A Fast Alternating Direction Method for TVL1-L2 Signal Reconstruction From Partial Fourier Data

Junfeng Yang, Yin Zhang, and Wotao Yin

***Abstract*—Recent compressive sensing results show that it is possible to accurately reconstruct certain compressible signals from relatively few linear measurements via solving nonsmooth convex optimization problems. In this paper, we propose the use of the alternating direction method—a classic approach for optimization problems with separable variables (D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations," *Computer and Mathematics with Applications*, vol. 2, pp. 17–40, 1976; R. Glowinski and A. Marrocco, "Sur lapproximation par elements finis dordre un, et la resolution par penalisation-dualite dune classe de problemes de Dirichlet nonlineaires," *Rev. Francaise dAut. Inf. Rech. Oper.*, vol. R-2, pp. 41–76, 1975)—for signal reconstruction from partial Fourier (i.e., incomplete frequency) measurements. Signals are reconstructed as minimizers of the sum of three terms corresponding to total variation, ℓ_1-norm of a certain transform, and least squares data fitting. Our algorithm, called RecPF and published online, runs very fast (typically in a few seconds on a laptop) because it requires a small number of iterations, each involving simple shrinkages and two fast Fourier transforms (or alternatively discrete cosine transforms when measurements are in the corresponding domain). RecPF was compared with two state-of-the-art algorithms on recovering magnetic resonance images, and the results show that it is highly efficient, stable, and robust.**

***Index Terms*—Compressive sensing (CS), compressed sensing, alternating direction method, magnetic resonance imaging (MRI), MRI reconstruction, fast Fourier transform (FFT), discrete cosine transform (DCT), total variation.**

I. Introduction

Let $\bar{u} \in \mathbb{R}^N$ be an unknown signal. Following the standard treatment, we vectorize two-dimensional images or higher dimensional data into one-dimensional vectors. In most cases, the number of salient features hidden in a signal is much fewer than its resolution, which means that $\bar{u}$ is usually sparse or compressible under a suitable basis. Let $\Psi = [\psi_1, \psi_2, \ldots, \psi_N] \in \mathbb{C}^{N\times N}$ be an orthonormal basis of $\mathbb{C}^N$. Then there exists an unique $\bar{x} \in \mathbb{C}^N$ such that

$$\bar{u} = \sum_{i=1}^{N} \psi_i \bar{x}_i = \Psi \bar{x}. \tag{1}$$

We say that $\bar{u}$ is K-sparse under Ψ if $\|\bar{x}\|_0$, the number of nonzeros in $\bar{x}$, is no more than K, and that $\bar{u}$ is compressible if $\bar{x}$ has only a few large (in magnitude) components. The case of interest is when $K \ll N$ or $\bar{u}$ is highly compressible.

Traditional data acquisition and reconstruction from frequency data follow the basic principle of the Nyquist density sampling theory, which states that the sample rate for faithful reconstruction is at least two times of the frequency bandwidth. In many applications, such as digital images and video cameras, the Nyquist sampling rate is so high that signal compression becomes necessary prior to storage and transmission. For example, in transform coding only the K (usually $K \ll N$) dominant components of $\bar{x}$ determined by (1) are saved while the rest are computed and then thrown away. The idea of compressive sensing (CS) goes against conventional wisdoms in data acquisition. In CS, a sparse or compressible signal is reconstructed from a small number of its projections onto a certain subspace. Let M be an integer satisfying $K < M \ll N$ and $\Phi \in \mathbb{C}^{M\times N}$ be a general nonadaptive sensing matrix. Instead of acquiring $\bar{u}$, one first obtains

$$b = \Phi\bar{u} = A\bar{x} \in \mathbb{C}^M, \quad A = \Phi\Psi \tag{2}$$

and then reconstructs $\bar{x}$ (and thus $\bar{u}$ by (1)) from the much shorter projection vector b via some reconstruction algorithms. Here, nonadaptiveness means that Φ does not depend on $\bar{u}$. The basic CS theory [10], [11], [20] justifies that it is highly probable to reconstruct $\bar{x}$ accurately from b as long as $\bar{x}$ is compressible (or even exactly when $\bar{x}$ is sparse and b is noiseless) and A possesses certain nice attributes. To make CS practical, one needs to design a good sensing matrix A (encoder), which ensures that b morally contains at least as much information as $\bar{x}$ does, and an efficient reconstruction algorithm (decoder), which can recover $\bar{x}$ from b.

A. Encoders and Decoders

For encoders, recent results indicate that stable reconstruction for both K-sparse and compressible signals can be ensured by the *restricted isometry property* (RIP) [13], [11]. It has become clear that for a sparse or compressible $\bar{x}$ to be reconstructed from b, it is sufficient if A satisfies the RIP of certain degrees. While verifying the RIP is a difficult task, the authors of [11], [20] showed that this property holds with high probability for Gaussian random matrices, e.g., matrix with independent and

Manuscript received January 28, 2009; revised November 19, 2009. Current version published March 17, 2010. The work of J.-F. Yang was supported by the Chinese Scholarship Council during his visit to Rice University. The work of Y. Zhang was supported in part by the National Science Foundation (NSF) under Grant DMS-0811188 and the Office of Naval Research (ONR) under Grant N00014-08-1-1101. The work of W. Yin was supported in part by the NSF CAREER Award DMS-0748839, ONR Grant N00014-08-1-1101, Air Force Office of Scientific Research Grant FA9550-09-C-0121, and an Alfred P. Sloan Research Fellowship. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mario Figueiredo.

J.-F. Yang is with the Department of Mathematics, Nanjing University, Nanjing, Jiangsu 210093, China (e-mail: jfyang@nju.edu.cn).

Y. Zhang and W. Yin are with the Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005 USA (e-mail: yin.zhang@rice.edu; wotao.yin@rice.edu).

Digital Object Identifier 10.1109/JSTSP.2010.2042333

identically distributed (i.i.d.) Gaussian entries. For orthonormal Ψ, moreover, $A = \Phi\Psi$ will have the desired RIP attribute if Φ is an i.i.d. Gaussian matrix. For other distributions which lead to the RIP, see, e.g., [2]. It is implied from the results in [47] that for "almost all" random sensing matrices, including the class of sub-Gaussian matrices, the probability of getting $\bar{x}$ from b is asymptotically at least $1 - \exp(-c_0(N - M))$ provided that $M > c_1 K \log(N/K)$, where $c_0, c_1 > 0$ are absolute constants. Aside from random sensing matrices, exact reconstruction is also attainable when A is a random partial orthonormal matrix [12] and particularly random partial Fourier matrix [11], which has important applications in magnetic resonance imaging (MRI) and is the focus of this paper. The analysis of such cases relies on the incoherence between the sensing system and the sparsifying basis of the underlying signal, see [12] and [8]. We point out that these theoretical analyses apply to merely random but not regular sampling sometimes necessary in practical applications such as MRI due to hardware constraints.

Being underdetermined, (2) usually have infinitely many solutions. Given that b is acquired from a highly sparse signal, a reasonable approach would be to seek the sparsest one among all the solutions of (2), i.e.,

$$\min_x \{\|x\|_0 : Ax = b\}. \tag{3}$$

Decoder (3) is able to recover a K-sparse signal exactly with overwhelming probability using only $K+1$ i.i.d. Gaussian measurements [3]. Unfortunately, directly solving this ℓ_0 problem is generally impractical in computation. A common substitute for (3) is the well-known basis pursuit problem [17]

$$\min_x \{\|x\|_1 : Ax = b\}. \tag{4}$$

It has been shown that, under some desirable conditions, with high probability problems (3) and (4) share common solutions (see, for example, [21]). For the ℓ_1 decoder (4), the number of measurements sufficient for exact reconstruction of a K-sparse signal is $O(K \log(N/K))$ when A is i.i.d. Gaussian [14] and $O(K \log N)$ when A is a random partial Fourier matrix (as in MRI) [11], both of which are, though larger than K, much smaller than N. Moreover, (4) is easily transformed to a linear program (when x is real) and thus can be solved efficiently at least in theory. Therefore, decoder (4) is both sparsity promoting and computationally tractable, establishing the theoretical soundness of the decoder. When $\bar{x}$ is compressible but not sparse, or when measurements are contaminated with noise, an appropriate relaxation to $Ax = b$ is desirable. For example, an appropriate relaxation under Gaussian noise is given by

$$\min_x \{\|x\|_1 : \|Ax - b\|_2 \leq \sigma\} \tag{5}$$

where $\sigma > 0$ is related to the noise level. There exist stability results saying that the ℓ_2 distance between $\bar{x}$ and the solution of (5) is no more than $O(\sigma + K^{-1/2}\|\bar{x} - \bar{x}(K)\|_1)$, where $\bar{x}(K)$ keeps the K dominant components in $\bar{x}$ and zero filling the rest; see [10] for example. A related problem to (5) is

$$\min_x \|x\|_1 + \mu\|Ax - b\|_2^2 \tag{6}$$

where $\mu > 0$. From the optimization theory, problems (5) and (6) are equivalent in the sense that solving one of the two will determine the parameter in the other such that both give the same solution. Aside from ℓ_1 related decoders, there exist other reconstruction techniques including the class of greedy algorithms; see [39] for example.

B. Compressive Imaging via a TVL1-L2 Model

Hereafter, we assume that $\bar{u}$ is a two-dimensional grayscale digital image with N pixels, and its partial frequency observation is given by

$$f_p = P\mathcal{T}\bar{u} + \omega \tag{7}$$

where $\mathcal{T} \in \mathbb{C}^{N\times N}$ represents a specific transform matrix, $P \in \mathbb{R}^{p\times N}$ is a selection matrix containing p rows of the identity matrix of order N, and $\omega \in \mathbb{C}^p$ represents random noise. In CS, $P\mathcal{T}$ serves as a sensing matrix. Model (7) characterizes the nature of a number of data acquisition systems. In the application of MRI reconstruction, data collected by an MR scanner are, roughly speaking, in the frequency domain (called k-space) rather than the spatial domain. Traditionally, MRI acquisition includes two key stages: k-space data acquisition and analysis. During the first stage, energy from a radio frequency pulse is directed to a small section of the targeted anatomy at a time. As a result, the protons within that area are forced to spin at a certain frequency and get aligned to the direction of the magnet. Upon stopping the radio frequency, the physical system returns to its normal state and releases energy that is then recorded for analysis. This recorded data consists of one or more entries of $P\mathcal{T}\bar{u}$. This process is repeated until enough data is collected for reconstructing a high-quality image in the second stage. For more details about how the MRI system works as related to CS, see [34] and references therein. Unfortunately, this data acquisition process is quite time consuming due to physiological and hardware constraints, so patients must endure long scanning sessions while their bodies are restrained in order to reduce motion artifacts. All these facts hint at the importance of reducing the scan time, which means collecting less data, without sacrificing the image quality.

In the rest of this paper, we will concentrate on the case of partial Fourier data, i.e., in (7) $\mathcal{T} = \mathcal{F}$ being a two-dimensional discrete Fourier transform matrix. We will propose and study a new algorithm for reconstructing an image $\bar{u}$ from a subset of its Fourier coefficients, though our results will equally apply to other partial spectral data, such as DCT, under proper boundary conditions.

We consider reconstructing $\bar{u}$ from f_p via the CS methodology. Let $\mathcal{F}_p = P\mathcal{F}$ and

$$\theta(u, f_p) = (1/2) \cdot \|\mathcal{F}_p u - f_p\|_2^2.$$

In our approach, $\bar{u}$ is reconstructed as a solution of the following TVL1-L2 model:

$$\min_u \sum_i \|D_i u\|_2 + \tau\|\Psi^\top u\|_1 + \mu\theta(u, f_p) \tag{8}$$

where $\sum_i$ is taken over all pixels, $\sum_i \|D_i u\|_2$ is a discretization of the total variation (TV) of u, $\|\Psi^\top u\|_1$ is the ℓ_1-norm of the

representation of u under Ψ, and $\tau, \mu > 0$ are scalars which are used to balance regularization and data fidelity. Since MR images commonly possess a blocky structure, the use of TV in regularization exploits image sparsity and preserves edges. In addition, it is known that MR images usually have sparse representations under certain wavelet bases [34]. Therefore, we will choose the sparsity promoting basis Ψ as a wavelet and particularly the Haar wavelet basis in our experiments.

Inverse imaging problems with TV and ℓ_1 regularization have been considered in [34] for MRI reconstruction. Specifically, model (8) has been used in [9], [32], and [35] and reported to reconstruct high quality MR images from a small number of Fourier coefficients [35]. An algorithm for inverse problems with compound regularization (e.g., (8)) was recently proposed in [7], and an alternating Bregman iterative method [36], [46] was recently applied to these problems in [1]. In addition, variational image restoration with an ℓ_1-like regularizer was studied in [4], which also suggested decreasing data fidelity by fixed-step gradient descents followed by solving denoising problems. We note that the algorithm in [4] reduces to the well known iterative shrinkage/thresholding (IST) method (e.g., [26], [19]) when the regularization is TV or the ℓ_1-norm of wavelet coefficients, and thus has practical performance similar to those of the IST method.

The main contribution of this paper is a very simple and fast algorithm, called RecPF, based on the general optimization framework of Glowinski and Marocco [28] and Gabay and Mercier [27] for solving model (8). In addition, we compare our algorithm to the recent two-step IST algorithm proposed in [6] and an operator splitting approach proposed in [35].

Two versions of RecPF have been developed for solving the same model (8), and they are based on different algorithms. The first version RecPF_v1 implements the algorithm in our previous paper [45], which is based on variable splitting, quadratic penalty, as well as parameter continuation. The second version RecPF_v2 was finished afterward, and it has improved performance thanks to the use of the classic augmented Lagrangian and an alternating direction technique. Both versions have been put online for download.[1] This paper mainly focuses on the algorithm and performance of RecPF_v2, but the algorithm of RecPF_v1 and the differences between the two algorithms are summarized in Section II-C.

C. Notation

Let the superscript $\top$ denote the transpose (conjugate transpose) operator for real (complex) matrices or vectors. For vectors v_i and matrices A_i, $i = 1, 2$, we let $(v_1; v_2) = (v_1^\top, v_2^\top)^\top$ and $(A_1; A_2) = (A_1^\top, A_2^\top)^\top$. For any i, D_i in (8) is a 2-by-N matrix such that the two entries of $D_i u$ represent the horizontal and vertical local finite differences of u at pixel i, whereas D_i near the boundary are defined to be compatible with $\mathcal{T}$ (more information will be given in Section II). The horizontal and vertical global finite difference matrices are denoted by $D^{(1)}$ and $D^{(2)}$, respectively. As such, $D^{(1)}$ and $D^{(2)}$ contains, respectively, the first and second rows of D_i for all i. In the rest of this paper, we let $\|\cdot\| = \|\cdot\|_2$. Additional notation is defined where it occurs.

[1] http://www.caam.rice.edu/optimization/L1/RecPF/

D. Organization

The paper is organized as follows. In Section II, we present the basic algorithm of RecPF_v2, explain its relationship to RecPF_v1, and discuss its connections to some recent work in the field of signal and image processing. A convergence result is also given without proof. Section III reports the results of our numerical experiments in which RecPF_v2 was compared to TwIST [6] and an operator splitting based algorithm [35]. Finally, some concluding remarks are given in Section IV.

II. Basic Algorithm and Related Work

The main difficulty in solving (8) comes from the nondifferentiability of its first and second terms. Our approach is to reformulate it as a linearly constrained problem and minimize its augmented Lagrangian. Instead of using the classic augmented Lagrangian method (ALM), which solves each unconstrained subproblem almost exactly, we propose the use of the cheaper alternating direction method (ADM).

This section is organized as follows. In Section II-A, we reformulate (8) as a constrained problem and describe the ALM. In Section II-B, we describe the use of ADM. Finally, in Section II-C we discuss the connections of this ADM to the previous RecPF_v1, as well as some recent work in signal and image processing.

A. Problem Reformulation and the ALM

By introducing auxiliary variables $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_N]$, where each $\mathbf{w}_i \in \mathbb{R}^2$, and $z \in \mathbb{R}^N$, problem (8) is equivalently transformed to

$$\min_{\mathbf{w}, z, u} \sum_i \|\mathbf{w}_i\| + \tau\|z\|_1 + \mu\theta(u, f_p) \quad \text{s.t. } \mathbf{w}_i = D_i u, \quad \forall i; \quad z = \Psi^\top u. \tag{9}$$

We point out that the splitting technique above is different from the one used in [7], which introduces a variable vector v and constraints $u = v$, minimizes $\sum_i \|D_i v\|$, applies quadratic penalty $\|u - v\|^2$ to penalize the violation of $u = v$. Despite the small difference between the two splitting strategies, ours enables fast solutions that take advantage of the fast Fourier transform (FFT) (which becomes clear in Section II-B next).

To tackle the linear constraints, we consider the augmented Lagrangian function of (9). For convenience, we introduce some notation. Given $\beta_1, \beta_2 > 0$, and for $s, t, \nu \in \mathbb{R}$, $\mathbf{s}, \mathbf{t}, \vec{\nu} \in \mathbb{R}^2$ we let

$$\phi_1(s, t, \nu) = \tau|s| - \nu(s - t) + (\beta_1/2) \cdot |s - t|^2$$

and

$$\phi_2(\mathbf{s}, \mathbf{t}, \vec{\nu}) = \|\mathbf{s}\| - \vec{\nu}^\top(\mathbf{s} - \mathbf{t}) + (\beta_2/2) \cdot \|\mathbf{s} - \mathbf{t}\|^2.$$

The augmented Lagrangian function of (9) is given by

$$\mathcal{L}_A(\mathbf{w}, z, u, \lambda_1, \lambda_2) = \sum_i \phi_2(\mathbf{w}_i, D_i u, (\lambda_2)_i) + \sum_i \phi_1\left(z_i, \psi_i^\top u, (\lambda_1)_i\right) + \mu\theta(u, f_p) \tag{10}$$

where, for each i, $(\lambda_1)_i \in \mathbb{R}$, $(\lambda_2)_i \in \mathbb{R}^2$, and ψ_i is the ith column of Ψ. For simplicity, we assume $\beta_1 = \beta_2 \equiv \beta$ without loss of generality because, otherwise, letting $\tau \leftarrow \tau\sqrt{\beta_2/\beta_1}$, $\lambda_1 \leftarrow \sqrt{\beta_2/\beta_1}\lambda_1$ and $(\Psi, z) \leftarrow \sqrt{\beta_1/\beta_2}(\Psi, z)$ equalizes the penalty parameters.

Given $(\mathbf{w}, z, u)^k \triangleq (\mathbf{w}^k, z^k, u^k)$, $(\lambda_1)^k$ and $(\lambda_2)^k$, the ALM for (9) is an iterative algorithm based on the iteration

$$\begin{cases} (\mathbf{w}, z, u)^{k+1} \leftarrow \arg\min \mathcal{L}_A\left(\mathbf{w}, z, u, (\lambda_1)^k, (\lambda_2)^k\right) \\ (\lambda_1)_i^{k+1} \leftarrow (\lambda_1)_i^k - \beta(z_i^{k+1} - \psi_i^\top u^{k+1}), \quad \forall i, \\ (\lambda_2)_i^{k+1} \leftarrow (\lambda_2)_i^k - \beta(\mathbf{w}_i^{k+1} - D_i u^{k+1}), \quad \forall i. \end{cases} \tag{11}$$

To guarantee convergence of the ALM, each minimization subproblem needs to be solved to certain high accuracy before the iterative updates of multipliers. In our case, the joint minimization with respect to all $\mathbf{w}$, z and u in (10) is not always numerically efficient. Therefore, each iteration of the ALM is relatively expensive. In contrast, the ADM approach described in the next subsection has a much cheaper per-iteration cost. In a nutshell, by utilizing the separable structure of the variables in (9), the ADM decreases $\mathcal{L}_A$ at each iteration by just one alternating minimization followed by immediate multiplier updates.

B. Solving the TVL1-L2 Model by the ADM

Although (10) has more decision variables than (8), it is easier to minimize (10) especially with respect to $\mathbf{w}$, z, and u each. First, for fixed u and λ (here and after $\lambda = (\lambda_1, \lambda_2)$), the minimization of $\mathcal{L}_A$ with respect to $\mathbf{w}$ and z can be carried out in parallel because all $\mathbf{w}_i$ and z_i are separated from one another.

For fixed u and λ, the minimizer z_i is given by

$$z_i = s_1\left(\psi_i^\top u + (\lambda_1)_i/\beta, \tau/\beta\right), \quad \forall i \tag{12}$$

where $s_1(\cdot, \tau/\beta)$, known as the one-dimensional shrinkage operator, is defined as

$$s_1(\xi, \tau/\beta) \triangleq \max\{|\xi| - \tau/\beta, 0\} \cdot \operatorname{sgn}(\xi), \quad \xi \in \mathbb{R} \tag{13}$$

and the minimizer $\mathbf{w}_i$ is given by

$$\mathbf{w}_i = s_2(D_i u + (\lambda_2)_i/\beta, 1/\beta), \quad \forall i \tag{14}$$

where $s_2(\cdot, 1/\beta)$, known as the two-dimensional shrinkage first introduced in [40], is defined as

$$s_2(\vec{\xi}, 1/\beta) \triangleq \max\{\|\vec{\xi}\| - 1/\beta, 0\} \cdot \vec{\xi}/\|\vec{\xi}\|, \quad \vec{\xi} \in \mathbb{R}^2 \tag{15}$$

where $0 \cdot (0/0) = 0$ is assumed. We note that the computational costs for both (12) and (14) are linear in N.

Second, for fixed $(\mathbf{w}, z)$ and λ, the minimization of $\mathcal{L}_A$ with respect to u becomes a least squares problem which is diagonalized by a 2-D discrete Fourier transform. To simplify representation, we let $w_j \triangleq (\mathbf{w}_1(j); \ldots; \mathbf{w}_N(j))$, $j = 1, 2$, $w \triangleq (w_1; w_2)$, and $D \triangleq (D^{(1)}; D^{(2)})$. With this notation, the minimization of $\mathcal{L}_A$ with respect to u, after variable reordering, can be rewritten as

$$\begin{aligned} &\min_u -(\lambda_2)^\top(w - Du) + (\beta/2) \cdot \|w - Du\|^2 \\ &-(\lambda_1)^\top(z - \Psi^\top u) + (\beta/2) \cdot \|z - \Psi^\top u\|^2 + \mu\theta(u, f_p) \end{aligned} \tag{16}$$

which, by noting the orthonormality of Ψ, is equivalent to the normal equations

$$Mu = y \tag{17}$$

where

$$M = D^\top D + I + (\mu/\beta) \cdot \mathcal{F}_p^\top \mathcal{F}_p$$

and

$$y = D^\top(w - \lambda_2/\beta) + \Psi(z - \lambda_1/\beta) + (\mu/\beta) \cdot \mathcal{F}_p^\top f_p.$$

Since $D^{(1)}$ and $D^{(2)}$ are finite difference matrices, under the periodic boundary conditions for u, they are block circulant matrices and can be diagonalized by the 2-D discrete Fourier transform $\mathcal{F}$. It is worth pointing out that if $\mathcal{T}$ is a discrete cosine transform, the same result holds under the symmetric boundary conditions. Let $\hat{D}^{(j)} = \mathcal{F} D^{(j)} \mathcal{F}^\top$, which is diagonal, $j = 1, 2$, and $\hat{D} = (\hat{D}^{(1)}; \hat{D}^{(2)})$. Multiplying by $\mathcal{F}$ on both sides of (17), we obtain

$$\hat{M}\mathcal{F}(u) = \hat{y} \tag{18}$$

where

$$\hat{M} = \hat{D}^\top \hat{D} + I + (\mu/\beta) \cdot P^\top P$$

is a diagonal matrix noting that $P^\top P$ is diagonal, and

$$\hat{y} = \mathcal{F}\left(D^\top w + \Psi z\right) + (\mu/\beta) \cdot P^\top f_p.$$

Therefore, solving (18) is straightforward, which means that (17) can be easily solved for given w and z as follows. Before iterations begin, compute $\hat{M}$. At each iteration, first compute $D^\top w$ and Ψz. Then, obtain $\hat{y}$ by applying an FFT. Finally, solve (18) using $\hat{M}$ to obtain $\mathcal{F}(u)$ and thus u after applying an inverse FFT to $\mathcal{F}(u)$.

We note that the orthonormality of Ψ is required. For non-orthonormal Ψ, the above procedure cannot be applied to minimize (16). Moreover, the above alternating technique is limited to the cases where $\mathcal{T}$ in (7) must correspond to an orthonormal matrix that can diagonalize the finite difference operators $D^{(1)}$ and $D^{(2)}$ under suitable boundary conditions (e.g., $\mathcal{T}$ can be a FFT (or DCT) matrix together with the periodic (or symmetric) boundary conditions imposed on u).

One can circularly applying (12), (14), and (17) until $\mathcal{L}_A$ is minimized jointly with respect to $(\mathbf{w}, z, u)$ and update the multipliers as in the ALM (11). However, we choose to update λ immediately after computing (12), (14) and (17) just once. This gives the ADM as follows.

Algorithm 1: Input problem data P, f_p and model parameters $\tau, \mu > 0$. Given $\beta > 0$ and $\gamma \in (0, (\sqrt{5}+1)/2)$. Initialize $u = u^0$, $\lambda_1 = (\lambda_1)^0$ and $\lambda_2 = (\lambda_2)^0$. Set $k = 0$.

While "not converged," **Do**

1) Compute z and $\mathbf{w}$ by

$$\begin{aligned} z_i^{k+1} &\leftarrow s_1\left(\psi_i^\top u^k + (\lambda_1)_i^k/\beta, \tau/\beta\right), \ \forall i, \\ \mathbf{w}_i^{k+1} &\leftarrow s_2\left(D_i u^k + (\lambda_2)_i^k/\beta, 1/\beta\right), \ \forall i; \end{aligned}$$

2) Compute u^{k+1} by solving (17), where y depends on $(w,z)=(w^{k+1},z^{k+1})$ and $\lambda=((\lambda_1)^k,(\lambda_2)^k)$.
3) Update λ_1 and λ_2 by

$$(\lambda_1)_i^{k+1} \leftarrow (\lambda_1)_i^k - \gamma\beta\left(z_i^{k+1} - \psi_i^\top u^{k+1}\right),\ \forall i,$$
$$(\lambda_2)_i^{k+1} \leftarrow (\lambda_2)_i^k - \gamma\beta\left(\mathbf{w}_i^{k+1} - D_i u^{k+1}\right),\ \forall i;$$

4) $k \leftarrow k+1$.

End Do

We simply terminated Algorithm 1 when the relative change becomes small enough, i.e.,

$$\|u^{k+1}-u^k\| \le \epsilon(1+\|u^k\|) \tag{19}$$

where $\epsilon>0$ is a tolerance.

The idea of ADM described above dates back to the work by Glowinski and Marocco [28] and Gabay and Mercier [27], who proposed the method to utilize the separable structure of variables. The ADM was extensively studied in optimization and variational analysis. For example, in [30] it is interpreted as the Douglas–Rachford splitting method [22] applied to a dual problem. The equivalence between the ADM and a proximal point method was shown in [23]. Moreover, the ADM has been extended to inexact minimization of subproblems; see [23] and [33].

In step 3) of Algorithm 1, a step length of $\gamma\in(0,(\sqrt{5}+1)/2)$ is permitted. Similarly, without invalidating the convergence of the ALM (11), the update to the multipliers can also be multiplied by $\gamma\in(0,2)$. The shrinkage in the permitted range from (0,2) to $(0,(\sqrt{5}+1)/2)$ is related to relaxing the exact minimization of $\mathcal{L}_A$ in the ALM to merely one round of alternating minimization in the ADM. To the best of our knowledge, the convergence of the ADM for $\gamma\in(0,(\sqrt{5}+1)/2)$ was first established in [29] in the context of variational inequality, which covers the proof of the follow theorem.

Theorem 2.1: For any $\beta>0$ and $\gamma\in(0,(\sqrt{5}+1)/2)$, the sequence $\{(w^k,z^k,u^k)\}$ generated by Algorithm 1 from any starting point (w^0,z^0,u^0) converges to a solution of (9).

C. Connections to Recent Work

The splitting technique for TV regularization shown in (9) for model (8) was first proposed in [40] and [41] without using the Lagrange multipliers, in the context of image deconvolution, where the authors used the classic quadratic penalty method to enforce the equality constraints and continuation technique to accelerate convergence. Similar ideas were applied to multichannel deconvolution in [43] and impulsive noise removal in [44].

Since the splitting/penalty approach utilizes shrinkage and FFT in the same way described in the previous subsection (with $\lambda=0$) and thus gives rise to robust and efficient algorithms, we applied the same idea to solving (8) in [45]. The resulting algorithm was called initially RecPF (reconstruction form partial Fourier data) and now RecPF_v1, which minimizes a function of the form

$$\sum_i(\phi_2(\mathbf{w}_i,D_iu,0)+\phi_1(z_i,\psi_i^\top u,0))+\mu\theta(u,f_p)$$

where ϕ_1 and ϕ_2 are as defined in Section II-A with $\beta_1,\beta_2\gg 0$. Continuation on β_1,β_2 significantly reduces the total number of iterations. Soon after submitting [45], we became aware that RecPF_v1 can be further improved by introducing the augmented Lagrangian (10) and applying the ADM. We implemented the ADM for (8), as well as the TV-based deconvolution problems in [41] and [44], and have publicized the MATLAB programs RecPF_v2 online. As expected, the ADM-based programs run faster. One can attribute the improvement in performance to the fact that the use of the augmented Lagrangian eliminates the need of excessively large values for the penalty parameters β_1,β_2, which cause ill-conditioning. Therefore, in Section III we merely present comparison results between RecPF_v2 and other algorithms. For practical performance of RecPF_v1 and its convergence properties, see [45].

By combining the splitting technique for TV [40] and the Bregman iterative algorithm [36], [46], a split Bregman method was proposed in [31] for a class of inverse problems, which is equivalent to the ALM. It was empirically observed that the split Bregman method converges well enough when only one alternating minimization is performed at each iteration in the same way as in the ADM. Lately, an alternating Bregman iterative method was proposed in [1] for the solution of inverse problems with compound regularizers, which is essentially an ADM. Also, a variable splitting and constrained optimization approach was recently applied to frame-based deconvolution in [25]. The split Bregman algorithm and its relationship with the Douglas–Rachford splitting was analyzed in [38]. Lately, [24] describes the relationships between the split Bregman method, ALM, and ADM, and [18] discusses a proximal decomposition method for convex variational inverse problems.

III. Experimental Results

A. General Description

In the remaining of this paper, RecPF refers to RecPF_v2. In this section, we present MR image reconstruction results of RecPF and two other recently proposed algorithms: a two-step iterative shrinkage/thresholding algorithm (TwIST) [6] and an operator splitting-based algorithm that we call OS [35], both of which have been regarded fast for solving inverse sparse reconstruction problems, see, e.g., [5], [16], and [42]. All experiments were performed under Windows Vista Premium and MATLAB v7.8 (R2009a) running on a Lenovo laptop with an Intel Core 2 Duo CPU at 1.8 GHz and 2 GB of memory. The finite difference and shrinkage operations were implemented in the C programming language, which was linked to MATLAB via the `mex` interface.

We generated our test sets from three images, the Shepp–Logan phantom and two real brain MR images. The CS measurement data f_p were generated according to (7). In each test, we obtained f_p by first rescaling the intensity values of the tested image to [0,1], then applying a partial FFT to the resulting image, and finally adding Gaussian noise to the partial FFT result. The partial FFT generated the samples in the Fourier domain along a number of radial lines through the center; for example, Fig. 1 shows 22 radial lines in a Fourier

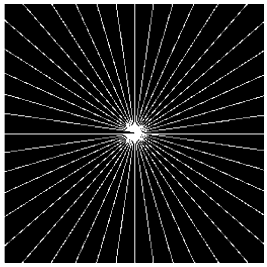


Fig. 1. Fourier domain sampling positions with 22 radial lines for test 1.

TABLE I
TEST IMAGES INFORMATION AND MODEL PARAMETER VALUES
(RLs = NUMBER OF RADIAL LINES)

Test	Image	Size	RLs	Sample Ratio	(μ, τ)
1	phantom	256×256	22	9.36%	(1e2∼1e9, 0)
2	brain-1	256×256	66	26.85%	(2e3, 1)
3	brain-2	512×512	88	18.59%	(2e3, 1)

TABLE II
RESULTS OF TWIST ON TEST 1

	TwIST (without continuation)				TwIST (with continuation)			
μ	Err	Obj	Iter	T	Err	Obj	Iter	T
1e2	5.5%	14.96	38	30	5.5%	14.96	38	31
1e3	5.1%	1.854	140	90	5.3%	1.869	45	46
1e4	6.5%	.2906	571	337	5.3%	.2904	74	59
1e5	14.7%	.1298	3167	1686	8.4%	.1290	53	43

TABLE III
RESULTS OF RECPF ON TEST 1

	Results of RecPF: $\mu = 10^p$								
p	Err	Obj	Iter	T	p	Err	Obj	Iter	T
2	7.8%	14.49	63	3.0	6	4.6%	1.7e-3	79	4.7
3	5.2%	1.665	74	3.7	7	4.7%	1.7e-4	79	4.8
4	6.0%	.1725	76	4.1	8	4.7%	1.7e-5	80	4.4
5	6.0%	.0173	76	3.8	9	4.6%	1.7e-6	80	4.5

domain. Different numbers of radial lines resulted in different sampling ratios. In all tests, we added Gaussian noise to both the real and the imaginary parts of Fourier coefficients. The additive noise had a mean zero and standard deviation 0.01.

In our experiments, we simply set $\beta_1 = \beta_2 = 10$ and $\gamma = 1.618$ for Algorithm 1, which, though may be suboptimal, are sufficient to demonstrate the efficiency of RecPF. The parameters used in TwIST and OS were set to be optimized values after our numerous trials, and they are described in the subsections below. We tried different starting points for the three algorithms and found that their performance was insensitive to starting points. Therefore, we simply set the starting image u to be zero. Table I summarizes the test data, as well as the values of the parameters μ and τ used in (8).

We note that both TwIST and OS can be applied to problem (8) with $\mathcal{F}_p$ being replaced by a general linear operator A as long as the matrix-vector multiplications Au and $A^\top v$ are easy to compute, but RecPF cannot because it needs (17) to be diagonalizable by fast transforms.

In addition, because both TwIST and OS scale (8) by $1/\mu$ (in their models, small parameters are put in front of the regularization terms), to keep compatibility we multiply the objective function by $1/\mu$ for RecPF. Hence, the objective values presented below for (μ, τ) have been scaled by $1/\mu$.

B. Comparison With TwIST

In test 1, we compared RecPF with TwIST on solving

$$\min_u \mu^{-1}\Phi_{\text{reg}}(u) + \frac{1}{2}\|Au - b\|^2$$

where $\Phi_{\text{reg}}(\cdot)$ can be either TV or ℓ_1 regularization and A is a linear operator. The iteration framework of TwIST is

$$u_{k+1} = (1-\alpha)u_{k-1} + (\alpha - \delta)u_k + \delta\Psi_\mu(\xi_k)$$

where $\alpha, \delta > 0$ are parameters, $\xi_k = u_k + A^\top(b - Au_k)$, and

$$\Psi_\mu(\xi_k) = \arg\min_u \mu^{-1}\Phi_{\text{reg}}(u) + \frac{1}{2}\|u - \xi_k\|^2. \qquad (20)$$

Its latest version, TwIST_v1, was not designed to solve problem (8) with both the TV and ℓ_1 regularization terms. Hence, in order to compare RecPF with TwIST on solving the same model, we dropped the ℓ_1-term in (8) by setting $\tau = 0$ for RecPF and letting $\Phi_{\text{reg}} = \sum_i \|D_i u\|$, $A = \mathcal{F}_p$ and $b = f_p$ for TwIST. TwIST_v1 solves the subproblem (20) using Chambolle's algorithm [15]. In contract, RecPF without the ℓ_1 term has a much cheaper per-iteration cost of two FFTs (including one inverse FFT).

First, we ran the monotonic variant of TwIST in TwIST_v1, which was terminated when the relative change in the objective function fell below $tol = 10^{-5}$. In TwIST_v1, the parameters α and δ were determined carefully based on the spectral distribution of $A^\top A$. In our case of $A = \mathcal{F}_p$, the minimum and maximum eigenvalues of $A^\top A$ were obviously 0 and 1, respectively. Therefore, we assigned a relatively small value 10^{-3} to the TwIST parameter `lam1` (which was used to compute α and δ), as recommended in the TwIST_v1's documentation. Furthermore, to speed up convergence, TwIST allowed maximally ten iterations as default for each call of Chambolle's algorithm to solve (20). Based on our experimental results, TwIST deteriorates as μ becomes large. Therefore, we also applied heuristic continuation on μ when it is large, i.e., initialize μ a smaller value and increase it gradually to the target one. Warm start technique was used. In our experiments, we applied continuation when $\mu \geq 10^3$, in which case we initialize it to be 10^2 and multiply it by ten in each continuation step. Similar continuation was also applied to tol correspondingly. The results of TwIST (with and without continuation) for μ between 10^2 to 10^5 are listed in Table II. For the same range of μ, we terminated RecPF when the obtained function value is no bigger than that of TwIST and the condition (19) is satisfied with $\epsilon = 10^{-4}$. We also tested μ values bigger than 10^5, in which case RecPF was terminated by condition (19) only. The results of RecPF are given in Table III. In both tables, the following quantities are listed: the error in the reconstructed image relative to the original image (Err), the final objective function value (Obj), the number of iterations (Iter), and the CPU time (T) in seconds.

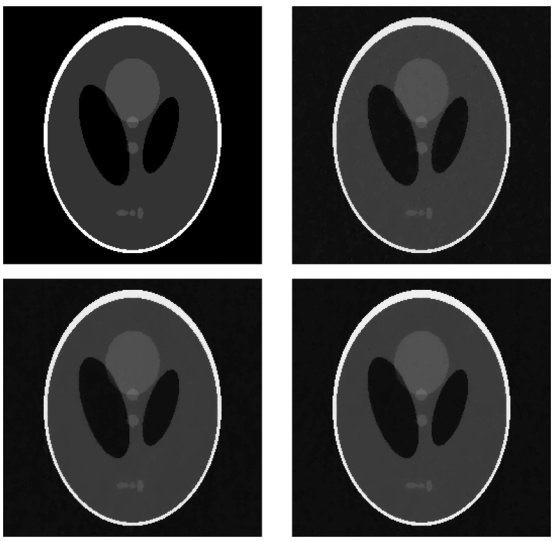


Fig. 2. Reconstructed images in test 1. Top left: original image. Top right: reconstruction by TwIST with $\mu = 10^3$ (Err $= 5.1\%$). Bottom left: reconstruction by RecPF with $\mu = 10^3$ (Err $= 5.2\%$). Bottom right: reconstructed by RecPF with $\mu = 10^9$ (Err $= 4.6\%$).

We point out that upon termination, RecPF returned smaller objective values than TwIST in all tests, though TwIST's solutions had smaller relative errors for $\mu = 10^2$ and 10^3. From Tables II and III, it can be seen that RecPF is much faster than TwIST in attaining a comparable accuracy, primarily because of the difference in their per-iteration costs: TwIST solves the TV problem (20) using Chambolle's algorithm (which takes its own iterations) but RecPF solves (17) at a smaller cost of two FFTs. Furthermore, the multiplier update lets RecPF to converge without a large β and thus makes it robust and efficient. It can be seen from Tables II and III that RecPF had stable iteration numbers and CPU times for different values of μ but TwIST was not as stable. The parameter $\mu = 10^3$ gave the reconstructed images in the best quality for both TwIST and RecPF when μ is between 10^2 and 10^5, which are depicted in Fig. 2.

We also tested the two algorithms on data containing less or no noise and observed similar relative performance. For example, on noiseless data, both algorithms (with the above described settings) were able to converge to solutions with relative errors less than 1% for $\mu = 10^3$, but RecPF was faster than TwIST. We note that, although it is possible to make TwIST faster by tuning some parameters (e.g., setting the maximum denoising steps allowed in Chambolle's algorithm to smaller than 10), TwIST_v1 is generally not as efficient as RecPF on problem (8).

In our experiments, we observed that TwIST_v1 required careful selections of parameters such as α, δ, μ and error tolerance tol in order to obtain results comparable to those of RecPF. In comparison, RecPF requires very little tuning. As mentioned, $\beta = 10$ and $\gamma = 1.618$ were used throughout our tests where only the error tolerance ϵ in (19) varied. Interestingly, the performance of RecPF appeared to be insensitive to the values of μ, and it converged well even with huge μ values. The results of RecPF with μ values as large as 10^9 are given in Table III, where the resulting relative errors are even a little better than those results with μ between 10^2 and 10^5. For example, the recovered image by RecPF with $\mu = 10^9$ has a relative error 4.6%, which is depicted at the bottom right corner in Fig. 2. We tested even larger μ values and obtained equally good images. This behavior can be explained by closely examining the linear system in (18).

Recall that P is a selection matrix. From the formulations of $\hat{M}$ and $\hat{y}$ (18) it becomes clear that 1) the value of μ only affects those Fourier coefficients in $\mathcal{F}(u)$ corresponding to the sample positions; and 2) as μ gets larger, the entries of $\mathcal{F}(u)$ corresponding to the sampled positions get closer to f_p. In the limit as $\mu \to \infty$, solving (18) simply fills $\mathcal{F}(u)$ with f_p at the sample positions and updates the remaining entries of $\mathcal{F}(u)$ independent of μ. This separation makes RecPF very stable with respect to large values of μ and allows it to faithfully executes TV regularization.

C. Comparison With OS

In tests 2 and 3, we compared RecPF with OS [35] on solving problem (8) with both TV and ℓ_1 terms. OS iterates the fix-point (21) below, in which $s \in \mathbb{R}^N$, $\mathbf{w}_i, \mathbf{t}_i \in \mathbb{R}^2$, $i = 1, \ldots, N$, are auxiliary variables and $\delta_1, \delta_2 > 0$ are constants

$$\begin{cases} s = \Psi^\top u - (\delta_1/\mu) \cdot \Psi^\top \left(\sum_i D_i^\top \mathbf{w}_i + \mu \nabla \theta(u, f_p) \right) \\ \mathbf{t}_i = \mathbf{w}_i + \delta_2 D_i u, \ \forall i \\ u = \Psi\{\max(|s| - \delta_1 \tau/\mu, 0) \circ \operatorname{sgn}(s)\} \\ \mathbf{w}_i = \min(1/\delta_2, \|\mathbf{t}_i\|) \cdot \mathbf{t}_i / \|\mathbf{t}_i\|, \ \forall i. \end{cases} \tag{21}$$

The authors showed that for any fixed $\delta_1, \delta_2 > 0$, u is a solution of (8) if and only if it satisfies (21). Given u^k and $\{\mathbf{w}_i^k, \ \forall i\}$, s^k and $\{\mathbf{t}_i^k, \ \forall i\}$ can be computed by the first two equations, and then be used to compute u^{k+1} and $\{\mathbf{w}_i^{k+1}, \ \forall i\}$ in the last two equations in (21). For δ_1 and δ_2 in certain ranges, such iterations converge. Similar to RecPF, every iteration of OS involves shrinkages, FFTs, and discrete wavelet transforms (DWTs). Furthermore, OS takes advantage of continuation, i.e., decreasing τ and $1/\mu$ from larger values to prescribed small ones. For a fixed pair of (τ, μ), the fixed-point iterations of OS terminate once one of the following two conditions is met:

$$f_k - f_{k+1} < \varepsilon_2 \sqrt{\tau_c/\tau_t} \max\{1, f_k\} \tag{22}$$

$$\|u_{k+1} - u_k\| < \varepsilon_1 \max\{1, \|u_k\|\} \tag{23}$$

where f_k is the objective value at u_k, τ_c, and τ_t are the current and the target values of τ, respectively, and $\varepsilon_1, \varepsilon_2 > 0$ are stopping tolerances.

In both tests, we set $\delta_1 = \delta_2 = 0.8$, $\varepsilon_1 = 10^{-4}$ and $\varepsilon_2 = 5 \times 10^{-4}$ for OS. For RecPF, we set $\epsilon = 10^{-3}$ in (19) which, although much looser than the previous tolerance used in test 1, was sufficient for RecPF to return better results than OS.

Since the two algorithms used different stopping criteria and the multiple tuning parameters of OS such as δ_1 and δ_2 affect the convergence speed, it was rather difficult to compare them at their best performance. Since OS implements the fixed-point iterations based on (21) that do not directly aim at decreasing the

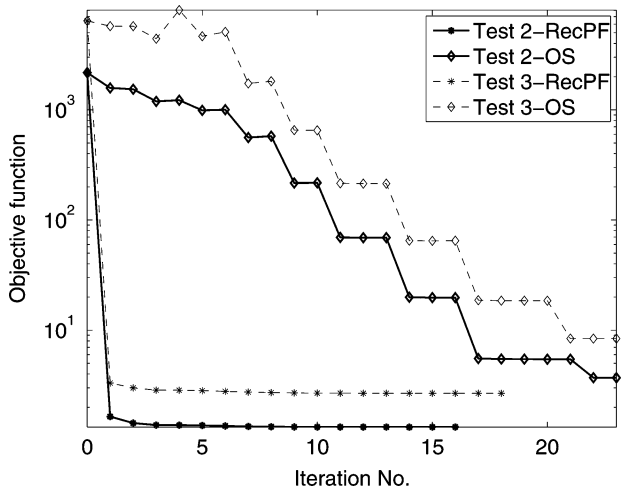


Fig. 3. Comparing RecPF with OS: objective value versus iteration numbers.vsk 6

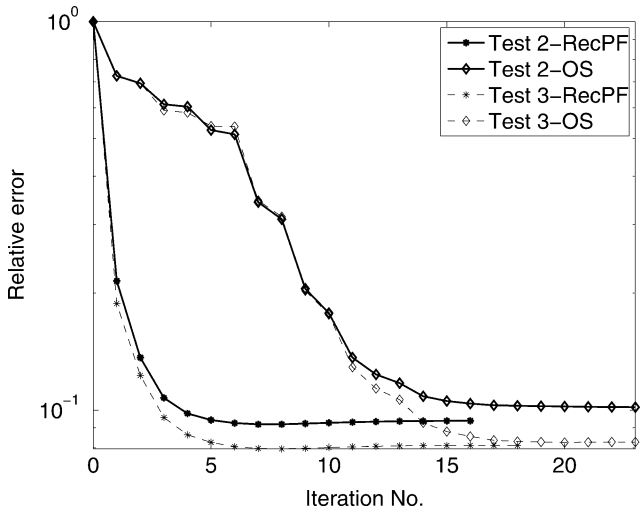


Fig. 4. Comparing RecPF with OS: relative error versus iteration numbers.

objective function, its objective values do not decrease monotonically for certain choices of δ. We tried different δ values and observed the following. For smaller δ's, OS tended to yield monotonically decreasing objectives but converged slowly. For larger δ's and looser stopping criteria, OS was faster but lost objective monotonicity and returned solutions with larger relative errors. After trying different combinations of ε's and δ's for OS, we determined to use the above parameter values, giving the best compromise between convergence speed and image quality, to generate the results presented next.

In both tests 2 and 3, the sparsity promoting basis Ψ was set to be the Haar wavelet transform using the Rice Wavelet Toolbox and its default setting [37]. The per-iteration computational cost of both methods is two FFTs (including one inverse FFT) and two DWTs (including one inverse DWT). Therefore, it is fair to present our numerical results in plots of objective value and relative error versus iteration numbers, which are given in Figs. 3 and 4, respectively. The reconstructed images are given in Fig. 5.

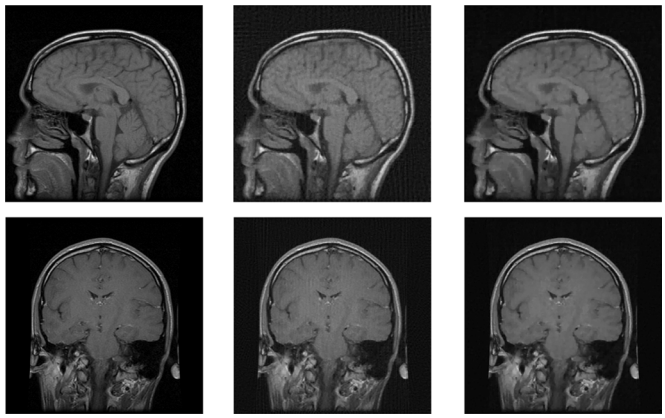


Fig. 5. Results of tests 2 and 3. Top row (results of test 2) from left to right: original, recovered by OS (Err: 10.18%) and RecPF (Err: 8.21%); Bottom row (results of test 3) from left to right: original, recovered by OS (Err: 8.28%) and RecPF (Err: 6.96%).

As can be seen from Figs. 3 and 4, RecPF converged much faster than OS in terms of both objective functions and relative errors. Moreover, in both tests RecPF achieved and maintained both lower objective values and relative errors throughout the entire iteration process. In both tests, RecPF took much fewer iterations than OS to attain the same level of relative error. In addition, images reconstructed by RecPF have higher qualities than those by OS as is evidenced by Fig. 5.

Our other experiments yielded consistent results. In general, when stricter tolerances are used, better results can be obtained from both algorithms at a cost of more iteration numbers. Independent of tolerances used, the ratio of their performances stays similar.

IV. Conclusion

Based on the classic augmented Lagrangian approach and a simple splitting technique, we proposed the use of alternating direction method for solving signal reconstruction problems with partial frequency data (DFT or DCT coefficients). Our algorithm minimizes the sum of a TV and/or ℓ_1-norm regularization terms together with a fidelity term. At each iteration, the main computation of RecPF only involves fast and stable operations consisting of shrinkages and FFTs (or DCTs).

Compared to the two efficient algorithms TwIST [6] and OS [35], RecPF is more efficient and robust for reconstructing large-scale signals or images. Furthermore, RecPF requires very little tuning of parameters, and it consistently performs well with a very large dynamic range of regularization/fidelity weight parameters. We hope that RecPF is useful in relevant areas of compressive sensing such as sparsity-based, rapid MR image reconstruction.

Acknowledgment

The authors would like to thank the three anonymous referees for their valuable suggestions and comments that have helped improve the paper.

REFERENCES

[1] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Image restoration with compound regularization using a Bregman iterative algorithm," in *Proc. Workshop Signal Process. With Adaptive Sparse Structured Representations SPARS'09*, Saint-Malo, France, 2009.

[2] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approx.*, vol. 28, pp. 253–263, 2008.

[3] D. Baron, M. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk, "Distributed compressed sensing," [Online]. Available: http://dsp.rice.edu/cs/DCS112005.pdf

[4] J. Bect, L. Blanc-FLeraud, G. Aubert, and A. Chambolle, "A ℓ_1-unified variational framework for image restoration," in *Proc. Eur. Conf. Comput. Vis.*, 2004, no. 4, pp. 1–13.

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

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

[7] J. M. 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.

[8] E. Candés and Y. Plan, "Near-ideal model selection by L1 minimization," *Ann. Statist.*, vol. 37, no. 5A, pp. 2145–2177.

[9] E. Candés and J. Romberg, "Practical signal recovery from random projections," in *Proc. Wavelet Applicat. Signal Image Process. XI, Proc. SPIE Conf. 5914*, 2004.

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

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

[12] E. Candés and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Problems*, vol. 23, pp. 969–985, 2006.

[13] E. Candés and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

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

[15] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, pp. 89–97, 2004.

[16] R. Chartrand, "Fast algorithms for nonconvex compressive sensing: MRI reconstruction from very few data," in *Proc. IEEE Int. Symp. Biomed. Imag. (ISBI)*, 2009.

[17] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, 1998.

[18] P. L. Combettes and J.-C. Pesquet, "A proximal decomposition method for solving convex variational inverse problems," *Inverse Problems*, vol. 24, no. 6, 2008, Article ID 065014.

[19] C. De Mol and M. Defrise, "A note on wavelet-based inversion algorithms," *Contemp. Math.*, vol. 313, pp. 85–96, 2002.

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

[21] D. Donoho, "For most large underdetermined systems of linear equations, the minimal l1-norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, no. 7, pp. 907–934, 2006.

[22] J. Douglas and H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Trans. Amer. Math. Soc.*, vol. 82, pp. 421–439, 1956.

[23] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, 1992, North-Holland.

[24] E. Esser, "Applications of Lagrangian-based alternating direction methods and connections to split Bregman," UCLA, Los Angeles, CAM Rep. 09–31, 2009.

[25] M. Figueiredo, J. Bioucas-Dias, and M. V. Afonso, "Fast frame-based image deconvolution using variable splitting and constrained optimization," in *Proc. IEEE Workshop Statistical Signal Process.—SSP2009*, Cardiff, U.K., 2009.

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

[27] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations," *Comp. Math. Appl.*, vol. 2, pp. 17–40, 1976.

[28] R. Glowinski and A. Marrocco, "Sur lapproximation par elements finis dordre un, et la resolution par penalisation-dualite dune classe de problemes de Dirichlet nonlineaires," *Rev. Francaise dAut. Inf. Rech. Oper.*, vol. R-2, pp. 41–76, 1975.

[29] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*. New York, Berlin, Heidelberg, Tokyo: Springer-Verlag, 1984.

[30] R. Glowinski and P. Le Tallec, "Augmented Lagrangian and operator-splitting methods in nonlinear mechanics," in *SIAM Studies in Applied Mathematics*. Philadelphia, PA: SIAM, 1989.

[31] T. Goldstein and S. Osher, "The split Bregman method for l1 regularized problems," UCLA, Los Angeles, CAM Rep. 08–29, 2008.

[32] L. He, T.-C. Chang, S. Osher, T. Fang, and P. Speier, "MR image reconstruction by using the iterative refinement method and nonlinear inverse scale space methods," UCLA, Los Angeles, CAM Rep. 06–35, 2006.

[33] B. S. He, L. Z. Liao, D. Han, and H. Yang, "A new inexact alternating directions method for monotone variational inequalities," *Math. Progam. Ser. A*, vol. 92, pp. 103–118, 2002.

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

[35] S. Ma, W. Yin, Y. Zhang, and A. Chakraborty, "An efficient algorithm for compressed MR imaging using total variation and wavelets," in *Proc.IEEE Conf. CVPR*, 2008, pp. 1–8.

[36] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterated regularization method for total variation-based image restoration," *Multiscale Model. Simul.*, vol. 4, pp. 460–489, 2005.

[37] *Rice Wavelet Toolbox*, [Online]. Available: http://www.dsp.rice.edu/software/rwt.shtml

[38] S. Setzer, "Douglas-rachford splitting and frame shrinkage," in *Proc. 2nd Int. Conf. Scale Space Methods and Variational Methods in Comput. Vis., Lecture Notes in Computer Science*, 2009, Split Bregman Algorithm.

[39] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[40] Y. Wang, W. Yin, and Y. Zhang, "A fast algorithm for image deblurring with total variation regularization," TR07-10 2007, CAAM, Rice Univ..

[41] Y. Wang, J.-F. 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.

[42] S. J. Wright, R. D. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 3373–3376.

[43] J.-F. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 569–592, 2009.

[44] J.-F. Yang, Y. Zhang, and W. Yin, "An efficient TVL1 algorithm for deblurring of multichannel images corrupted by impulsive noise," *SIAM J. Sci. Comput.*, vol. 31, no. 4, pp. 2842–2865, 2009.

[45] J.-F. Yang, Y. Zhang, and W. Yin, "A fast TVL1-L2 minimization algorithm for signal reconstruction from partial fourier data," TR08-27 2008, CAAM, Rice Univ..

[46] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for L1-minimization with applications to compressed sensing," *SIAM J. Imag. Sci.*, vol. 1, no. 1, pp. 143–168, 2008.

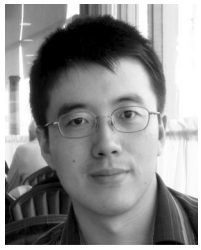
[47] Y. Zhang, "On theory of compressive sensing via ℓ_1-minimization: Simple derivations and extensions," TR08-11 CAAM, Rice Univ. Submitted to SIREV.

Junfeng Yang received the B.S. degree in mathematics from Hebei Normal University, Hebei, China, in 2003 and the Ph.D. degree in computational mathematics from Nanjing University, Nanjing, China, in 2009,

He is an Assistant Professor in the Department of Mathematics at Nanjing University. His research interests include optimization theory and algorithms and applications in signal and image processing.

Yin Zhang received the B.S. degree in environmental engineering from Chongqing Institute of Architecture and Engineering, Chongqing, China, in 1977 and the Ph.D. degree in applied mathematics from the State University of New York at Stony Brook in 1987

He is a Professor in the Department of Computational and Applied Mathematics at Rice University, Houston, TX. His recent research activities concentrate on developing and studying numerical algorithms for practical optimization problems arising from various applications including image and data processing.



Wotao Yin received the B.S. degree in mathematics from Nanjing University, Nanjing, China, in 2001 and the Ph.D. degree in operation research from Columbia University, New York, in 2006.

He is an Assistant Professor in the Department of Computational and Applied Mathematics at Rice University, Houston, TX. His research interests include developing and studying numerical algorithms for ill-posed inverse problems, sparse optimization, and compressive sensing.