A fast TVL1-L2 minimization algorithm for 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 a simple and fast algorithm for signal reconstruction from partial Fourier data. The algorithm minimizes the sum of three terms corresponding to total variation, ℓ_1-norm regularization and least squares data fitting. It uses an alternating minimization scheme in which the main computation involves shrinkage and fast Fourier transforms (FFTs), or alternatively discrete cosine transforms (DCTs) when available data are in the DCT domain. We analyze the convergence properties of this algorithm, and compare its numerical performance with two recently proposed algorithms. Our numerical simulations on recovering magnetic resonance images (MRI) indicate that the proposed algorithm is highly efficient, stable and robust.**

***Index Terms*—compressive sensing, compressed sensing, MRI, MRI reconstruction, fast Fourier transform, discrete cosine transform.**

I. Introduction

Let $\bar{u} \in \mathbb{R}^N$ be an unknown signal. Following the standard treatment, we will 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 less 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 orthogonal 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 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 certain subspace. Let M be an integer satisfies $K < M \ll N$ and $\Phi \in \mathbb{C}^{M\times N}$ be a general nonadaptive sensing matrix. Instead of acquiring $\bar{u}$, CS 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 is fixed and does not depend on $\bar{u}$. Basic CS theory [5], [6], [12] justifies that it is extremely probable to reconstruct $\bar{x}$ accurately or even exactly from b as long as $\bar{x}$ is sparse or compressible and A possesses certain nice attributes. To make CS practical, one needs to design a good sensing matrix A (encoder), which ensures that b contains enough information as $\bar{x}$ does, and an efficient reconstruction algorithm (decoder), which recovers $\bar{x}$ 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 *restricted isometry property* (RIP) [7], [6]. 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, authors of [6], [12] showed that this property holds with high probability for random matrices, e.g., matrix with independent and identical distributed (iid) Gaussian entries. For orthogonal Ψ, moreover, $A = \Phi\Psi$ will have the desired RIP attribute if Φ is iid Gaussian matrix. For other distributions which lead to RIP, see e.g., [1]. It is pointed out in [23] that for "almost all" random sensing matrices the recoverability of getting $\bar{x}$ from b is asymptotically identical. Aside from random sensing matrices, exact reconstruction is also attainable when A is a random partial Fourier matrix [6], which has important applications in magnetic resonance imaging (MRI) and will be the focus of this paper.

Being underdetermined, equation (2) usually has infinitely many solutions. If we know in advance that b is acquired from a highly sparse signal, a reasonable approach would be seeking

J. Yang is a PhD candidate in the Department of Mathematics, Nanjing University, Nanjing, Jiangsu, 210093, China (E-Mail: jfyang2992@yahoo.com.cn).

Y. Zhang is a professor in the Department of Computational and Applied Mathematics, Rice University, Houston, TX, 77005, USA (Tel.: 713-348-5744, Fax: 713-348-5318, E-Mail: yin.zhang@rice.edu).

W. Yin is an assistant professor in the Department of Computational and Applied Mathematics, Rice University, Houston, TX, 77005, USA (Tel.: 713-348-5368, E-Mail: wotao.yin@rice.edu).

the sparsest one among all 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 $K+1$ iid Gaussian measurements [2]. Unfortunately, directly solving this ℓ_0 decoder is generally impractical in computation. A common substitute for (3) is the well known basis pursuit problem [10]:

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

It has been shown that, under some desirable conditions, with high probability problems (3) and (4) share common solutions (see, for example, [13]). For 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 iid Gaussian matrix [8] and $O(K\log N)$ when A is random partial Fourier matrix (as in MRI) [6], both of which are, though larger than K, much smaller than N. Moreover, (4) is easily transformed to a linear program 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 decoding model (4). 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

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

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 [5] for example. A related problem to (5) is

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

where $\lambda > 0$. From 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 [19] for example.

B. Compressible image reconstruction 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 in a certain frequency and get aligned to the direction of the magnet. Upon stopping the radio frequency, the physical system gets back 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 MRI system works as related to CS, see [16] and references therein. Unfortunately, this data acquisition process is quite time consuming due to physiological and hardware constraints. Meanwhile, patients have to endure long scanning sessions while their bodies are restrained in order to reduce motion artifacts. All these facts hint at the importance of reducing scan time, which means collecting less data, without sacrificing image quality.

In the rest of this paper, we will concentrate on the case of partial Fourier data, i.e., in (7) $\mathcal{T} = \mathcal{F}$ is 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 the solution of the following TVL1-L2 model

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

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 $\tau, \lambda > 0$ are scalars 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 [16]. Therefore, we will choose the sparsity promoting basis Ψ as a wavelet basis.

Model (8) was studied in [4], [15], [16], [17] and reported to reconstruct high quality MR images from a small number of Fourier coefficients [17]. Our main contribution in this paper is a very simple and fast algorithm for solving model (8).

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 in $D_i u$ 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 $\|\cdot\| = \|\cdot\|_2$ and let $\rho(\cdot)$ be the spectral radius of its argument.

D. Organization

The paper is organized as follows. In Section II, the main algorithm is presented followed by a study of optimality conditions of (8). Section III focuses on the algorithm's convergence. Numerical results are given in Section IV, where our algorithm is compared to TwIST [3] and an operator splitting based algorithm [17]. Finally, some conclusion remarks are given in Section V.

II. Basic algorithm and optimality

The main difficulty in solving (8) is caused by the non-differentiability of its first and second terms. Our approach is to approximate them by using the classic quadratic penalty technique in optimization, which dates back to Courant's work [11] in 1943. This approach can also be viewed as applying the half-quadratic technique [14] to an appropriate approximation of (8); see [21].

A. An alternating algorithm

We introduce auxiliary variables $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_N]$, where each $\mathbf{w}_i \in \mathbb{R}^2$, and $z \in \mathbb{R}^N$. Clearly, (8) is equivalent to

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

To relax the equality constraints and penalize their violations by quadratic functions, we further introduce

$$\phi_1(s,t) = \tau|s| + (\beta_1/2)\cdot|s-t|^2,\ s,t \in \mathbb{R}$$

and

$$\phi_2(\mathbf{s},\mathbf{t}) = \|\mathbf{s}\| + (\beta_2/2)\cdot\|\mathbf{s}-\mathbf{t}\|^2,\ \mathbf{s},\mathbf{t} \in \mathbb{R}^2$$

for given $\beta_1, \beta_2 > 0$. Problem (9) can then be approximated by

$$\min \sum_i \phi_2(\mathbf{w}_i, D_i u) + \sum_i \phi_1(z_i, \psi_i^\top u) + \lambda\theta(u, f_p), \tag{10}$$

where ψ_i is the ith column of Ψ. For simplicity, we assume $\beta_1 = \beta_2 \equiv \beta$. We note that this assumption does not lose generality because, otherwise, a scaling of $\tau \leftarrow \tau\sqrt{\beta_2/\beta_1}$ and $(\Psi, z) \leftarrow \sqrt{\beta_1/\beta_2}(\Psi, z)$ will equalize the penalty parameters. Therefore, we derive our algorithm and analyze its convergence under this assumption, though the two parameters can be different in practical implementation. In spite of more decision variables compared to (8), problem (10) is easier to minimize with respect to $\mathbf{w}$, z, and u each. For a fixed u, the minimization 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 in (10). Based on these observations, it is easy to apply alternating minimization to (10) as follows. First, for a fixed u, the minimizer z_i is obtained by the one-dimensional shrinkage:

$$z_i = s_1(\psi_i^\top u), \tag{11}$$

where $s_1(t)$ minimizes $\phi_1(s,t)$ for a fixed t, given by

$$s_1(t) \triangleq \max\{|t| - \tau/\beta, 0\}\cdot\operatorname{sgn}(t),\ t \in \mathbb{R}, \tag{12}$$

and the minimizer $\mathbf{w}_i$ is given by the two-dimensional shrinkage [20]:

$$\mathbf{w}_i = s_2(D_i u), \tag{13}$$

where $s_2(\mathbf{t})$ minimizes $\phi_2(\mathbf{s},\mathbf{t})$ for a fixed $\mathbf{t}$, given by

$$s_2(\mathbf{t}) \triangleq \max\{\|\mathbf{t}\| - 1/\beta, 0\}\cdot\mathbf{t}/\|\mathbf{t}\|,\ \mathbf{t} \in \mathbb{R}^2, \tag{14}$$

where $0\cdot(0/0) = 0$ is assumed. The computational costs for (11) and (13) are linear in terms of N. Secondly, for fixed $\mathbf{w}$ and z, the minimization of (10) with respect to u becomes a least squares problem

$$\min_u \sum_i \left(\|\mathbf{w}_i - D_i u\|^2 + |z_i - \psi_i^\top u|^2\right) + 2\eta\theta(u, f_p), \tag{15}$$

where $\eta = \lambda/\beta$. Let $w_j \triangleq (\mathbf{w}_1(j); \ldots; \mathbf{w}_N(j))$, $j = 1, 2$. From the definitions of $D^{(j)}$ and w_j, $j = 1, 2$, it follows

$$\sum_i \|\mathbf{w}_i - D_i u\|^2 = \|w_1 - D^{(1)}u\|^2 + \|w_2 - D^{(2)}u\|^2.$$

Because of the orthogonality of Ψ, the normal equations of (15) can be written as

$$Mu = y, \tag{16}$$

where

$$M = (D^{(1)})^\top D^{(1)} + (D^{(2)})^\top D^{(2)} + I + \eta\mathcal{F}_p^\top\mathcal{F}_p$$

and

$$y = (D^{(1)})^\top w_1 + (D^{(2)})^\top w_2 + \Psi z + \eta\mathcal{F}_p^\top f_p.$$

Since $D^{(1)}$ and $D^{(2)}$ are finite difference operators, under the periodic boundary conditions for u, they are circulant matrices and can be diagonalized by the 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, $j = 1, 2$. Multiplying $\mathcal{F}$ on both sides of (16), we obtain

$$\hat{M}\mathcal{F}(u) = \hat{y}, \tag{17}$$

where

$$\hat{M} = (\hat{D}^{(1)})^\top\hat{D}^{(1)} + (\hat{D}^{(2)})^\top\hat{D}^{(2)} + I + \eta P^\top P$$

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

$$\hat{y} = \mathcal{F}[(D^{(1)})^\top w_1 + (D^{(2)})^\top w_2 + \Psi z] + \eta P^\top f_p.$$

Therefore, solving (17) is straightforward, which means that (16) can be easily solved for given w_1, w_2, and z as follows. First, compute $(D^{(j)})^\top w_j$, $j = 1, 2$, and Ψz. Second, compute $\hat{y}$ which costs one FFT. Third, solve (17) to get $\mathcal{F}(u)$ where it is assumed that the constants $\hat{D}^{(1)}$ and $\hat{D}^{(2)}$ are precalculated.

Finally, apply the inverse FFT to $\mathcal{F}(u)$ to obtain the solution u to (16).

Since minimizing the objective function in (10) with respect to each variable is computationally inexpensive, we propose the following alternating minimization algorithm:

Algorithm 1: Input P, f_p; $\tau, \lambda, \beta > 0$. Initialize $u = u^0$.
While "not converged", **Do**
1) Given u, compute z and $\mathbf{w}$ by (11) and (13).
2) Given z and $\mathbf{w}$, compute u by solving (16).
End Do

B. Optimality conditions and practical implementation

In order to specify a stopping criterion for Algorithm 1, we derive optimality conditions for (8) and (10). Let the signum set-valued function be defined as

$$\mathrm{SGN}(t) = \begin{cases} \{\mathrm{sgn}(t)\} & t \neq 0, \\ \text{[-1,1]} & t = 0, \end{cases}$$

which is the subdifferential of $|t|$. For $v \in \mathbb{R}^N$, we let

$$\mathrm{SGN}(v) = \left\{ \xi \in \mathbb{R}^N : \xi_i \in \mathrm{SGN}(v_i), \forall i \right\},$$

which is the subdifferential of $\|v\|_1$. We need the following propositions, the proofs of which are elementary and can be found in [20], [22], for example.

Proposition 1: For any $A \in \mathbb{R}^{p\times n}$, it holds that

$$\partial_x \|Ax\| = \begin{cases} \{A^\top Ax/\|Ax\|\} & \text{if } Ax \neq 0; \\ \left\{A^\top h : \|h\| \leq 1, h \in \mathbb{R}^p\right\} & \text{otherwise.} \end{cases}$$

Proposition 2: For any $B \in \mathbb{R}^{m\times n}$, it holds that

$$\partial_x \|Bx\|_1 = \left\{ B^\top g : g \in \mathrm{SGN}(Bx) \right\}.$$

Since the objective function is convex, a triplet $(z, \mathbf{w}, u)$ is a solution of (10) if and only if the subdifferential of the objective at $(z, \mathbf{w}, u)$ contains the origin. In light of propositions 1 and 2 with A and B being identity matrices of appropriate sizes, $(z, \mathbf{w}, u)$ solves (10) if and only if

$$\begin{cases} \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|} + \beta(\mathbf{w}_i - D_i u) = 0, & i \in I_1 \triangleq \{i : \mathbf{w}_i \neq \mathbf{0}\}, \\ \|D_i u\| \leq 1/\beta, & i \in I_2 \triangleq \{i : \mathbf{w}_i = \mathbf{0}\}, \\ \tau\,\mathrm{sgn}(z_i) + \beta(z_i - \psi_i^\top u) = 0, & i \in I_3 \triangleq \{i : z_i \neq 0\}, \\ |\psi_i^\top u| \leq \tau/\beta, & i \in I_4 \triangleq \{i : z_i = 0\}, \end{cases}$$
$$D^\top(Du - w) + u - \Psi z + \eta \nabla\theta(u, f_p) = 0,$$

where $\nabla\theta(u, f_p) \triangleq \mathcal{F}_p^\top(\mathcal{F}_p u - f_p)$, $D = (D^{(1)}; D^{(2)})$, and $w = (w_1; w_2)$. Eliminating $\mathbf{w}$ and z from the above equations, we get

$$\sum_{i\in I_1} D_i^\top \frac{D_i u}{\|D_i u\|} + \sum_{i\in I_2} D_i^\top h_i + \tau \sum_{i\in I_3} \mathrm{sgn}(\psi_i^\top u)\psi_i + \tau \sum_{i\in I_4} g_i \psi_i + \lambda \nabla\theta(u, f_p) = 0, \quad (18)$$

where $h_i \triangleq \beta D_i u$ satisfies $\|h_i\| \leq 1$ and $g_i \triangleq \beta \psi_i^\top u/\tau$ satisfies $|g_i| \leq 1$. Now, we show that (18) is an approximation of the optimality of (8). Let u^* be any solution of (8) and define

$$I_1^* \triangleq \{i : D_i u^* \neq 0\},\ I_2^* \triangleq \{i : D_i u^* = 0\},$$
$$I_3^* \triangleq \{i : \psi_i^\top u \neq 0\},\ I_4^* \triangleq \{i : \psi_i^\top u = 0\}.$$

In light of propositions 1 and 2, there exist $\{h_i^* \in \mathbb{R}^2 : \|h_i^*\| \leq 1, i \in I_2^*\}$ and $\{g_i^* \in \mathbb{R} : |g_i^*| \leq 1, i \in I_4^*\}$ such that

$$\sum_{i\in I_1^*} D_i^\top \frac{D_i u^*}{\|D_i u^*\|} + \sum_{i\in I_2^*} D_i^\top h_i^* + \tau \sum_{i\in I_3^*} \mathrm{sgn}(\psi_i^\top u^*)\psi_i + \tau \sum_{i\in I_4^*} g_i^* \psi_i + \lambda \nabla\theta(u^*, f_p) = 0. \quad (19)$$

Equation (18) differs from (19) only in the index sets involved. As β increases, (18) approximates (19) closer.

The stopping criterion of Algorithm 1 is based on the optimality conditions of (10). Let

$$\begin{cases} r_1(i) = (\mathbf{w}_i/\|\mathbf{w}_i\|)/\beta + \mathbf{w}_i - D_i u & i \in I_1, \\ r_2(i) = \|D_i u\| - 1/\beta & i \in I_2, \\ r_3(i) = \tau\,\mathrm{sgn}(z_i)/\beta + z_i - \psi_i^\top u & i \in I_3, \\ r_4(i) = |\psi_i^\top u| - \tau/\beta & i \in I_4, \end{cases}$$

and

$$r_5 = D^\top(Du - w) + u - \Psi z + \eta \nabla\theta(u, f_p).$$

We terminate Algorithm 1 once the following conditions are satisfied:

$$\begin{cases} \max\{\|r_1(i)\| : i \in I_1\} \leq \epsilon, \\ \max\{r_2(i) : i \in I_2\} \leq \epsilon, \\ \max\{|r_3(i)| : i \in I_3\} \leq \epsilon, \\ \max\{r_4(i) : i \in I_4\} \leq \epsilon, \\ \|r_5\| \leq \epsilon \cdot \|D^\top w + \Psi z + \eta \mathcal{F}_p^\top f_p\|, \end{cases} \quad (20)$$

where $\epsilon > 0$ is a prescribed tolerance, and the last item in (20) measures the relative error of (16) for given w and z.

In a practical implementation of Algorithm 1, we assign β a small value at the beginning and increase it gradually. For a fixed β, we apply Algorithm 1 to (10) until condition (20) with a prescribed accuracy is satisfied. The obtained solution is then used to warm-start Algorithm 1 corresponding to the next β. Such a continuation or "path-following" technique is widely used in the class of penalty methods and also well-justified by our convergence results in Section III.

We now formally present the default algorithm of this paper that will be used in our numerical experiments.

Algorithm 2: Input data P, f_p, model parameters $\tau, \lambda > 0$, and tolerance $\epsilon > 0$. Initialize $\beta_1 = \beta_2 = \beta = 2^5$ and $u = u^0$.
While $\beta <= 2^{10}$, **Do**
1) Starting from u, solve (10) by Algorithm 1 until (20) is met, and return solution u_β.
2) Update $u \leftarrow u_\beta$, $\beta \leftarrow 2 * \beta$.
End Do

It is certainly possible to both increase β and stop the inner iterations in a more adaptive way for better performance. However, we keep our implementation as simple as Algorithm 2 throughout our experiments in this paper. The final value 2^{10} for β is so chosen because, based on empirical evidence, it is sufficiently large to make (10) a close approximation of (8) in the sense that a larger β value would not notably increase the accuracy of solution. The remaining free parameter in

Algorithm 2 is the error tolerance ϵ used in (20), which will be specified in Section IV.

III. Convergence analysis

It is well-known that the quadratic penalty method applied to a problem like (9) converges to its solution as the penalty parameter goes to infinity. In this section, we study the convergence of Algorithm 1, as well as its rate, for a fixed β. We establish global convergence, finite convergence for some auxiliary variables and q-linear convergence results, which are stronger than the classic results of penalty methods.

For $u, v \in \mathbb{R}^N$, we let, for s_1, s_2 defined in (12) and (14),

$$S_1(u) = \begin{pmatrix} s_1(u_1) \\ \vdots \\ s_1(u_N) \end{pmatrix} \text{ and } S_2(u; v) = \begin{pmatrix} s_2(u_1; v_1) \\ \vdots \\ s_2(u_N; v_N) \end{pmatrix}.$$

Recall that $w = (w_1; w_2)$. Given u^k, z and w are iterated as

$$z^{k+1} = S_1(\Psi^\top u^k) \tag{21}$$

and

$$w^{k+1} = S_2\left(D^{(1)}u^k; D^{(2)}u^k\right), \tag{22}$$

respectively. Let $v = (z; w)$, $\xi = \eta \mathcal{F}_p^\top f_p$ and

$$H = \left(\Psi^\top; D^{(1)}; D^{(2)}\right).$$

Then the new iterate u^{k+1} satisfies

$$Mu^{k+1} = H^\top v^{k+1} + \xi. \tag{23}$$

Let

$$h(v) = \begin{pmatrix} h^{(1)}(v) \\ h^{(2)}(v) \\ h^{(3)}(v) \end{pmatrix} = \begin{pmatrix} \Psi^\top \\ D^{(1)} \\ D^{(2)} \end{pmatrix} M^{-1}\left(H^\top v + \xi\right)$$

and

$$S \circ h = \begin{pmatrix} S_1 \circ h^{(1)} \\ S_2 \circ (h^{(2)}; h^{(3)}) \end{pmatrix}.$$

Then the iteration of Algorithm 1 can be expressed as

$$v^{k+1} = S \circ h(v^k) \tag{24}$$

and

$$u^{k+1} = M^{-1}\left(H^\top v^{k+1} + \xi\right). \tag{25}$$

Since the objective function of (10) is coercive (it goes to infinity whenever $\|(z; w; u)\|$ does), there exists $(v^*; u^*)$ such that

$$v^* = S \circ h(v^*) \tag{26}$$

and

$$u^* = M^{-1}\left(H^\top v^* + \xi\right). \tag{27}$$

Lemma 3.1: For any $u \neq v$, it holds that

$$\|h(u) - h(v)\| \leq \|u - v\|$$

with the equality holding if and only if $h(u) - h(v) = u - v$.

Lemma 3.2: Let v^* be any fixed point of $S \circ h$. For any v, we have $\|S \circ h(v) - S \circ h(v^*)\| < \|v - v^*\|$ unless v is a fixed point of $S \circ h$.

Theorem 3.3: The sequence $\{(z^k, w^k, u^k)\}$ generated by Algorithm 1 from any starting point (z^0, w^0, u^0) converges to a solution (z^*, w^*, u^*) of (10).

The proofs of Lemmas 3.1 and 3.2, and Theorem 3.3 are similar to those in [20] for a slightly different problem and thus are omitted.

Next we develop a finite convergence property for parts of the auxiliary variables z and w. Let

$$g_i(v) = \begin{pmatrix} h_i^{(2)}(v) \\ h_i^{(3)}(v) \end{pmatrix} \in \mathbb{R}^2, \ \forall i,$$

where $h_i^{(j)}(v)$ is the ith component of $h^{(j)}(v)$, $j = 2, 3$. We will make use of the following index sets:

$$L_1 = \left\{i : |\psi_i^\top u^*| \equiv |h_i^{(1)}(v^*)| < \tau/\beta\right\},$$
$$L_2 = \left\{i : \|D_i u^*\| \equiv \|g_i(v^*)\| < 1/\beta\right\},$$

and their complements $E_j = \{1, \ldots, N\} \setminus L_j$, $j = 1, 2$.

Theorem 3.4: The sequence $\{(z^k, w^k, u^k)\}$ generated by Algorithm 1 from any starting point $(v^0, u^0) = (z^0, w^0, u^0)$ satisfies $z_i^k = z_i^* = 0$, $\forall\, i \in L_1$, and $\mathbf{w}_i^k = \mathbf{w}_i^* = 0$, $\forall\, i \in L_2$ for all but finite numbers of iterations that do not exceed $\|v^0 - v^*\|^2/\omega_1^2$ and $\|v^0 - v^*\|^2/\omega_2^2$, respectively, where

$$\omega_1 \triangleq \min_{i \in L_1}\left\{\tau/\beta - |h_i^{(1)}(v^*)|\right\} > 0 \tag{28}$$

and

$$\omega_2 \triangleq \min_{i \in L_2}\left\{1/\beta - \|g_i(v^*)\|\right\} > 0. \tag{29}$$

Next we present the q-linear convergence of u^k and the remaining components in v^k, i.e., those which can remain nonzero infinitely many iterations. For convenience, we let

$$L = L_1 \cup (N + L_2) \cup (2N + L_2)$$

and $E = \{1, \ldots, 3N\} \setminus L$ be the complement of L. Let v_L be the subvector of v with components in L and similarly for v_E. Furthermore, let

$$T = HM^{-1}H^\top \text{ and } T_{EE} = [T_{i,j}]_{i,j \in E}.$$

Theorem 3.5: There exists K such that the sequence $\{(v^k, u^k)\}$ generated by Algorithm 1 satisfies

1) $\|v_E^{k+1} - v_E^*\| \leq \sqrt{\rho((T^2)_{EE})}\|v_E^k - v_E^*\|$;
2) $\|u^{k+1} - u^*\|_{H^\top H} \leq \sqrt{\rho((T^2)_{EE})}\|u^k - u^*\|_{H^\top H}$;

for all $k > K$.

The proofs of Theorems 3.4 and 3.5 are given in Appendices A and B, respectively. Theorem 3.5 states that Algorithm 1 generates a sequence of points that converge q-linearly with a convergence rate depending on the spectral radius of the submatrix $(T^2)_{EE}$ rather than that of T^2. From the previous definitions about the index sets, it is not difficult to argue that smaller β tends to yield a set E with a smaller cardinality and thus gives a faster convergence rate according to Theorem 3.5, which justifies the advantages of continuation on β. Since $(T^2)_{EE}$ is a minor of T^2, from the definition of T, it is obvious that $\rho((T^2)_{EE}) \leq \rho(T^2) \leq 1$ and $\rho(T^2) < 1$ whenever $\tau > 0$.

IV. Experimental results

A. General description

For simplicity, we give Algorithm 2 the name RecPF (reconstruction form partial Fourier data). In this section, we present simulation results of MR image reconstruction obtained by RecPF and two other recently proposed algorithms: a two-step iterative shrinkage/thresholding algorithm (TwIST) [3] and an operator splitting based algorithm that we call OS [17], both of which have been regarded fast for solving (8). All experiments were performed under Windows Vista Premium and MATLAB v7.6 (R2008a) running on a Lenovo laptop with an Intel Core 2 Duo CPU at 1.8 GHz and 2 GB of memory.

We generated our test sets using three images, the Shepp-Logan phantom image and two brain MR images, with different sampling ratios. In each test, we generated the data f_p in (7) by first rescaling the intensity values of the tested image to $[0, 1]$ followed by applying a partial FFT and adding Gaussian noise. To apply a partial FFT, we sampled the Fourier domain along a number of radial lines (RLs) spread out from the center; for example, Figure 1 shows 22 radial lines in a Fourier domain.

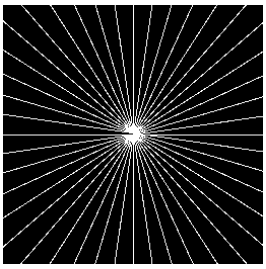

Fig. 1. Fourier domain sampling positions with 22 radial lines for test 1.

In all tests, the additive Gaussian noise had a mean zero and standard deviation 0.01. We tried different starting points for the three algorithms and found that all of them are insensitive towards starting points. Therefore, we simply initialized the starting image u by zero. Table I summarizes the test data, as well as the values of the parameters λ and τ used in (8).

TABLE I
Test images information and model parameter values

Test	Image	Size	RLs	Sample Ratio	(λ, τ)
1	phantom	256×256	22	9.36%	(1e2∼1e5, 0)
2	brain-1	256×256	66	26.85%	(2e3, 1)
3	brain-2	512×512	88	18.59%	(2e3, 1)

It is important to 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 Au and $A^\top v$ are computable for vectors u and v with appropriate lengths, but RecPF is limited to the cases where $\mathcal{T}$ in (7) must correspond to an orthogonal 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).

In addition, both TwIST and OS solve a scaled version of (8) where the objective function in (8) is scaled by $1/\lambda$. To keep compatibility, we multiply the objective function by $1/\lambda$ for RecPF. Hence, the objective values presented below for (λ, τ) were scaled by $1/\lambda$.

B. Comparison with TwIST

In test 1, we compare RecPF with TwIST, which in general solves

$$\min_u \lambda^{-1}\Phi_{\rm reg}(u) + \frac{1}{2}\|Au - b\|^2,$$

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

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

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

$$\Psi_\lambda(\xi_k) = \arg\min_u \lambda^{-1}\Phi_{\rm reg}(u) + (1/2)\cdot\|u - \xi_k\|^2. \quad (30)$$

Its latest version, TwIST_v1, cannot solve problem (8) with both the TV and ℓ_1 regularization terms. Hence, in order to compare RecPF with TwIST on the same model, we drop the ℓ_1-term in (8) by setting $\tau = 0$ for RecPF and letting $\Phi_{\rm reg} = \sum_i \|D_i u\|_2$ and $A = \mathcal{F}_p$ for TwIST. In TwIST_v1, the subproblem (30) is solved iteratively by Chambolle's algorithm [9]. In comparison, RecPF with the ℓ_1 term dropped has a per-iteration cost of 2 FFTs (including 1 inverse FFT), which is much lower than solving (30) by Chambolle's algorithm. This is one of the main reasons that RecPF runs faster.

We set the stopping parameter $\epsilon = 10^{-3}$ in (20) for RecPF, and used the monotonic variant of TwIST in TwIST_v1, which was stopped when the relative change in the objective function fell below $tol = 10^{-3}$. In TwIST_v1, the parameters α and δ are determined carefully based on the spectral distribution of $A^\top A$. In this particular case, i.e. $A = \mathcal{F}_p$, the minimum and maximum eigenvalues of $A^\top A$ are obviously 0 and 1, respectively. Therefore, we assigned a relatively small value 10^{-3} to the TwIST parameter `lam1` (which determines α and δ), as recommended by the TwIST_v1 documentation.

Table II gives the performance results of RecPF and TwIST for different values of λ. We also tested TwIST with `lam1`$= 10^{-4}$, but do not include the obtained results since they are similar to those for `lam1`$= 10^{-3}$. Table II lists the following quantities: 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 in seconds (CPU).

TABLE II
Results on Test 1.

	TwIST				RecPF			
λ	Err	Obj	Iter	CPU	Err	Obj	Iter	CPU
1e+2	5.4%	15.15	25	25	8.6%	14.79	180	15
1e+3	4.4%	1.919	106	66	4.5%	1.838	195	16
1e+4	43.4%	.4452	10	11	4.8%	.2998	187	15
1e+5	43.5%	.1368	10	9	4.9%	.1359	186	15

We point out that upon termination RecPF returned smaller objective values than TwIST for both $\lambda = 10^2$ and $\lambda = 10^3$, though the solutions from TwIST have slightly smaller relative errors. For $\lambda = 10^4$, TwIST failed to converge with $tol = 10^{-3}$ and obtained an image with relative error 43.4%. In this case, if tol is decreased from 10^{-3} to 10^{-4} without other

changes, TwIST obtained a solution with relative error 7.1% but in much longer CPU time (about 280 seconds). For $\lambda = 10^5$, similar phenomena happened except TwIST also failed with $tol = 10^{-4}$.

From Table II, it can be seen that RecPF is faster than TwIST to attain a comparable accuracy. This is primarily because of the difference in per-iteration costs of the two: TwIST solves the TV problem (30) using Chambolle's algorithm (that requires its own iteration loop), while RecPF solves (16) at an approximate cost of 2 FFTs. In terms of the iteration numbers and CPU time consumed, RecPF is relatively stable as λ varies while TwIST appears to be sensitive.

The reconstructed images corresponding to $\lambda = 10^3$ appear to have the best quality for both methods, and are given in Figure 2.

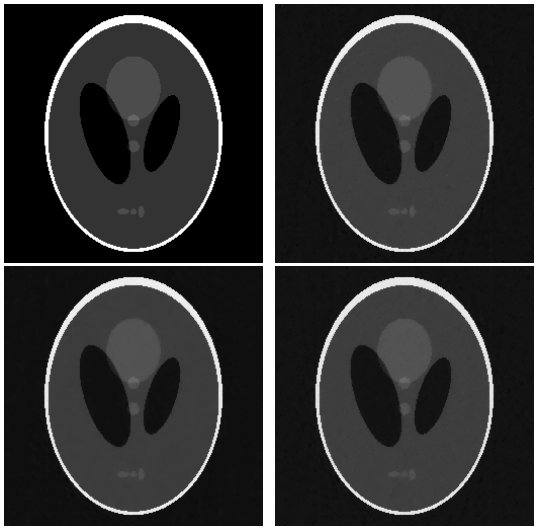

Fig. 2. Reconstructed images in test 1. Top left: original image. Top right: reconstruction by TwIST with λ=1e+3 (Err = 4.42%). Bottom left: reconstruction by RecPF with λ=1e+3 (Err = 4.48%). Bottom right: reconstructed by RecPF with λ=1e+10 (Err = 4.89%).

We also tested the two algorithms on data containing less or no noise and observed similar relative performances. For example, on noiseless data, both algorithms were able to converge to solutions with relative errors less than 1% for $\lambda = 10^3$, but RecPF was faster than TwIST.

In these experiments, we found that TwIST_v1 seems to require careful selections of parameters such as α, δ, λ and error tolerance tol in order to obtain results comparable to those of RecPF. This behavior is exemplified by the fact that TwIST failed for $(\lambda, tol) = (10^4, 10^{-3})$ or $(10^5, 10^{-4})$, where `lam1`$= 10^{-3}$ was used, by returning solutions with relative errors over 40% (similar behavior happened for `lam1`$= 10^{-4}$ as well), while it did return a good solution with $(\lambda, tol) = (10^4, 10^{-4})$ at a cost of more iterations and a longer CPU time. We tested different stopping criteria provided by TwIST_v1 and found that the above cases apparently were not isolated cases.

In comparison, RecPF is simple and requires little tuning. As mentioned, the default setting given in Algorithm 2 was used throughout our tests where only the error tolerance ϵ in (20) was varied. Interestingly, the performance of RecPF appears to be totally insensitive to the values of λ, and it converges well even with huge λ values. For example, we tested RecPF with $\lambda = 10^{10}$ and obtained an image with relative error 4.89%, which is depicted at the bottom right corner in Figure 2. We tested even larger λ and obtained equally good images. This behavior can be explained by closely examining the linear system in (17).

Recall that $\eta = \lambda/\beta$ and P is a selection matrix. From the formulations of $\hat{M}$ and $\hat{y}$ below (17) it becomes clear that (i) the value of λ only affects those Fourier coefficients in $\mathcal{F}(u)$ corresponding to the positions where samples have been taken; and (ii) as λ gets larger, the entries of $\mathcal{F}(u)$ corresponding to the sampled positions gets closer to f_p. In the limit as $\lambda \to \infty$, solving (17) simply fills $\mathcal{F}(u)$ with f_p at the sampled positions, and updates the remaining entries of $\mathcal{F}(u)$ independent of λ. This separation makes RecPF very stable with large λ and allows it to faithfully retain the effect of TV regularization.

C. Comparison with OS

In tests 2 and 3, we compare RecPF with OS [17] on solving problem (8) with both TV and ℓ_1 terms. OS iterates the fixpoint equations (31) 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:

$$\begin{cases} s = \Psi^\top u - (\delta_1/\lambda) \cdot \Psi^\top \left(\sum_i D_i^\top \mathbf{w}_i + \lambda \nabla \theta(u, f_p) \right), \\ \mathbf{t}_i = \mathbf{w}_i + \delta_2 D_i u, \ \ \forall i, \\ u = \Psi \left\{ \max\left(|s| - \delta_1 \tau/\lambda, 0 \right) \circ \operatorname{sgn}(s) \right\}, \\ \mathbf{w}_i = \min\left(1/\delta_2, \|\mathbf{t}_i\|\right) \cdot \mathbf{t}_i / \|\mathbf{t}_i\|, \ \ \forall i. \end{cases} \tag{31}$$

The authors showed that for any fixed $\delta_1, \delta_2 > 0$, u is a solution of (8) if and only if it satisfies (31). 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 (31). 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), and OS also has continuation, decreasing τ and $1/\lambda$ from larger values to prescribed small ones. For a fixed pair of (τ, λ), the fixed-point iterations of OS terminate if one of the following two conditions are met:

$$f_k - f_{k+1} < \varepsilon_2 \sqrt{\tau_c/\tau_t} \max\{1, f_k\}, \tag{32}$$

$$\|u_{k+1} - u_k\| < \varepsilon_1 \max\{1, \|u_k\|\}, \tag{33}$$

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 the stopping tolerance $\epsilon = 2.5 \times 10^{-2}$ in (20) which, although much looser than the previous tolerance used in test 1, is sufficient to obtain better results than OS.

Since the two algorithms use different stopping criteria and OS has multiple tuning parameters such as δ_1 and δ_2 that influence the convergence speed, it is rather difficult to compare them at their best performance. Since OS implements

the fixed-point iterations based on (31) that do not directly aim at decreasing the 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 tends to yield monotonically decreasing objectives but converges slowly. For larger δ's and looser stopping criteria, OS becomes faster but loses objective monotonicity and returns solutions with larger relative errors. After having tried different combinations of ε's and δ's for OS, we determined to use the above parameter values in the results presented below as a best compromise between convergence speed and image quality.

In both tests 2 and 3, the sparsity promoting basis Ψ was set to be the Haar wavelet transform using the Rice Wavelet Toolbox in its default setting [18]. The per-iteration computational cost of both methods is 2 FFTs (including 1 inverse FFT) and 2 DWTs (including 1 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 Figures 3 and 4, respectively. The reconstructed images are give in Figure 5.

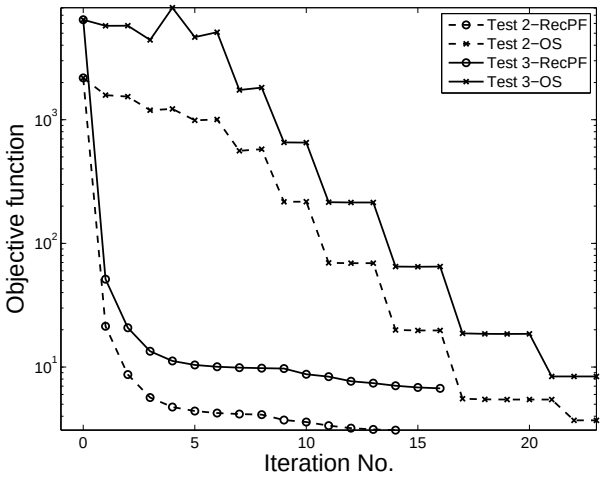

Fig. 3. Comparing RecPF with OS: objective value vs. iteration numbers.

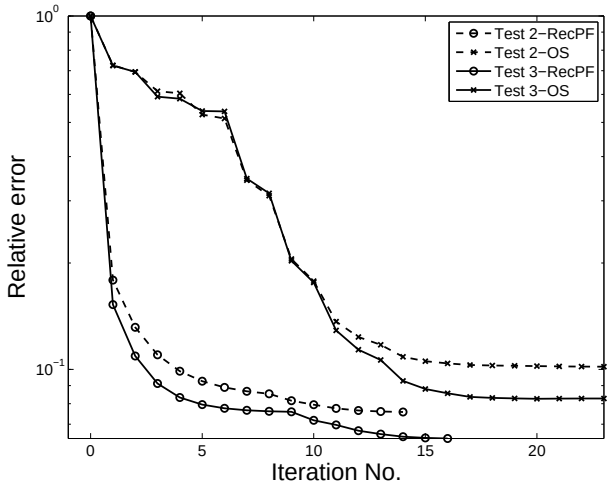

Fig. 4. Comparing RecPF with OS: relative error vs. iteration numbers.

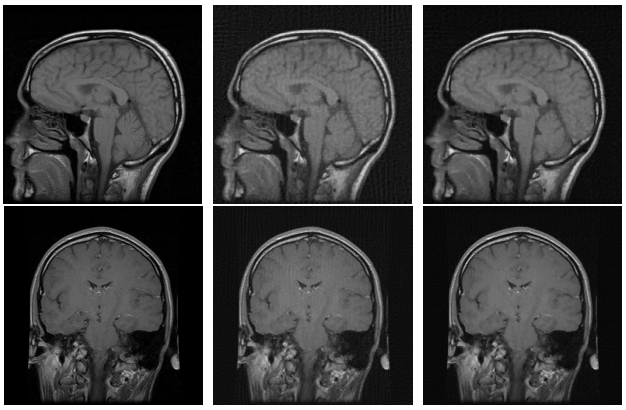

Fig. 5. Results of tests 2 and 3. Top row (results of test 2) from left to right: original, recovered by OS (Err: 10.18%) and RecPF (Err: 7.58%); Bottom row (results of test 3) from left to right: original, recovered by OS (Err: 8.27%) and RecPF (Err: 6.38%).

As can be seen from Figures 3 and 4, RecPF converged 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 less 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 Figure 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, their performance difference stay with a similar ratio.

V. Conclusion

Based on the classic penalty function approach in optimization, a highly efficient alternating minimization algorithm, called RecPF, was proposed for reconstructing large-scale signals or images from a subset of their frequency data (DFT or DCT coefficients). The algorithm minimizes a potential function with one or both of TV and ℓ_1-norm regularization terms. At each iteration, the main computation of RecPF only involves fast and stable operations consisting of shrinkages and FFTs (or DCTs). RecPF enjoys strong convergence properties and its practical convergence is accelerated by continuation on penalty parameters.

RecPF is compared to two efficient algorithms TwIST [3] and OS [17], which can solve problems with broader data types than RecPF. On image reconstruction problems with frequency data, the strong numerical evidence presented in this paper shows that RecPF is able to take the advantage of the frequency date type and achieves much higher performance in terms of both reconstruction speed and quality than TwIST and OS. It is hoped that RecPF can be useful in relevant areas of compressive sensing such as sparsity-based, rapid MR image reconstruction.

APPENDIX A
PROOF OF THEOREM 3.4

From (13), for any $i \in \{1, \ldots, N\}$, it holds that

$$\begin{aligned} \|\mathbf{w}_i^{k+1} - \mathbf{w}_i^*\|^2 &= \left\|s_2 \circ g_i(v^k) - s_2 \circ g_i(v^*)\right\|^2 \\ &\leq \left\|g_i(v^k) - g_i(v^*)\right\|^2. \end{aligned} \tag{34}$$

Suppose $\mathbf{w}_i^{k+1} \neq 0$ for some $i \in L_2$, then

$$\begin{aligned} & \|\mathbf{w}_i^{k+1} - \mathbf{w}_i^*\|^2 \\ = & \left\|s_2 \circ g_i(v^k) - s_2 \circ g_i(v^*)\right\|^2 \\ = & \left(\|g_i(v^k)\| - 1/\beta\right)^2 \\ \leq & \left\{\|g_i(v^k) - g_i(v^*)\| - (1/\beta - \|g_i(v^*)\|)\right\}^2 \\ \leq & \|g_i(v^k) - g_i(v^*)\|^2 - (1/\beta - \|g_i(v^*)\|)^2 \\ \leq & \|g_i(v^k) - g_i(v^*)\|^2 - \omega_2^2, \end{aligned} \tag{35}$$

where the first equality comes from the iteration of $\mathbf{w}_i$ in (13) and the definition of $g_i(v)$; the second equality holds because of $\|g_i(v^*)\| < 1/\beta$, $\mathbf{w}_i^{k+1} \neq 0$ and the definition of s_2; the first inequality is triangular inequality; the second inequality follows from the fact that $\|g_i(v^k) - g_i(v^*)\| \geq 1/\beta - \|g_i(v^*)\| > 0$; and the last one uses the definition of ω_2 in (29). Furthermore,

$$\begin{aligned} \|z^{k+1} - z^*\|^2 &= \|S_1 \circ h^{(1)}(v^k) - S_1 \circ h^{(1)}(v^*)\|^2 \\ &\leq \|h^{(1)}(v^k) - h^{(1)}(v^*)\|^2. \end{aligned} \tag{36}$$

Combining (34), (35) and (36), we get

$$\begin{aligned} \|v^{k+1} - v^*\|^2 &= \|z^{k+1} - z^*\|^2 + \|w^{k+1} - w^*\|^2 \\ &= \|z^{k+1} - z^*\|^2 + \sum_i \|\mathbf{w}_i^{k+1} - \mathbf{w}_i^*\|^2 \\ &\leq \|h^{(1)}(v^k) - h^{(1)}(v^*)\|^2 \\ &+ \sum_i \|g_i(v^k) - g_i(v^*)\|^2 - \omega_2^2 \\ &= \|h(v^k) - h(v^*)\|^2 - \omega_2^2 \\ &\leq \|v^k - v^*\|^2 - \omega_2^2. \end{aligned} \tag{37}$$

Therefore, for $i \in L_2$, it holds that $\mathbf{w}_i^k = \mathbf{w}_i^* = 0$ in no more than $\|v^0 - v^*\|^2/\omega_2^2$ iterations.

Similarly, for any $i \in \{1, \ldots, N\}$, from (11) we have

$$\begin{aligned} \left(z_i^{k+1} - z_i^*\right)^2 &= \left(s_1 \circ h_i^{(1)}(v^k) - s_1 \circ h_i^{(1)}(v^*)\right)^2 \\ &\leq \left|h_i^{(1)}(v^k) - h_i^{(1)}(v^*)\right|^2. \end{aligned} \tag{38}$$

Suppose $z_i^{k+1} \neq 0$ for some $i \in L_1$, similar discussion as in (35) gives

$$\left(z_i^{k+1} - z_i^*\right)^2 \leq \left|h_i^{(1)}(v^k) - h_i^{(1)}(v^*)\right|^2 - \omega_1^2, \tag{39}$$

where ω_1 is defined in (28). Combining (34), (38) and (39), similar arguments as in (37) gives

$$\|v^{k+1} - v^*\|^2 \leq \|v^k - v^*\|^2 - \omega_1^2. \tag{40}$$

Therefore, $z_i^k = z_i^* = 0$ in no more than $\|v^0 - v^*\|^2/\omega_1^2$ iterations for $i \in L_1$.

APPENDIX B
PROOF OF THEOREM 3.5

From (24)–(27), there hold

$$u^{k+1} - u^* = M^{-1}H^\top(v^{k+1} - v^*) \tag{41}$$

and

$$\begin{aligned} & \|v^{k+1} - v^*\|^2 \\ = & \|z^{k+1} - z^*\|^2 + \|w^{k+1} - w^*\|^2 \\ = & \|S_1(\Psi^\top u^k) - S_1(\Psi^\top u^*)\|^2 \\ + & \|S_2(D^{(1)}u^k; D^{(2)}u^k) - S_2(D^{(1)}u^*; D^{(2)}u^*)\|^2 \\ \leq & \|\Psi^\top(u^k - u^*)\|^2 + \|D(u^k - u^*)\|^2 \\ = & \|H(u^k - u^*)\|^2. \end{aligned} \tag{42}$$

Combining the recursion (41), (42) and the definition of T, it holds

$$\|v^{k+1} - v^*\|^2 \leq \|HM^{-1}H^\top(v^k - v^*)\|^2 = \|T(v^k - v^*)\|^2.$$

Since we are only interested in the asymptotic behavior of Algorithm 1, without loss of generality, we assume that $v_L^k = v_L^* = 0$. The above inequality becomes

$$\begin{aligned} \|v_E^{k+1} - v_E^*\|^2 &\leq (v_E^k - v_E^*)^\top (T^2)_{EE}(v_E^k - v_E^*) \\ &\leq \rho((T^2)_{EE})\|v_E^k - v_E^*\|^2, \end{aligned}$$

which implies assertion 1 of this theorem. Multiplying H on both sides of (41), from $v_L^k = 0$ and (42), we get

$$\begin{aligned} \|H(u^{k+1} - u^*)\|^2 &\leq (v_E^{k+1} - v_E^*)^\top (T^2)_{EE}(v_E^{k+1} - v_E^*) \\ &\leq \rho((T^2)_{EE})\|H(u^k - u^*)\|^2, \end{aligned}$$

which completes the proof of Theorem 3.5.

ACKNOWLEDGMENT

The work of J. Yang has been supported by the Chinese Scholarship Council during his visit to Rice University. The work of Y. Zhang has been supported in part by NSF Grant DMS-0811188 and ONR Grant N00014-08-1-1101. The work of W. Yin has been supported in part by NSF CAREER Award DMS-0748839 and ONR Grant N00014-08-1-1101.

REFERENCES

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

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

[3] 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, 2007.

[4] E. Candès, and J. Romberg, *Practical signal recovery from random projections*, Wavelet Applications in Signal and Image Processing XI, Proc. SPIE Conf. 5914, 2004

[5] E. Candès, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate information*, Communications on Pure and Applied Mathematics, vol. 59, pp. 1207-1233, 2005.

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

[7] E. Candès, and T. Tao, *Decoding by linear programming*, IEEE Trans. Inform. Theory, vol. 51, no. 12, pp. 4203–4215, 2005.

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

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

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

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

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

[13] D. Donoho, *For Most Large Underdetermined Systems of Linear Equations, the minimal l1-norm solution is also the sparsest solution*, Communications on pure and applied mathematics, vol. 59, no. 7, pp. 907–934, 2006.

[14] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization*, IEEE Trans. Image Process., vol. 4, pp. 932–946, 1995.

[15] 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 CAM Report 06-35, 2006.

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

[17] S. MA, W. YIN, Y. ZHANG, AND A. CHAKRABORTY, *An efficient algorithm for compressed MR imaging using total variation and wavelets*, IEEE Conference on CVPR, pp. 1–8, 2008.

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

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

[20] 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.

[21] J. Yang, W. Yin, Y. Zhang, and Y. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, SIAM Journal on Imaging Sciences, to appear.

[22] J. Yang, Y. Zhang, and W. Yin, *An efficient TVL1 algorithm for deblurring of multichannel images corrupted by impulsive noise*, TR08-12, CAAM, Rice University, Submitted to SISC.

[23] Y. Zhang, *On theory of compressive sensing via ℓ_1-minimization: simple derivations and extensions*, TR08-11, CAAM, Rice University. Submitted to SIREV.