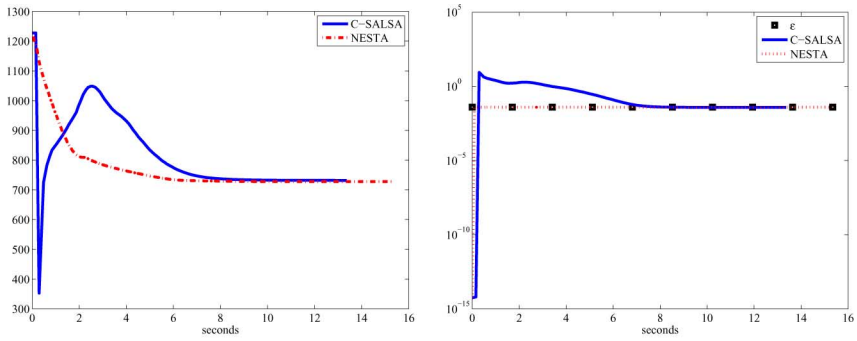
(a) (b) (c)

Fig. 3. MRI reconstruction. (a) 128 × 128 Shepp–Logan phantom. (b) Mask with 22 radial lines. (c) Image estimated using C-SALSA.

If the convolution represented by $\mathbf{B}$ is not cyclic, then $\mathbf{B}$ is a (non-circulant) Toeplitz matrix and it cannot be factorized as $\mathbf{B} = \mathbf{U}^H\mathbf{D}\mathbf{U}$. In this case, solving the linear system in (35) is not as simple as applying (37). However, there are very fast preconditioned conjugate gradient (PCG) methods for Topelitz matrices, which use the FFT to perform the required matrix-vector products, thus, having $O(n \log n)$ cost (see [22] for very recent work on this topic and pointers to a vast literature).

2) Deconvolution With Synthesis Formulation: In this case, as seen in Section I-B, we have $\mathbf{BW}$ instead of $\mathbf{B}$, and even if $\mathbf{B}$ is a cyclic convolution, $\mathbf{BW}$ is not diagonalizable by the DFT. To sidestep this difficulty, we assume that $\mathbf{W}$ contains a 1-tight (Parseval) frame (i.e., $\mathbf{W}\mathbf{W}^H = \mathbf{I}$). Using the Sherman–Morrison–Woodbury (SMW) matrix inversion lemma

$$(\mathbf{W}^H\mathbf{B}^H\mathbf{B}\mathbf{W} + \mathbf{I})^{-1} = \mathbf{I} - \mathbf{W}^H \underbrace{\mathbf{B}^H(\mathbf{B}\mathbf{B}^H + \mathbf{I})^{-1}\mathbf{B}}_{\mathbf{F}}\mathbf{W} \tag{38}$$



(a) (b)

Fig. 4. MRI reconstruction with TV regularization. (a) Evolution of the objective function over time. (b) Evolution of the constraint $\|\mathbf{Bx}-\mathbf{y}\|$ over time.

thus, line 4 of C-SALSA-1 and C-SALSA-2 can be written as

$$\mathbf{u}_{k+1} = (\mathbf{r}_k - \mathbf{W}^H\mathbf{F}\mathbf{W}\mathbf{r}_k). \tag{39}$$

If $\mathbf{B}$ represents a cyclic convolution, then, as shown previously, it can be factorized as $\mathbf{B} = \mathbf{U}^H\mathbf{D}\mathbf{U}$, thus, multiplication by $\mathbf{F}$ corresponds to applying an image filter in the Fourier domain

$$\mathbf{F} = \mathbf{U}^H\mathbf{D}^*(|\mathbf{D}|^2+\mathbf{I})^{-1}\mathbf{D}\mathbf{U}$$

which has $O(n\log n)$ cost, since all the matrices in $\mathbf{D}^*\left(|\mathbf{D}|^2+\mathbf{I}\right)^{-1}\mathbf{D}$ are diagonal and the products by $\mathbf{U}$ and $\mathbf{U}^H$ are carried out via the FFT. The cost of (38) will, thus, be either $O(n\log n)$ or the cost of the products by $\mathbf{W}^H$ and $\mathbf{W}$. In the case of a noncyclic convolution, $\mathbf{B}$ is Toeplitz and cannot be inverted directly as in (37); in this case, we would have to resort, as mentioned previously, to a PCG algorithm.

For most tight frames used in image processing, there are fast $O(n\log n)$ algorithms to compute the products by $\mathbf{W}^H$ and $\mathbf{W}$ [42]. For example, in the case of translation-invariant wavelet transforms, these products can be computed using the undecimated wavelet transform with $O(n\log n)$ total cost [40]. Curvelets also constitute a Parseval frame for which fast $O(n\log n)$ implementations of the forward and inverse transform exist [12]. Yet another example of a redundant Parseval frame is the complex wavelet transform, which has $O(n)$ computational cost [39], [49]. In conclusion, for a large class of choices of $\mathbf{W}$, each iteration of the C-SALSA algorithm has $O(n\log n)$ cost.

3) Missing Pixels: Image Inpainting: In the analysis prior formulation of this problem, the observation matrix $\mathbf{B}$ models the loss of some image pixels. It 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 $\mathbf{B}\mathbf{B}^H = \mathbf{I}$. Using this fact together with the SMW formula leads to

$$(\mathbf{B}^H\mathbf{B}+\mathbf{I})^{-1} = \mathbf{I}-\mathbf{B}^H(\mathbf{I}+\mathbf{B}\mathbf{B}^H)^{-1}\mathbf{B} \tag{40}$$

$$= \mathbf{I}-\frac{1}{2}\mathbf{B}^H\mathbf{B}. \tag{41}$$

Since $\mathbf{B}^H\mathbf{B}$ is equal to an identity matrix with some zeros in the diagonal (corresponding to the positions of the missing observations), the matrix in (41) is diagonal with elements either equal to 1 or 1/2. Consequently, line 4 of C-SALSA-1 and C-SALSA-2 corresponds to multiplying this diagonal matrix by $\mathbf{r}_k$, obviously with $O(n)$ cost.

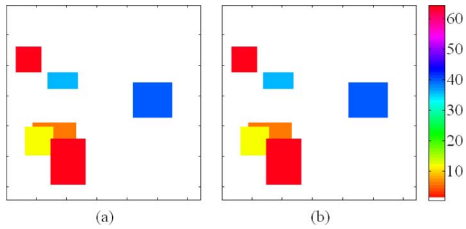


Fig. 5. TV based image reconstruction. (a) Original image with dynamic range = 40 dB. (b) Estimate using C-SALSA.

In the frame-based synthesis formulation, we have $\mathbf{BW}$ instead of $\mathbf{B}$. Using the SMW formula yet again, and the facts that $\mathbf{B}\mathbf{B}^H = \mathbf{I}$ and $\mathbf{W}\mathbf{W}^H = \mathbf{I}$, we have

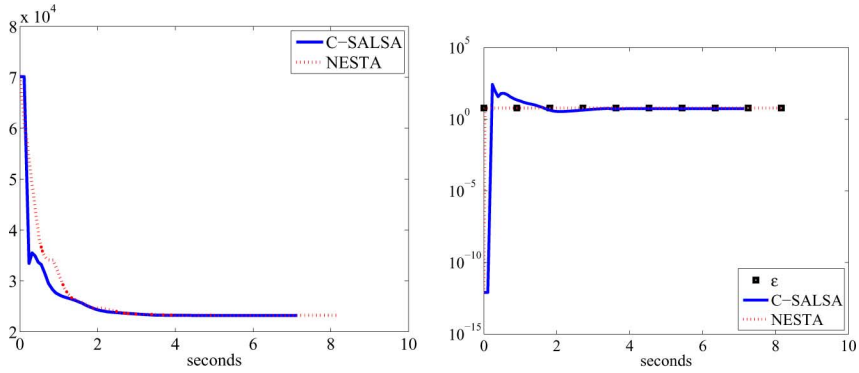
$$(\mathbf{W}^H\mathbf{B}^H\mathbf{B}\mathbf{W}+\mathbf{I})^{-1} = \mathbf{I}-\frac{1}{2}\mathbf{W}^H\mathbf{B}^H\mathbf{B}\mathbf{W}. \tag{42}$$

As noted in the previous paragraph, $\mathbf{A}^H\mathbf{A}$ is equal to an identity matrix with zeros in the diagonal, i.e., a binary mask. Thus, the multiplication by $\mathbf{W}^H\mathbf{A}^H\mathbf{A}\mathbf{W}$ corresponds to synthesizing the image, multiplying it by this mask, and computing the representation coefficients of the result. In conclusion, the cost of line 4 of C-SALSA-1 and C-SALSA-2 is again that of the products by $\mathbf{W}$ and $\mathbf{W}^H$, usually $O(n\log n)$.

4) Partial Fourier Observations (MRI Reconstruction): Finally, we consider the case of partial Fourier observations, which is used to model MRI acquisition and has been the focus of recent interest due to its connection to compressed sensing [13], [41]. In the analysis formulation, $\mathbf{B} = \mathbf{M}\mathbf{U}$, where $\mathbf{M}$ is an $m\times n$ binary matrix ($m<n$) again, formed by a subset of rows of the identity, and $\mathbf{U}$ is the DFT matrix. Due to its particular structure, matrix $\mathbf{M}$ satisfies $\mathbf{M}\mathbf{M}^H = \mathbf{I}$; this fact together with the matrix inversion lemma leads to

$$(\mathbf{B}^H\mathbf{B}+\mathbf{I})^{-1} = \mathbf{I}-\frac{1}{2}\mathbf{U}^H\mathbf{M}^H\mathbf{M}\mathbf{U} \tag{43}$$

where $\mathbf{M}^H\mathbf{M}$ is equal to an identity with some zeros in the diagonal. Consequently, the cost of line 4 of C-SALSA-1 and C-SALSA-2 is again that of the products by $\mathbf{U}$ and $\mathbf{U}^H$, i.e., $O(n\log n)$ using the FFT.



(a) (b)

Fig. 6. TV based image reconstruction (dynamic range $= 40$ dB). (a) Evolution of the objective function over time. (b) Evolution of the constraint $\|\mathbf{Bx} - \mathbf{y}\|$ over time.

(a) (b) (c)

Fig. 7. Image inpainting with TV regularization. (a) Original cameraman image. (b) Image with 40% pixels missing. (c) Estimated using C-SALSA.

TABLE VII
MRI RECONSTRUCTION—COMPARISON

Algorithm	Calls to $\mathbf{B}, \mathbf{B}^H$	Iterations	time (seconds)	MSE
NESTA	1228 (1161/1261)	307	15.50	9.335e-6
C-SALSA	366 (365/368)	122	12.89	2.440e-6

In the synthesis case, the observation matrix has the form $\mathbf{MUW}$. Clearly, the case is again similar to (42), but with $\mathbf{UW}$ and $\mathbf{W}^H\mathbf{U}^H$ instead of $\mathbf{W}$ and $\mathbf{W}^H$, respectively. Again, the cost of line 4 of C-SALSA-1 and C-SALSA-2 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 $\mathbf{W}^H$.

D. Computational Complexity

As shown in the previous section, the cost of line 4 of C-SALSA-1 and C-SALSA-2 is $O(n \log n)$. The other lines of the algorithms simply involve: 1) matrix-vector products involving $\mathbf{B}$, $\mathbf{W}$, $\mathbf{P}$, or their conjugate transposes, which have $O(n \log n)$ cost; 2) vector additions, which have $O(n)$ cost; and 3) the computation of the Moreau proximal maps (lines 5 and 6 of C-SALSA-1 and C-SALSA-2). In the case of the projections on a ball (line 6), it is clear from (28) that the cost is $O(n)$.

Finally, we consider the computational cost of the Moreau proximal map of the regularizer ϕ (line 5 of C-SALSA-1 and C-SALSA-2). In some cases, this map can be computed exactly in closed form; for example, if $\phi(\mathbf{x}) \equiv \|\mathbf{x}\|_1$, then $\mathbf{\Psi}_{\tau\phi}$ is simply a soft threshold and the cost is $O(n)$. In other cases, the Moreau proximal map does not have a closed form solution; for example, if $\phi(\mathbf{x}) \equiv \mathrm{TV}(\mathbf{x})$, the corresponding $\mathbf{\Psi}_{\tau\phi}$ has to be computed using one of several available iterative algorithms [14], [20]. Most of these iterative algorithms can be implemented with $O(n)$ cost, although with a factor that depends upon the number of iterations. In our implementation of C-SALSA we use Chambolle's algorithm [14].

In summary, for a wide choice of regularizers and frame representations, the C-SALSA algorithms have $O(n \log n)$ computational complexity.

IV. EXPERIMENTS

In this section, we report results of experiments aimed at comparing the speed of C-SALSA[6] with that of the current state of the art methods (that are freely available online): SPGL1[7][52], and NESTA[8] [6].

We consider three standard and often studied imaging inverse problems: image deconvolution (using both wavelet and

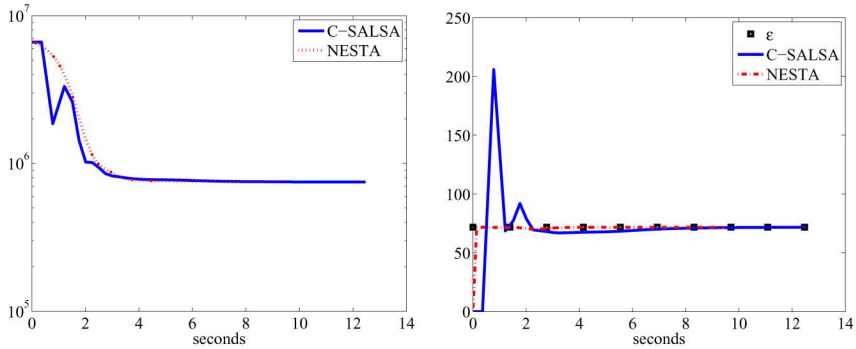
[6]C-SALSA is available at http://cascais.lx.it.pt/~mafonso/salsa.html.

[7]Available at http://www.cs.ubc.ca/labs/scl/spgl1.

[8]Available at http://www.acm.caltech.edu/~nesta.

TABLE VIII
IMAGE RECONSTRUCTION (HIGH DYNAMIC RANGE) USING TV—COMPUTATIONAL LOAD

Dyn. range	Avg. calls to $\mathbf{B}, \mathbf{B}^H$ (min/max)		Iterations		CPU time (seconds)		MSE	
(dB)	NESTA	C-SALSA	NESTA	C-SALSA	NESTA	C-SALSA	NESTA	C-SALSA
20	1213 (1169/1273)	226 (224/227)	303	76	8.99	7.24	0.00241743	0.000543426
40	991 (961/1017)	227 (224/227)	248	76	7.34	7.002	0.00432206	0.000651107
60	731 (721/737)	282 (281/284)	183	95	4.92	8.35	0.005294	0.00072848
80	617 (613/617)	353 (350/353)	154	118	4.16	10.72	0.00702862	0.000664638



(a) (b)

Fig. 8. Image inpainting with TV regularization. (a) Evolution of the objective function over time. (b) Evolution of the constraint $\|\mathbf{Bx} - \mathbf{y}\|$ over time.

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 [41].

All experiments were performed using MATLAB, on a Windows XP desktop computer with an Intel Pentium-IV 3.0 GHz processor and 1.5 GB of RAM. The number of calls to the operators $\mathbf{B}$ and $\mathbf{B}^H$, the number of iterations, CPU times, and MSE values presented are the averages values over 10 runs of each experiment. The number of calls reported for each experiment is the average over the 10 instances, with the minimum and maximum indicated in the parentheses. Since the stopping criteria of the implementations of the available algorithms differ, to compare the speed of the algorithms in a way that is as independent as possible from these criteria, the experimental protocol that we followed was the following: we first run one of the algorithms with its stopping criterion, and then run C-SALSA until the constraint in (1) is satisfied and the MSE of the estimate is below that obtained by the other algorithms.

The value of ε in (1) used in all cases was $\sqrt{m + 8\sqrt{m}}\sigma$, where m is the number of observations, and σ is the noise standard deviation. The parameter μ was hand-tuned for fastest convergence.

A. Image Deconvolution With Wavelets

We consider five benchmark cyclic deblurring problems [30], summarized in Table I, all on the well-known Cameraman image. The regularizer is $\phi(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$, thus, $\boldsymbol{\Psi}_{\tau\phi}$ is an element-wise soft threshold. We compare C-SALSA against SPGL1 and NESTA in the synthesis case, and against only NESTA in the analysis case, since SPGL1 is hardwired with $\|\mathbf{x}\|_1$ as the regularizer, and not $\|\mathbf{Px}\|_1$. Since the restored images are visually indistinguishable from those obtained in [30], and the SNR improvements are also very similar, we simply compare the speed of the algorithms, that is, the number of calls to the operators $\mathbf{B}$ and $\mathbf{B}^H$, the number of iterations, and the computation time.

In the first set of experiments, $\mathbf{W}$ is a redundant Haar wavelet frame with four levels. For the synthesis case, the CPU times taken by each of the algorithms are presented in Table II. Table III presents the corresponding results for the case with the analysis prior. In the second set of experiments, $\mathbf{W}$ is an orthogonal Haar wavelet basis; the results are reported in Table IV for the synthesis case, and in Table V for the analysis case. To visually illustrate the relative speed of the algorithms, Fig. 1 plots the evolution of the constraint $\|\mathbf{Bu}_k - \mathbf{y}\|$, versus time, in experiments 1, for the synthesis prior case, with redundant wavelets.

B. Image Deblurring With TV

The same five image deconvolution problems listed in Table I were also addressed using TV regularization (more specifically, the isotropic discrete total variation, as defined in [14]). The corresponding Moreau proximal mapping is computed using five iterations of Chambolle's algorithm [14].

Table VI compares the performance of C-SALSA and NESTA, in terms of speed. The evolutions of the objective function and the constraint for experiment 1 are plotted in Fig. 2.

We can conclude from Tables II–VI that, in image deconvolution problems, both with wavelet-based and TV-based regular-

TABLE IX
IMAGE INPAINTING: COMPARISON

	Calls to $\mathbf{B}, \mathbf{B}^H$	Iterations	time (seconds)	MSE
NESTA	403 (401/405)	101	10.29	81.316
C-SALSA	143 (143/143)	47	12.97	75.003

ization, C-SALSA is almost always clearly faster than the fastest of the other competing algorithms.

C. MRI Image Reconstruction

We now consider the problem of reconstructing the 128×128 Shepp–Logan phantom [shown in Fig. 3(a)] from a limited number of radial lines [22, in our experiments, as shown in Fig. 3(b)] of its 2-D discrete Fourier transform. The projections are also corrupted with circular complex Gaussian noise, with variance $\sigma^2 = 0.5 \times 10^{-6}$. We use TV regularization (as described in Section IV-B), with the corresponding Moreau proximal mapping implemented by 10 iterations of Chambolle's algorithm [14].

Table VII shows the number of calls, number of iterations, and CPU times, while Fig. 4 plots the evolution of the objective function and constraint over time. Fig. 3(c) shows the estimate obtained using C-SALSA (the estimate NESTA is, naturally, visually indistinguishable). Again, we may conclude that C-SALSA is faster than NESTA, while achieving comparable values of mean squared error of the reconstructed image.

1) High Dynamic Range TV Reconstruction: A related example that we will consider here is the reconstruction of images composed of random squares, from their partial Fourier measurements, with TV regularization (see [6, Sec. 6:4]). The dynamic range of the signals (the amplitude of the squares) varies from 20 dB to 80 dB. The size of each image is 128×128, the number of radial lines in the DFT measurement mask is 27 (corresponding to $m/n \approx 0.2$), and the Gaussian noise standard deviation is $\sigma = 0.1$.

Fig. 4(c) shows the original image with a dynamic range of 40 dB and the estimate obtained using C-SALSA. Fig. 6 shows the evolution over time of the objective and the error constraint for C-SALSA and NESTA, while Table VIII compares the two algorithms with respect to the number of calls to $\mathbf{A}$ and $\mathbf{A}^H$, number of iterations, CPU time, and MSE obtained, over 10 random trials. It is clear from Table VIII that C-SALSA uses considerably fewer calls to the operators $\mathbf{A}$ and $\mathbf{A}^H$ than NESTA.

D. Image Inpainting

Finally, we consider an image inpainting problem, as explained in Section III-C. The original image is again the Cameraman, and the observation consists in losing 40% of its pixels, as shown in Fig. 7. The observations are also corrupted with Gaussian noise (with an SNR of 40 dB). The regularizer is again TV, implemented by 10 iterations of Chambolle's algorithm.

The image estimate obtained by C-SALSA is shown in Fig. 7, with the original also shown for comparison. The estimate obtained using NESTA was visually very similar. Table IX compares the performance of the two algorithms, and Fig. 8 shows the evolution of the objective function for each of them.

V. CONCLUSION

We have presented a new algorithm for solving the constrained optimization formulation of regularized image reconstruction/restoration. The approach, which can be used with any type of convex regularizers (wavelet-based, TV), is based upon a VS technique which yields an equivalent constrained problem. This constrained problem is then addressed using an ALM, more specifically, the ADMM. Our algorithm works for any convex regularizer for which the Moreau proximal mapping can be efficiently computed, and is therefore more general purpose than some of the available state-of-the-art methods which are available only for either ℓ_1- and/or TV regularization. Experiments on a set of standard image recovery problems (deconvolution, MRI reconstruction, inpainting) have shown that the proposed algorithm (termed C-SALSA, for constrained split augmented Lagrangian shrinkage algorithm) is usually faster than previous state-of-the-art methods. Automating the choice of the value of the parameter μ remains an open question.

REFERENCES

[1] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "A fast algorithm for the constrained formulation of compressive image reconstruction and other linear inverse problems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Dallas, TX, 2010, pp. 4034–4037.

[2] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

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

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

[5] 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.

[6] S. Becker, J. Bobin, and E. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM J. Imag. Anal.*, vol. 4, pp. 1–39, 2011.

[7] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Bristol, U.K.: IOP Publishing, 1998.

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

[9] 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.

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

[11] J.-F. Cai, S. Osher, and Z. Shen, "Split Bregman methods and frame based image restoration," *SIAM J. Multiscale Model. Sim.*, vol. 8, pp. 337–369, 2009.

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

[13] 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.

[14] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, no. 1–2, pp. 89–97, 2004.
[15] 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. F. Faugeras, Eds. New York: Springer-Verlag, 2005.
[16] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, 1998.
[17] 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.
[18] P. Combettes and V. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM J. Multiscale Model. Sim.*, vol. 4, pp. 1168–1200, 2005.
[19] R. Courant, "Variational methods for the solution of problems with equilibrium and vibration," *Bull. Amer. Math. Soc.*, vol. 49, pp. 1–23, 1943.
[20] J. Dahl, P. Hansen, T. Jensen, and S. Jensen, "Algorithms and software for total variation image reconstruction via first-order methods," *Numer. Alg.*, vol. 53, pp. 67–92, 2010.
[21] I. Daubechies, M. De Friese, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, pp. 1413–1457, 2004.
[22] M. Domínguez-Jiménez and P. J. S. G. Ferreira, "A new preconditioner for Toeplitz matrices," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 758–761, Sep. 2009.
[23] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[24] 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.
[25] M. Elad, B. Matalon, and M. Zibulevsky, "Image denoising with shrinkage and redundant representations," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, New York, 2006, pp. 1924–1931.
[26] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Probl.*, vol. 23, pp. 947–968, 2007.
[27] 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.
[28] M. Figueiredo, J. Bioucas-Dias, and M. Afonso, "Fast frame-based image deconvolution using variable splitting and constrained optimization," in *Proc. IEEE Workshop Statist. Signal Process.*, Cardiff, U.K., 2009, pp. 109–112.
[29] M. Figueiredo and J. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3133–3145, Dec. 2010.
[30] 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.
[31] 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.
[32] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations," *Comput. Math. Appl.*, vol. 2, pp. 17–40, 1976.
[33] R. Glowinski and A. Marroco, "Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité d'une classe de problemes de dirichlet non lineares," *Rev. Française d'Automat. Inf. Recherche Opérationelle*, vol. 9, pp. 41–76, 1975.
[34] T. Goldstein and S. Osher, "The split Bregman algorithm for L_1 regularized problems," Comput. Appl. Math., Univ. California, Los Angeles, Tech. Rep. 08–29, 2008.
[35] E. Esser, "Applications of Lagrangian-based alternating direction methods and connections to split-Bregman," Comput. Appl. Math., Univ. California, Los Angeles, Tech. Rep. 09–31, 2009.
[36] T. Hale, W. Yin, and Y. Zhang, "A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing," Dept. Comput. Appl. Math., Rice Univ., Houston, TX, Tech. Rep 07–07, 2007.
[37] M. Hestenes, "Multiplier and gradient methods," *J. Optim. Theory Appl.*, vol. 4, pp. 303–320, 1969.
[38] S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinvesky, "An interior-point method for large-scale ℓ_1-regularized least squares," *IEEE J. Sel. Top. Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.
[39] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *J. Appl. Comput. Harmon. Anal.*, vol. 10, pp. 234–253, 2001.
[40] 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, pp. 10–12, 1996.
[41] M. Lustig, D. Donoho, and J. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Resonance Med.*, vol. 58, pp. 1182–1195, 2007.
[42] S. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic, 2009.
[43] 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.
[44] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, ser. A, vol. 103, pp. 127–152, 2005.
[45] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer-Verlag, 2006.
[46] M. Powell, "A method for nonlinear constraints in minimization problems," in *Optimization*, R. Fletcher, Ed. New York: Academic, 1969, pp. 283–298.
[47] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton Univ. Press, 1970.
[48] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.
[49] I. Selesnick, "Hilbert transform pairs of wavelet bases," *IEEE Signal Process. Lett.*, vol. 8, no. 6, pp. 170–173, Jun. 2001.
[50] I. Selesnick and M. Figueiredo, "Signal restoration with overcomplete wavelet transforms: Comparison of analysis and synthesis priors," in *Proc. SPIE*, 2009, vol. 7446.
[51] S. Setzer, "Split Bregman algorithm, Douglas-Rachford splitting, and frame shrinkage," in *Proc. 2nd Int. Conf. Scale Space Methods Variational Methods Comput. Vis.*, 2009, pp. 464–476.
[52] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, pp. 890–912, 2008.
[53] 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, no. 3, pp. 248–272, 2008.
[54] 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.
[55] J. Yang and Y. Zhang, Alternating Direction Algorithms for L1-Problems in Compressive Sensing CAAM, Rice Univ., Houston, TX, Tech. Rep. 09–37, 2009.
[56] J. Yang, Y. Zhang, and W. Yin, A Fast Alternating Direction Method for TVL1-L2 Signal Reconstruction From Partial Fourier Data CAAM, Rice Univ., Houston, TX, Tech. Rep. 08–27, 2008.
[57] 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 (S'10) 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 was 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.