

---

# A Highly Scalable Parallel Algorithm for Isotropic Total Variation Models

---

Jie Wang<sup>1</sup>

Qingyang Li<sup>1</sup>

Sen Yang<sup>1</sup>

Wei Fan<sup>2</sup>

Peter Wonka<sup>3,1</sup>

Jieping Ye<sup>1</sup>

JIE.WANG.USTC@ASU.EDU

QINGYANG.LI@ASU.EDU

SENYANG@ASU.EDU

DAVID.FANWEI@HUAWEI.COM

PWONKA@GMAIL.COM

JIEPING.YE@ASU.EDU

<sup>1</sup>Arizona State University, Tempe, AZ 85287 USA

<sup>2</sup>Huawei Noahs Ark Lab, Hong Kong, China

<sup>3</sup>King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

## Abstract

Total variation (TV) models are among the most popular and successful tools in signal processing. However, due to the complex nature of the TV term, it is challenging to efficiently compute a solution for large-scale problems. State-of-the-art algorithms that are based on the alternating direction method of multipliers (ADMM) often involve solving large-size linear systems. In this paper, we propose a highly scalable parallel algorithm for TV models that is based on a novel decomposition strategy of the problem domain. As a result, the TV models can be decoupled into a set of small and independent subproblems, which admit closed form solutions. This makes our approach particularly suitable for parallel implementation. Our algorithm is guaranteed to converge to its global minimum. With  $N$  variables and  $n_p$  processes, the time complexity is  $O(\frac{N}{\epsilon n_p})$  to reach an  $\epsilon$ -optimal solution. Extensive experiments demonstrate that our approach outperforms existing state-of-the-art algorithms, especially in dealing with high-resolution, mega-size images.

## 1. Introduction

Total variation (TV) models have found great success in a wide range of applications, including but not limited to image denoising, image deblurring, image reconstruction (Barbero & Sra, 2011; Candès et al., 2006; Casas et al., 1999; He et al., 2005; Kim et al., 2010; Lustig et al., 2005;

Osher et al., 2005; Vert & Bleakley, 2010; Vogel & Oman, 1998) etc. Rudin et al. (1992) first introduced the TV model to remove noise in a given image by using the TV term. The motivation for this model came from the fact that a noisy signal generally implies high total variation. Given an image  $Y \in \mathbb{R}^{m \times n}$ , the discrete version of the Rudin-Osher-Fatemi (ROF) model is:

$$\min_X \frac{1}{2} \|X - Y\|_F^2 + \lambda \|X\|_{TV}, \quad (1)$$

where  $\|\cdot\|_F$  and  $\|\cdot\|_{TV}$  are the Frobenius norm and TV norm respectively,  $\lambda$  is a positive parameter, and  $X \in \mathbb{R}^{m \times n}$  is the image to be recovered. In this paper, we consider the isotropic TV norm:

$$\|X\|_{TV} = \sum_{i=1}^m \sum_{j=1}^n \|D_{i,j}X\|_2, \quad (2)$$

where  $D_{i,j}X = ((D_1X)_{i,j}, (D_2X)_{i,j})^T$  is the discretized gradient at pixel  $(i, j)$ . For simplicity, we use the forward difference to define  $D_{i,j}X$ , i.e.,

$$(D_1X)_{i,j} = \begin{cases} x_{i+1,j} - x_{i,j} & \text{if } i < m \\ 0 & \text{if } i = m \end{cases}, \quad (3)$$

$$(D_2X)_{i,j} = \begin{cases} x_{i,j+1} - x_{i,j} & \text{if } j < n \\ 0 & \text{if } j = n \end{cases}. \quad (4)$$

It is worthwhile to mention that there are fast algorithms for the “anisotropic” TV model, where  $\|X\|_{TV}$  is defined by  $\sum_{i=1}^m \sum_{j=1}^n \|D_{i,j}X\|_1$ . However, those algorithms, which are usually based on graph cuts or maximum flow, are not applicable to isotropic TV models. As pointed out by Goldfarb & Yin (2009); Duan & Tai (2012), unless  $x_{i,j}$  are binary variables, the isotropic TV term is not graph representable. Moreover, Wahlberg et al. (2012) proposed an ADMM method to solve the 1D TV model. Recently, Condat (Condat, 2013) showed that the 1D TV denoising model admits a closed-form solution and thus can be solved by

a very fast noniterative algorithm. By utilizing this result, Yang et al. (2013) proposed an efficient ADMM algorithm for the multidimensional TV model. However, this method is only applicable to the anisotropic TV model.

Although the formulation in problem (1) is simple, it is computationally challenging to solve due to the complex nature of the TV norm. This makes it difficult to work with high-resolution, large-scale data. Existing algorithms usually demand high computational cost and memory usage, which prevent their use in large-scale problems. Simpler methods require moderate computational efforts and less memory space, but usually converge very slowly. These difficulties have motivated a lot of research efforts to better trade off the computational cost and the convergence rate. For example, Beck & Teboulle (2009a) propose fast gradient-based algorithms that combine an acceleration of the dual approach with a fast iterative shrinkage algorithm (FISTA) (Beck & Teboulle, 2009b); Goldstein & Osher (2009) develop a split Bregman method; Chambolle & Pock (2011) apply a primal-dual method to solve (1). As another popular application of TV-based models, the Magnetic Resonance (MR) image reconstruction has received great attention recently. Ma et al. (2008) propose an operator-splitting algorithm (TVCMRI) for MR image reconstruction. Huang et al. (2011) propose another efficient algorithm named FCSA, which utilizes the ideas of composite splitting (Combettes & Pesquet, 2011) and the acceleration technique introduced by FISTA.

In this paper, we propose a fast Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011; Chen et al., 2012; Chan et al., 2011; Esser et al., 2010; Esser, 2009) based algorithm, termed FAD, to solve the TV models. FAD is based on a novel decomposition strategy of the problem domain. As a result, the TV model in (1) can be decoupled into a set of small and independent subproblems. Each subproblem involves at most three variables and admits a closed-form solution. Thus, all of the subproblems can be solved efficiently in parallel. We implement our highly scalable parallel algorithm via OpenMP and MPI. Extensive experiments on synthetic and real data demonstrate the scalability and efficiency of FAD.

**Notation:** Let  $\|\cdot\|$  be the  $\ell_2$  norm. For matrix  $A$ ,  $[A]_{i,j}$  is its  $(i,j)^{th}$  entry. The inner product between matrices is defined as  $\langle A, B \rangle = \sum_{i,j} [A]_{i,j} [B]_{i,j}$ . Given a set  $\mathcal{C}$ , we use  $\mathcal{C}^\circ$ ,  $\text{relint } \mathcal{C}$ , and  $\partial \mathcal{C}$  to denote the interior points, relative interior and boundary points of  $\mathcal{C}$ , respectively. For any real number  $c$ , let  $d := \lfloor c \rfloor$  be the largest integer such that  $d \leq c$ , and  $\text{mod}(a, b) := a - \lfloor \frac{a}{b} \rfloor b$ . Let  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  be a closed proper convex function. The proximal operator of  $f$  is defined by:

$$\text{prox}_f(\mathbf{x}) = \underset{\mathbf{y}}{\text{argmin}} \left( f(\mathbf{y}) + \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 \right).$$

## 2. Optimization Methods

In this section, we present the details of FAD. We first briefly review ADMM in Section 2.1. In Section 2.2, we present our novel decomposition strategy to partition the variables in (1). FAD and its convergence properties are presented in Sections 2.3 and 2.4, respectively.

### 2.1. Review of ADMM

ADMM makes use of the decomposability of the dual ascent and meanwhile has the good convergence properties of the multipliers' method. Given a problem:

$$\min_{\mathbf{x}, \mathbf{z}} \{f(\mathbf{x}) + g(\mathbf{z}) : \mathbf{x} - \mathbf{z} = 0\}, \quad (5)$$

we assume that both  $f(\cdot)$  and  $g(\cdot)$  are convex. The augmented Lagrangian is given by:

$$L_\gamma(\mathbf{x}, \mathbf{z}; \theta) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\gamma}{2} \|\mathbf{x} - (\mathbf{z} - \theta)\|^2, \quad (6)$$

where  $\theta$  and  $\gamma > 0$  are the scaled augmented Lagrangian multipliers (Boyd et al., 2011) and the penalty parameter, respectively. ADMM attempts to solve problem (5) by iteratively minimizing  $L_\gamma(\mathbf{x}, \mathbf{y}; \theta)$  over  $\mathbf{x}$  and  $\mathbf{y}$ , respectively, and updating  $\theta$  accordingly (Boyd et al., 2011). The supplement presents details on ADMM.

### 2.2. The Decomposition Strategy

We first give an example to illustrate our idea in Section 2.2.1 and then consider the general case in Section 2.2.2.

#### 2.2.1. A SIMPLE EXAMPLE

Let us first consider a simple case as an example. Suppose that we have an image of  $4 \times 4$  pixels as illustrated in Fig. 1. The TV term  $\|X\|_{TV}$  can be written as:

$$\begin{aligned} \|X\|_{TV} &= \sum_{i=1}^4 \sum_{j=1}^4 \|D_{i,j} X\| \\ &= \sum_{i=1}^3 \sum_{j=1}^3 \sqrt{(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2} \\ &\quad + \sum_{i=1}^3 |x_{i+1,4} - x_{i,4}| + \sum_{j=1}^3 |x_{4,j+1} - x_{4,j}|. \end{aligned} \quad (7)$$

From the right-hand side (RHS) of Eq. (7), we can see

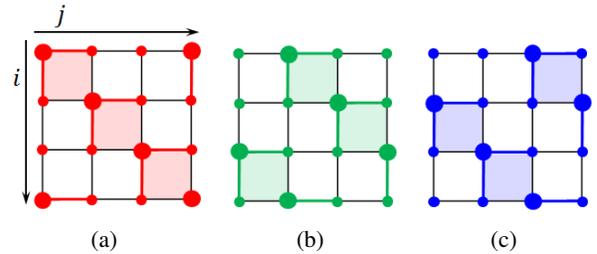


Figure 1. Illustration of the decomposition.

that none of the terms is separable from the others since most of the variables appear in different terms. Our goal

is to decompose  $\|X\|_{TV}$  into several different parts such that each term is separable from all the others. As shown in Fig. 1, we decompose  $\|X\|_{TV}$  into three disjoint parts, i.e.,

$$\|X\|_{TV} = \sum_{k=1}^3 \|X\|_{TV_k}. \quad (8)$$

Each part consists of the total variation norm  $\|D_{i,j}X\|$  associated with the big solid nodes. For example, there are six big solid red nodes in Fig. 1(a): (1, 1), (2, 2), (3, 3), (4, 4), (1, 4) and (4, 1). Thus, the first part  $\|X\|_{TV_1}$  is:

$$\begin{aligned} \|X\|_{TV_1} &= \sum_{k=1}^4 \|D_{k,k}X\| + \|D_{1,4}X\| + \|D_{4,1}X\| \\ &= \sum_{k=1}^3 \sqrt{(x_{k+1,k} - x_{k,k})^2 + (x_{k,k+1} - x_{k,k})^2} \\ &\quad + |x_{2,4} - x_{1,4}| + |x_{4,2} - x_{4,1}|. \end{aligned}$$

Recall that  $\|D_{4,4}X\| = 0$ . From the RHS of the above equation, we can see that each variable only appears in one term, which implies that every term is separable from all the other ones. Similarly, from Fig. 1(b) and Fig. 1(c),  $\|X\|_{TV_2}$  and  $\|X\|_{TV_3}$  are given by

$$\begin{aligned} \|X\|_{TV_2} &= \sum_{k=1}^3 \|D_{k,k+1}X\| + \sum_{k=1}^2 \|D_{k+2,k}X\|, \\ \|X\|_{TV_3} &= \sum_{k=1}^3 \|D_{k+1,k}X\| + \sum_{k=1}^2 \|D_{k,k+2}X\|. \end{aligned}$$

Clearly, for  $\|X\|_{TV_2}$ , each term on the RHS of the equation is separable from all the other ones and the same property holds for  $\|X\|_{TV_3}$ .

### 2.2.2. THE GENERAL CASE

Suppose that we have an image of  $m \times n$  pixels. Let  $\mathcal{D} = \{(i, j) : i = 1, 2, \dots, m, j = 1, 2, \dots, n\}$ . Then, we can divide  $\mathcal{D}$  into three non-overlapping subsets:

$$\mathcal{D}_k = \{(i, j) \in \mathcal{D} : \text{mod}(j - i, 3) = k - 1\}, k = 1, 2, 3.$$

Thus,  $\|X\|_{TV}$  can be written as:

$$\|X\|_{TV} = \sum_{k=1}^3 \|X\|_{TV_k} = \sum_{k=1}^3 \sum_{(i,j) \in \mathcal{D}_k} \|D_{i,j}X\|. \quad (9)$$

For each  $\|X\|_{TV_k} = \sum_{(i,j) \in \mathcal{D}_k} \|D_{i,j}X\|$  ( $k = 1, 2, 3$ ), every term  $\|D_{i,j}X\|$ ,  $(i, j) \in \mathcal{D}_k$  is separable from all the other ones  $\|D_{i',j'}X\|$  where  $(i', j') \in \mathcal{D}_k \setminus (i, j)$ . For details, please refer to Proposition 1 in the supplement.

**Remark 1.** *The isotropic TV norm introduced in Eq. (2) is the summation of the  $\ell_2$  norm of the discretized gradient over all pixels. The proposed decomposition strategy is to decompose the image into three non-overlapping sets of pixels, and write the TV norm as the summation of three terms as in Eq. (9). Each term refers to the summation of the  $\ell_2$  norm of the discretized gradient over one group of pixels. Notice that each  $\|X\|_{TV_k} = \sum_{(i,j) \in \mathcal{D}_k} \|D_{i,j}X\|$  also involves the pixel-level variables  $[X]_{i',j'}$  that are shared across the three groups (though not within the groups).*

### 2.3. ADMM for TV-Based Models

By using the decomposition strategy presented in the last section, problem (1) can be rewritten as:

$$\min_X \frac{1}{2} \|X - Y\|_F^2 + \lambda \sum_{k=1}^3 \|X\|_{TV_k}. \quad (10)$$

To use ADMM, (10) can be reformulated as:

$$\begin{aligned} \min_{Z; X_1, X_2, X_3} \quad & \frac{1}{2} \|Z - Y\|_F^2 + \lambda \sum_{k=1}^3 \|X_k\|_{TV_k}, \\ \text{s.t.} \quad & X_k = Z, k = 1, 2, 3, \end{aligned} \quad (11)$$

with global variable  $Z$  and local variables  $X_k$ ,  $k = 1, 2, 3$ . The augmented Lagrangian of problem (11) is:

$$L_\gamma(Z; X_1, X_2, X_3; \Theta_1, \Theta_2, \Theta_3) \quad (12)$$

$$= \frac{1}{2} \|Z - Y\|_F^2 + \lambda \sum_{k=1}^3 \|X_k\|_{TV_k} + \frac{\gamma}{2} \sum_{k=1}^3 \|X_k - (Z - \Theta_k)\|_F^2.$$

Denote  $X_1, X_2, X_3$  and  $\Theta_1, \Theta_2, \Theta_3$  by  $\mathbf{X}$  and  $\Theta$ , respectively. Thus,  $L_\gamma$  in Eq. (12) can be simplified to  $L_\gamma(Z; \mathbf{X}; \Theta)$ . We apply ADMM to solve (11). In each iteration, we first minimize  $L_\gamma(Z; \mathbf{X}; \Theta)$  with respect to  $Z$  and  $\mathbf{X}$ , and then update the dual variable  $\Theta$ . Assume that we are at the  $t^{\text{th}}$  step, i.e.,  $Z^t$ ,  $\mathbf{X}^t$  and  $\Theta^t$  are known. We then compute  $Z^{t+1}$ ,  $\mathbf{X}^{t+1}$  and  $\Theta^{t+1}$  as follows:

#### S1. Minimize $L_\gamma(Z^t; \mathbf{X}; \Theta^t)$ with respect to $\mathbf{X}$

Recall that  $\mathbf{X}$  refers to  $X_k$ ,  $k = 1, 2, 3$ . Thus, we need to minimize  $L_\gamma(Z^t; \mathbf{X}; \Theta^t)$  with respect to  $X_k$ ,  $k = 1, 2, 3$ .

Since  $X_k^{t+1} = \text{argmin}_{X_k} L_\gamma(Z^t; \mathbf{X}; \Theta^t)$ , we have

$$\begin{aligned} X_k^{t+1} &= \text{argmin}_{X_k} \lambda \|X_k\|_{TV_k} + \frac{\gamma}{2} \|X_k - (Z^t - \Theta_k^t)\|_F^2 \\ &= \text{argmin}_{X_k} \frac{1}{2} \|X_k - V_k^t\|_F^2 + \frac{\lambda}{\gamma} \|X_k\|_{TV_k}, \end{aligned} \quad (13)$$

where  $V_k^t = Z^t - \Theta_k^t$ . Note that, the first term on the RHS of (13) is separable. Due to the decomposition strategy in Section 2.2, the second term on the RHS of (13) is separable as well. Therefore, problem (13) can be decoupled into a set of subproblems that involve at most three variables.

Intuitively, the reason is that each subproblem involves a given pixel as well as the pixels above and to the right due to the forward difference. Suppose that pixel  $(p, q) \in \mathcal{D}_k$ . We have the following cases.

Case 1. Image interior (if  $p < m, q < n$ ). Let us define

$$\begin{aligned} \mathbf{u} &= ([X_k]_{p,q+1}, [X_k]_{p,q}, [X_k]_{p+1,q})^T, \\ \mathbf{w} &= ([V_k^t]_{p,q+1}, [V_k^t]_{p,q}, [V_k^t]_{p+1,q})^T. \end{aligned} \quad (14)$$

As we discussed in Section 2.2,  $\mathbf{u}$  is only involved in  $\|D_{p,q}X_k\|$ , not any other terms in  $\|X_k\|_{TV_k}$ . Thus, the following subproblem can be decoupled from problem (13):

$$\min_{\mathbf{u}} \frac{1}{2} \|\mathbf{u} - \mathbf{w}\|^2 + \frac{\lambda}{\gamma} \sqrt{(u_1 - u_2)^2 + (u_3 - u_2)^2}. \quad (15)$$

Case 2. Top boundary (if  $p = m, q < n$ ). Let

$$\mathbf{u} = ([X_k]_{p,q+1}, [X_k]_{p,q})^T, \mathbf{w} = ([V_k^t]_{p,q+1}, [V_k^t]_{p,q})^T. \quad (16)$$

Then, due to the same reason as in Case 1, we have the following subproblem that can be decoupled from (13):

$$\min_{\mathbf{u}} \frac{1}{2} \|\mathbf{u} - \mathbf{w}\|^2 + \frac{\lambda}{\gamma} |u_2 - u_1|. \quad (17)$$

Consider the right boundary ( $p < m, q = n$ ). Let

$$\mathbf{u} = ([X_k]_{p+1,q}, [X_k]_{p,q})^T, \mathbf{w} = ([V_k^t]_{p+1,q}, [V_k^t]_{p,q})^T. \quad (18)$$

We have the same subproblem as (17).

Case 3. Top right corner (if  $p = m$  and  $q = n$ ). We have  $\|D_{p,q} X_k\| = 0$  by the definitions in (3) and (4). Thus,

$$\min_{[X_k]_{p,q}} \frac{1}{2} ([X_k]_{p,q} - [V_k^t]_{p,q})^2 \Rightarrow [X_k^{t+1}]_{p,q} = [V_k^t]_{p,q}. \quad (19)$$

We solve problems (15) and (17) in Section 3.

**Remark 2.**  $[X_k]_{p,q}$  will be updated with one of the above three cases if group  $k$  contains pixel  $(p, q)$ , the pixel below, or the pixel to the left.

Case 4. By Remark 2, if none among the pixels  $(p, q)$ , the pixel below, or the pixel to the left belongs to group  $k$ ,  $[X_k]_{p,q}$  will not be updated. Take Fig. 1(a) as an example; this is the case for pixel  $(1, 3)$ . Clearly, the variable  $[X_k]_{p,q}$  does not appear in the second term on the RHS of (13), i.e.,  $\frac{\lambda}{\gamma} \|X_k\|_{TV_k}$ . Thus, we only need to solve:

$$\min_{[X_k]_{p,q}} \frac{1}{2} ([X_k]_{p,q} - [V_k^t]_{p,q})^2 \Rightarrow [X_k^{t+1}]_{p,q} = [V_k^t]_{p,q}. \quad (20)$$

**Remark 3.** **S1** involves two levels of decomposition. The first level comes from the fact that problem (13) involves minimizing over the three sets of local variables,  $X_k, k = 1, 2, 3$ , which are independent from each other. Thus, we can solve (13) for  $X_k, k = 1, 2, 3$ , in parallel. The second level is due to the four cases in **S1**. Specifically, for each  $k \in \{1, 2, 3\}$ , (13) breaks into a set of subproblems involving at most three variables [problems (15) and (17)]. Recall that, in Section 2.2, we decompose the image into three groups of non-overlapping pixels. We emphasize that these are referring to different levels of the decomposition.

### S2. Minimize $L_\gamma(Z; \mathbf{X}^{t+1}; \Theta^t)$ with respect to $Z$

Since  $Z^{t+1} = \operatorname{argmin}_Z L_\gamma(Z; \mathbf{X}^{t+1}; \Theta^t)$ , we have

$$\left. \frac{\partial L_\gamma(Z; \mathbf{X}^{t+1}; \Theta^t)}{\partial Z} \right|_{Z=Z^{t+1}} = 0 \quad (21)$$

$$\begin{aligned} \Rightarrow Z^{t+1} - Y - \gamma \sum_{k=1}^3 (X_k^{t+1} - Z + \Theta_k^t) &= 0 \\ \Rightarrow Z^{t+1} &= \frac{Y + \sum_{k=1}^3 (\Theta_k^t + X_k^{t+1})}{1 + 3\gamma}. \end{aligned} \quad (22)$$

Notice that, in this step, we update the global variable  $Z$  and thus the replicated variables  $X_k, k = 1, 2, 3$ , are coordinated across the three pixel groups.

### S3. Update the (scaled) dual variable $\Theta^t$

The dual variables  $\Theta_k^t, k = 1, 2, 3$ , are updated by

$$\Theta_k^{t+1} = \Theta_k^t + (X_k^{t+1} - Z^{t+1}), \quad (23)$$

which is equivalent to a dual ascent step (Boyd et al., 2011).

Our method is summarized in Algorithm 1.

---

#### Algorithm 1 FAD (Fast ADMM for TV Models)

---

**Input:**  $Y, \lambda, \gamma > 0$

**Initialize**  $Z^0 = X_k^0 := Y, \Theta_k^0 := 0, \quad k = 1, 2, 3$

**for**  $t = 0$  **to**  $T$  **do**

**for**  $k = 1$  **to**  $3$  **do**

$$V_k^t := Z^t - \Theta_k^t,$$

$$X_k^{t+1} := \operatorname{argmin}_{X_k} \frac{1}{2} \|X_k - V_k^t\|_F^2 + \frac{\lambda}{\gamma} \|X_k\|_{TV_k},$$

**end for**

$$Z^{t+1} := \frac{Y + \sum_{k=1}^3 (\Theta_k^t + X_k^{t+1})}{1 + 3\gamma},$$

**for**  $k = 1$  **to**  $3$  **do**

$$\Theta_k^{t+1} := \Theta_k^t + (X_k^{t+1} - Z^{t+1}),$$

**end for**

**end for**

**Return:**  $Z^T$

---

**Remark 4.** Recall that  $T$  is the number of iterations when Algorithm 1 terminates. We use the primal and dual residuals (Boyd et al., 2011) to specify the stopping criterion. If both residuals are smaller than a given parameter  $\epsilon$ , the algorithm stops. In this paper, we set  $\epsilon = 10^{-4}$ . In addition, it is known that a larger  $\gamma$  tends to result in a smaller primal residual but a larger dual residual. We keep  $\gamma$  fixed ( $\gamma = 10$ ) in this paper. For schemes about varying  $\gamma$ , we refer readers to (Boyd et al., 2011).

## 2.4. Convergence Analysis

The convergence properties of ADMM that solve the standard form in (5) have been extensively explored by (He & Yuan, 2012; Boyd et al., 2011; Monteiro & Svaiter, 2010; Eckstein & Bertsekas, 1992; Glowinski & Tallec, 1989). To establish the convergence properties of Algorithm 1, we can reformulate problem (11) as (5), and the resulting formulation satisfies the conditions required for convergence. Moreover, the convergence rate of Algorithm 1 can be shown as  $O(1/k)$  by following the procedure in (He & Yuan, 2012). For details, see the supplement.

## 3. Solutions to the Subproblems

We solve problems (15) and (17) in Sections 3.1 and 3.2, respectively. In Section 3.3, we give a brief complexity analysis of Algorithm 1.

### 3.1. Solution to Subproblem (15)

We focus on the following subproblem:

$$\min_{\mathbf{u}} \frac{1}{2} \|\mathbf{u} - \mathbf{w}\|^2 + \rho \sqrt{(u_1 - u_2)^2 + (u_3 - u_2)^2},$$

where  $\rho = \frac{\lambda}{\gamma} \geq 0$ . Let  $G \in \mathbb{R}^{2 \times 3}$  be defined by

$$G = \begin{pmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}. \quad (24)$$

The dual problem of (15) is:

$$\min_{\mathbf{s}} \left\{ \frac{1}{2} \|\mathbf{w} - \rho G^T \mathbf{s}\|^2 : s_1^2 + s_2^2 \leq 1 \right\}. \quad (25)$$

Assume that  $\mathbf{u}^*$  and  $\mathbf{s}^*$  are the optimal solutions of (15) and (25). By the KKT conditions, we have:

$$\mathbf{u}^* = \mathbf{w} - \rho G^T \mathbf{s}^*, \quad (26)$$

$$\begin{pmatrix} s_1^* \\ s_2^* \end{pmatrix} \in \begin{cases} \frac{(u_2^* - u_1^*, u_3^* - u_2^*)^T}{\|(u_2^* - u_1^*, u_3^* - u_2^*)^T\|}, & \text{if } (u_2^* - u_1^*, u_3^* - u_2^*)^T \neq 0, \\ \mathbf{v}, \|\mathbf{v}\| \leq 1, & \text{if } (u_2^* - u_1^*, u_3^* - u_2^*)^T = 0. \end{cases} \quad (27)$$

Note that problem (25) is smooth and convex. It is thus easier to solve than the nonsmooth problem (15). We can solve  $\mathbf{u}^*$  via  $\mathbf{s}^*$  according to Eq. (26). We rewrite (25) as:

$$\min_{\mathbf{s}} \left\{ \frac{\rho^2}{2} \left\| \frac{\mathbf{w}}{\rho} - G^T \mathbf{s} \right\|^2 : s_1^2 + s_2^2 \leq 1 \right\}. \quad (28)$$

Let  $\tilde{\mathcal{B}} = \{\mathbf{t} : \mathbf{t} = G^T \mathbf{s}, \mathbf{s} \in \mathcal{B}\}$ , where  $\mathcal{B} \in \mathbb{R}^2$  is the unit disc. Clearly,  $\tilde{\mathcal{B}} \in \mathbb{R}^3$  is a two dimensional linear manifold. Indeed, problem (28) is a projection problem. Let

$$\mathbf{t}^*(\rho) = \operatorname{argmin}_{\mathbf{t} \in \tilde{\mathcal{B}}} \left\| \frac{\mathbf{w}}{\rho} - \mathbf{t} \right\|^2 = P_{\tilde{\mathcal{B}}}(\mathbf{w}/\rho).$$

Then, we have  $\mathbf{t}^*(\rho) = G^T \mathbf{s}^*(\rho)$ . There is a one-to-one linear mapping between  $\mathbf{t}^*(\rho)$  and  $\mathbf{s}^*(\rho)$  since  $G^T$  is linear and has full column rank. Consequently,  $\mathbf{t}^*(\rho) \in \operatorname{relint} \tilde{\mathcal{B}}$  implies that  $\mathbf{s}^*(\rho) \in \mathcal{B}^\circ$  and  $\mathbf{t}^*(\rho) \in \partial \tilde{\mathcal{B}}$  implies that  $\mathbf{s}^*(\rho) \in \partial \mathcal{B}$  and vice versa. Since the projection operator  $P_{\tilde{\mathcal{B}}}(\mathbf{w}/\rho)$  is nonexpansive (Bertsekas, 2003),  $\mathbf{t}^*(\rho)$  varies continuously with  $\rho$ . When  $\rho$  is large enough,  $\mathbf{t}^*(\rho)$  must belong to  $\operatorname{relint} \tilde{\mathcal{B}}$  and thus  $\|\mathbf{s}^*\| \in \mathcal{B}^\circ$  (consider the extreme case  $\mathbf{t}^*(\infty) = 0 \in \operatorname{relint} \tilde{\mathcal{B}}$ ). Therefore, there must exist a  $\rho_{max}$  such that  $\rho > \rho_{max}$  implies  $\mathbf{s}^*(\rho) \in \mathcal{B}^\circ$ , i.e.,  $\|\mathbf{s}^*(\rho)\| < 1$ . By Eq. (27),  $\|\mathbf{s}^*(\rho)\| < 1$  implies that all components of  $\mathbf{u}^*(\rho)$  are equal. The following theorem solves for  $\rho_{max}$  and  $\mathbf{u}^*$  explicitly when  $\rho \geq \rho_{max}$ .

**Theorem 1.** Let  $\rho_{max} = \|(GG^T)^{-1}G\mathbf{w}\|$  and  $I$  be the identity matrix. If  $\rho \geq \rho_{max}$ , the optimal solution of the problem (15) is given by

$$\mathbf{u}^* = [I - G^T(GG^T)^{-1}G] \mathbf{w}. \quad (29)$$

**Remark 5.** The expression of  $\rho_{max}$  implies that if  $\mathbf{w}$  belongs to the null space of  $G$ , i.e., all three components of  $\mathbf{w}$  are equal, then  $\rho_{max} = 0$ . In this case, the solution to problem (15) is  $\mathbf{u}^* = \mathbf{w}$  [note that  $\mathbf{u}^*$  is the projection of  $\mathbf{w}$  onto the null space of  $G$  by Eq. (29)], which is independent of  $\rho$ . To avoid this trivial case, we always assume that  $\mathbf{w}$  is not a constant signal. Therefore, we consider only the case with  $\rho_{max} > 0$ .

We solve for  $\mathbf{u}^*$  with  $\rho < \rho_{max}$  in the following theorem.

**Theorem 2.** Let  $G = U\Sigma V^T$  be the singular value decom-

position of  $G$ . If  $\rho < \rho_{max}$ , the optimal solution to problem (15) is given by

$$\mathbf{u}^* = \left[ I - G^T(GG^T + \frac{\alpha}{\rho^2}I)^{-1}G \right] \mathbf{w}, \quad (30)$$

where  $\alpha > 0$  solves the following quartic equation:

$$\alpha^4 + c_3\alpha^3 + c_2\alpha^2 + c_1\alpha + c_0 = 0. \quad (31)$$

The parameters are given by

$$\begin{aligned} c_0 &= \rho^8 \sigma_1^4 \sigma_2^4 - \rho^6 (\tilde{w}_1^2 \sigma_2^4 + \tilde{w}_2^2 \sigma_1^4), \\ c_1 &= 2\rho^6 (\sigma_1^2 \sigma_2^4 + \sigma_1^4 \sigma_2^2) - 2\rho^4 (\tilde{w}_1^2 \sigma_2^2 + \tilde{w}_2^2 \sigma_1^2), \\ c_2 &= \rho^4 (\sigma_1^4 + \sigma_2^4 + 4\sigma_1^2 \sigma_2^2) - \rho^2 (\tilde{w}_1^2 + \tilde{w}_2^2), \\ c_3 &= 2\rho^2 (\sigma_1^2 + \sigma_2^2), \end{aligned}$$

and  $\tilde{\mathbf{w}} = \Sigma V^T \mathbf{w}$ ,  $\sigma_1^2 = 3$ ,  $\sigma_2^2 = 1$ . Eq. (31) has a unique positive root if  $\rho < \rho_{max}$ .

**Remark 6.** There are numerous efficient algorithms<sup>1</sup> to solve Eq. (31). Closed-form solutions of quartic equations are also available<sup>2</sup>, but they are computationally expensive and nonstable. In this paper, we use Newton's method to find the nonnegative root of Eq. (31), which never fails in practice. For discussions regarding the root finding techniques, we refer the reader to the supplement.

### 3.2. Solution to Subproblem (17)

Recall that problem (17) takes the form of

$$\min_{\mathbf{u}} h(\mathbf{u}) = \frac{1}{2} \|\mathbf{u} - \mathbf{w}\|^2 + \rho |u_2 - u_1|,$$

where  $\rho = \lambda/\gamma$ . We have the following theorem.

**Theorem 3.** Let  $\mathbf{u}^*$  be the optimal solution of (17). Then,

$$\mathbf{u}^* = \begin{cases} (w_1 - \rho, w_2 + \rho)^T, & \text{if } w_1 > w_2 + 2\rho, \\ (w_1 + \rho, w_2 - \rho)^T, & \text{if } w_1 < w_2 - 2\rho, \\ (\frac{w_1 + w_2}{2}, \frac{w_1 + w_2}{2})^T, & \text{otherwise.} \end{cases} \quad (32)$$

Notice that the result in Theorem 3 is a small variation on the standard soft thresholding operator.

**Remark 7.** Notice that problems (15) and (17) can be viewed as proximal operators of particular functions. Take problem (15) for example. Let  $f(\mathbf{u}) = \|\mathbf{G}\mathbf{u}\|$ . Then, the optimal solution  $\mathbf{u}^*$  of problem (15) can be expressed as:

$$\mathbf{u}^* = \operatorname{prox}_{\rho f}(\mathbf{w}).$$

Moreover, Fenchel's conjugate (Ruszczynski, 2006) of  $\rho f$  is given by:

$$(\rho f)^*(\mathbf{v}) = \sup_{\mathbf{u}} (\mathbf{u}^T \mathbf{v} - \rho f(\mathbf{u})) = \delta_{\rho \tilde{\mathcal{B}}}(\mathbf{v}),$$

where  $\delta_{\rho \tilde{\mathcal{B}}}(\mathbf{v}) = 0$  if  $\mathbf{v} \in \rho \tilde{\mathcal{B}}$  and  $\delta_{\rho \tilde{\mathcal{B}}}(\mathbf{v}) = \infty$  otherwise;

<sup>1</sup>[http://en.wikipedia.org/wiki/Quartic\\_function](http://en.wikipedia.org/wiki/Quartic_function)

<sup>2</sup>We found instances that the closed form solution can not give correct answers due to the limitation of the computers' precision.

that is,  $\delta_{\rho\bar{B}}(\mathbf{v})$  is the indicator function of  $\rho\bar{B}$ . Thus,

$$\text{prox}_{\delta_{\rho\bar{B}}}(\mathbf{w}) = \min_{\mathbf{u}} \frac{1}{2} \|\mathbf{u} - \mathbf{w}\|^2 + \delta_{\rho\bar{B}}(\mathbf{u}) = P_{\rho\bar{B}}(\mathbf{w}).$$

By Moreau decomposition (Rockafellar, 1970), we have

$$\text{prox}_{\rho f}(\mathbf{w}) = \mathbf{w} - \text{prox}_{(\rho f)^*}(\mathbf{w}) \Rightarrow \mathbf{u}^* = \mathbf{w} - \rho G^T \mathbf{s}^*,$$

which is the same as Eq. (26). By similar analysis as in Section 3.1, we can also solve (15). For more details, please refer to Parikh & Boyd (2013); Pustelnik et al. (2011).

### 3.3. Complexity Analysis of Algorithm 1

Each iteration of Algorithm 1 involves updating of the primal variables  $Z$ ,  $\mathbf{X}$ , and dual variables  $\Theta$ . We update  $Z$  and  $\Theta$  element-wise by simple arithmetic operations and thus the complexity is  $O(N)$ , where  $N = m \times n$  is the number of variables/pixels. The update of  $\mathbf{X}$  involves solving a set of small and independent subproblems. Because each subproblem involves at most three variables, the complexity of updating  $\mathbf{X}$  is proportional to  $N$ . Thus, the complexity of each iteration in Algorithm 1 is  $O(N)$ . From Section 2.4, to obtain an  $\epsilon$ -optimal solution, Algorithm 1 needs  $O(1/\epsilon)$  steps (He & Yuan, 2012). Thus, the complexity of Algorithm 1 to achieve an  $\epsilon$ -optimal solution is  $O(N/\epsilon)$ .

An appealing feature of Algorithm 1 is that every step of Algorithm 1 can be executed in parallel. Suppose that we have  $n_p$  processors. Then, we can run FAD simultaneously on these processors. Each processor solves a subset of subproblems (15) and (17) to update  $\mathbf{X}$  and updates a small piece of  $Z$  and  $\Theta$ . We call the parallel version of FAD ‘‘pFAD’’. The complexity of pFAD is  $O(\frac{N}{cn_p})$ . Clearly, the more processors we have, the more efficient pFAD will be. This is demonstrated in our experiments.

## 4. Implementation of pFAD via MPI

By the decomposition strategy in Section 2.2, the TV model in (1) can be decoupled into a set of small and independent subproblems as in (15) and (17). To fully utilize this feature, we implement pFAD via MPI such that pFAD can run on large systems of interconnected computer clusters in parallel. One of the key advantages of pFAD is that the communication overhead across processors is very small. Suppose that we divide the image into several blocks. Only the data on the ‘‘boundaries’’ need to be interchanged among the ‘‘adjacent’’ blocks (Fig. 1).

Fig. 2 shows an example of the synthetic images for testing. Each image  $\tilde{Y} \in \mathbb{R}^{n \times n}$  has eight randomly generated blocks. We add Gaussian noise  $\mathcal{N}(0, 0.2^2)$  to  $\tilde{Y}$  to generate the noisy image  $Y$ . The intensities are scaled to  $[0, 1]$ . Fig. 3(a) shows the speedup of pFAD compared with FAD when we increase the number of processors. The number of processors,  $c$ , is set to 1, 2, 4, 9, 16, 25, 100, 400. For

each specific number of processors, we fix  $\lambda = 0.35$  and vary  $n$ , the side length of the test images, from 100 to 5000 with a step size 100 (we observed similar patterns under different parameters). For each  $n$ , we perform 10 trials and record the average performance. Then we average the performance across all different  $n$  to get the speedup for each specific number of processors. Recall that Algorithm 1 involves updating the primal variables  $Z$  and  $\mathbf{X}$  and dual variables  $\Theta$ . Except  $\mathbf{X}$ , the update of  $Z$  and  $\Theta$  are element-wise and only require several simple arithmetic operations. Thus, the dominant part of Algorithm 1 is the update of  $\mathbf{X}$  since it requires that we solve a bunch of quartic equations. In Fig. 3(a), we report the speedup of updating  $\mathbf{X}$  and the total computational time. We can see that the speedup of updating  $\mathbf{X}$  approximately equals the number of processors. For 400 processors, the speedup is about 377 times. Fig. 3(a) also indicates that the speedup of the total computational time is about 357 times for 400 processors, since the cost of updating  $Z$  and  $\Theta$  is very low.

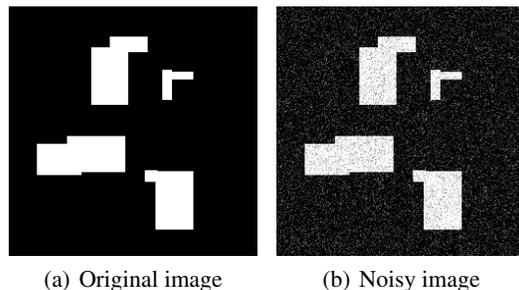


Figure 2. Example of synthetic images to evaluate the performance of MFISTA, SplitBregman and FAD/pFAD.

**Comparison with Other Algorithms** We compare FAD and its parallel variant pFAD with several state-of-the-art competitors including MFISTA<sup>3</sup> (Beck & Teboulle, 2009a) and SplitBregman method<sup>4</sup> (Goldstein & Osher, 2009). Note that it is not straightforward to implement MFISTA or SplitBregman in parallel since both of them involve manipulating or solving large-scale linear systems. Thus, we implement pFAD with OpenMP for this experiment and test all of the algorithms on a server with four quad-core (16 processors in total) Intel Xeon 2.93GHz CPUs and 65GB memory. To evaluate the performance of the aforementioned methods thoroughly, we test them on images whose sizes vary from  $100 \times 100$  to  $5000 \times 5000$ .

We compare the performance of the above algorithms by varying  $n$  and  $\lambda$ , respectively. We first fix  $\lambda = 0.35$  and vary the side length  $n$  of the images from 100 to 5000 with a step size of 100. Then, we fix  $n = 1000$  and vary  $\lambda$  from 0.15 to 0.9 with a step size of 0.05. For both settings, we perform 10 trials and report the average performance in

<sup>3</sup>[iew3.technion.ac.il/~becka/papers/tv\\_fista.zip](http://iew3.technion.ac.il/~becka/papers/tv_fista.zip)

<sup>4</sup>[tag7.web.rice.edu/Split\\_Bregman.html](http://tag7.web.rice.edu/Split_Bregman.html)

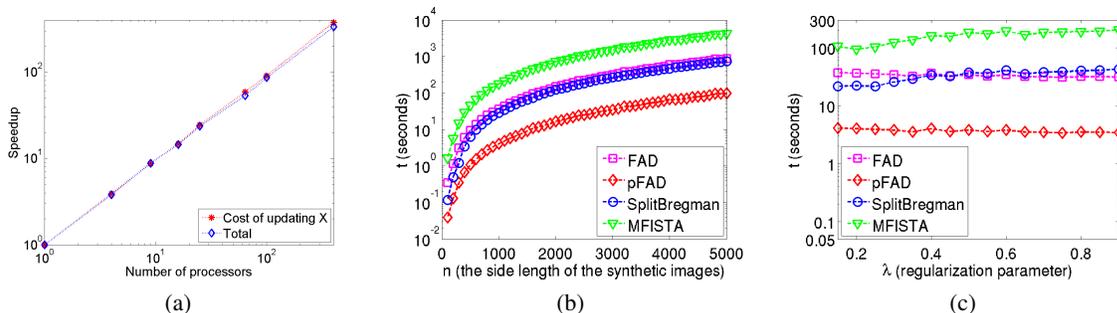


Figure 3. Efficiency evaluation. Fig. 3(a) shows the speedup of pFAD with respect to FAD using multiple processors. Fig. 3(b) and Fig. 3(c) present the efficiency comparison of FAD, pFAD, SplitBregman and MFISTA in terms of the computational time.

Fig. 3(b) and Fig. 3(c), respectively. For a fair comparison, in each setting, we first run FAD and record the objective function value when it stops. Then we run all the other algorithms until the objective function values are no larger than that of FAD (for FAD/pFAD, we set  $\epsilon = 10^{-4}$  and  $\gamma = 10$ ).

Fig. 3(b) indicates that the performance of FAD is comparable to the SplitBregman method and about six times faster than MFISTA. pFAD further improves the efficiency of FAD by about one order. Thus, pFAD is about 60 and 10 times faster than MFISTA and the SplitBregman method, respectively. Fig. 3(c) further demonstrates the efficiency of pFAD. We can observe that, pFAD and FAD become slightly faster when  $\lambda$  increases. From Theorems 1 and 2, when  $\rho \geq \rho_{max}$  ( $\lambda$  is proportional to  $\rho$  since  $\gamma$  is fixed), problem (15) admits a closed-form solution. Otherwise, we solve the quartic equation (31). Overall, FAD/pFAD is more stable than SplitBregman and MFISTA.

## 5. Applications

We apply pFAD to the problems of image denoising, deblurring and reconstruction and present the results below.

### 5.1. Image Denoising

We apply our methods to the image denoising problem. The original image<sup>5</sup> is a high-resolution photograph of the moon with  $5100 \times 4768$  pixels (about 13 MB). The noisy image is obtained by adding Gaussian noise  $\mathcal{N}(0, 0.2^2)$ . The regularization parameter  $\lambda$  is set to be 0.2. To give a better view, we show the results over a sub-image with  $512 \times 512$  pixels. The upper left corner of the sub-image is at (1800, 2200). The denoising process takes pFAD, SplitBregman and MFISTA 46.87, 445.66 and 2695.28 seconds, respectively. MFISTA takes about 45 minutes. SplitBregman is much more efficient than MFISTA, but still needs about 7 minutes. The proposed pFAD only needs several tens of seconds and thus performs the best.

<sup>5</sup><http://blog.astrophotographytargets.com/2012/03/>

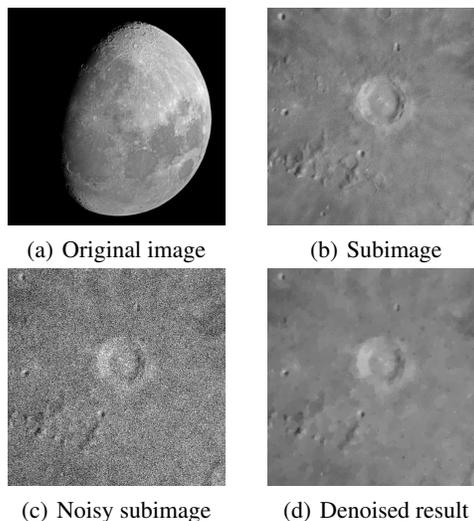


Figure 4. Image denoising.

### 5.2. Image Deblurring

The TV-based image deblurring model is:

$$\min_X \|\mathcal{B}(X) - Y\|_F^2 + \lambda \|X\|_{TV}, \quad (33)$$

where  $X$  is the image to be recovered,  $\mathcal{B}$  is a linear transformation encoding a certain blurring operation, and  $Y$  is the observed noisy and blurred image. Problem (33) can be efficiently solved via FISTA (Beck & Teboulle, 2009b). The key step is to solve the proximal operator of the TV term, which is equivalent to problem (1). Therefore, we can apply FAD or pFAD to solve the proximal operator to speedup the computation. The original image “text” in Fig. 5(a) is

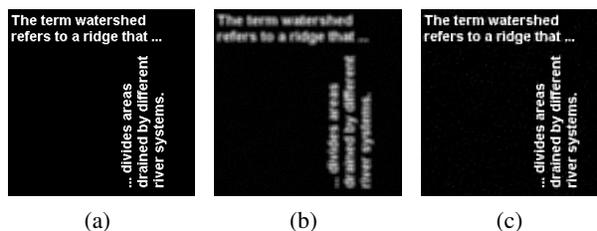


Figure 5. Image deblurring via FISTA-pFAD.

extracted from the Matlab image processing toolbox. We then apply a  $7 \times 7$  Gaussian filter with a standard devia-

Table 1. Computational time (in seconds) for MRI reconstruction.

	CARDIAC	BRAIN	CHEST	ARTERY
FCSA-MFISTA	0.682	0.607	0.633	0.712
FCSA-pFAD	0.037	0.033	0.041	0.031

tion 5 to get the blurred image. Fig. 5(b) is obtained by adding Gaussian noise  $\mathcal{N}(0, 0.02^2)$  to the blurred image. The regularization parameter  $\lambda$  of pFAD is set to be 0.001. Fig. 5(c) shows the deblurred image via FISTA-pFAD.

### 5.3. Image Reconstruction

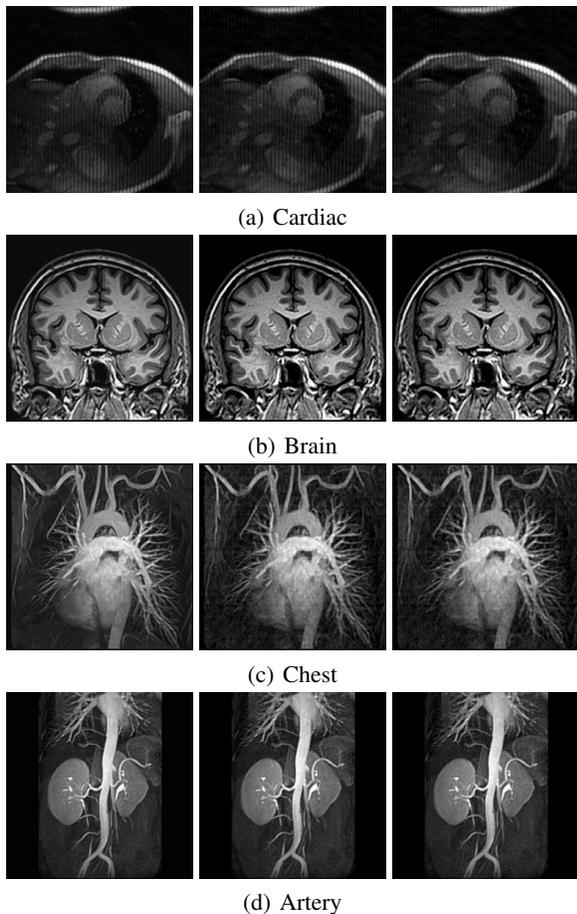


Figure 6. MRI reconstruction. The left column shows the original images. The images in the middle and right columns are the reconstructed ones by FCSA-MFISTA and FCSA-pFAD, respectively. The SNR for Cardiac, Brain, Chest and Artery are 17.56, 20.35, 16.06 and 23.70 respectively.

Magnetic Resonance Imaging (MRI) is a powerful tool in medical diagnosis because of its excellent performance in detecting soft tissue changes. However, a typical MRI examination procedure may take up 45 minutes. By noting the sparse nature of signals in a transformed domain, recent advances in the compressive sensing theory (CS) (Candès et al., 2006; Donoho, 2006) show that it is possible to accurately reconstruct signals with limited undersamples. Therefore, the duration of an MRI examination can be sig-

nificantly reduced. A popular formulation of the image reconstruction problem takes the form of (Lustig et al., 2007; Trzasko et al., 2007):

$$\min_X \frac{1}{2} \|\mathcal{F}(X) - Y\|^2 + \lambda_1 \|\mathcal{W}(X)\|_1 + \lambda_2 \|X\|_{TV}, \quad (34)$$

where  $X$  is the signal to be recovered,  $Y$  are the observed undersamples and  $\mathcal{F}$  and  $\mathcal{W}$  are partial Fourier and wavelet transformations, respectively. Huang et al. (2011) proposed an efficient algorithm called FCSA<sup>6</sup> to solve (34). One of the key steps of FCSA is to solve the proximal operator of the TV term by MFISTA. To improve the efficiency of FCSA further, we use pFAD to replace MFISTA.

We compare the performance of FCSA-MFISTA and FCSA-pFAD on four MRI images: cardiac, brain and artery. We use the default settings of the FCSA package. The sample ratio is about 20% and Gaussian noise  $\mathcal{N}(0, 0.01^2)$  is added to the undersamples to generate  $Y$ . For a fair comparison, we first run FCSA-pFAD and record the objective function values returned by pFAD. Then, we run FCSA-MFISTA and make sure it achieves the same precision level with FCSA-pFAD at each iteration. We run both FCSA-pFAD and FCSA-MFISTA for 50 iterations. Fig. 6 presents the results obtained by FCSA-MFISTA and FCSA-pFAD, which are almost identical to each other. Because both FCSA-MFISTA and FCSA-pFAD solve the same problem and achieve the same accuracy level, the resulting reconstructions and SNR are the same. The average time for the 50 iterations of the two methods is reported in Table 1, which indicates that FCSA-pFAD is about 20 times faster than is FCSA-MFISTA.

## 6. Conclusion

In this paper, we propose a fast alternating direction method for isotropic TV models. Our approaches are based on a novel decomposition strategy of the problem domain. By the decomposition, isotropic TV models can be decoupled into a set of small and independent subproblems, which can be solved efficiently in parallel. Different from existing ADMM based approaches, like SplitBregman methods, one appealing feature of our method is that no large-scale linear system is involved. Our empirical evaluation demonstrates the very low communication overhead and high scalability of the proposed method. One of our future directions is to extend our approach to higher dimensional problems, like video and functional MRI denoising.

## Acknowledgments

This work was supported in part by NIH (LM010730), NSF (IIS-0953662, CCF-1025177) and China 973 Fundamental R&D Program (No.2014CB340304). We would like to thank Virginia Unkefer for proofreading.

<sup>6</sup>[ranger.uta.edu/~huang/R\\_CSMRI.htm](http://ranger.uta.edu/~huang/R_CSMRI.htm)

## References

- Barbero, Á. and Sra, S. Fast newton-type methods for total variation regularization. In *ICML*, 2011.
- Beck, A. and Teboulle, M. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing*, 18:2419–2434, 2009a.
- Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2:183–202, 2009b.
- Bertsekas, D. P. *Convex Analysis and Optimization*. Athena Scientific, 2003.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.
- Candès, E., Romberg, J., and Tao, T. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52:489–509, 2006.
- Casas, E., Kunisch, K., and Pola, C. Regularization by functions of bounded variation and applications to image enhancement. *Applied Mathematics & Optimization*, 40:229–257, 1999.
- Chambolle, A. and Pock, T. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging. Vis.*, 40:120–145, 2011.
- Chan, R. H., Yang, J. F., and Yuan, X. M. Alternating direction method for image inpainting in wavelet domain. *SIAM Journal on Imaging Sciences*, 4:807–826, 2011.
- Chen, C., He, B., and Yuan, X. Matrix completion via an alternating direction method. *IMA Journal of Numerical Analysis*, 32:227–245, 2012.
- Combettes, P. and Pesquet, J. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, 2011.
- Condat, L. A direct algorithm for 1-D total variation denoising. *IEEE Signal Processing Letters*, 20:1054–1057, 2013.
- Donoho, D. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.
- Duan, Y. and Tai, X. Domain decomposition methods with graph cuts algorithms for total variation minimization. *Adv. Comput. Math*, 36:175–199, 2012.
- Eckstein, J. and Bertsekas, D. P. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992.
- Esser, E. Applications of lagrangianbased alternating direction methods and connections to split Bregman. Technical report, UCLA CAM, 2009.
- Esser, E., Zhang, X. Q., and Chan, T. F. A general framework for a class of first order primaldual algorithms for convex optimization in imaging science. *SIAM J. Imag. Sci.*, 3:1015–1046, 2010.
- Glowinski, R. and Tallec, P. Le. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*. SIAM Studies in Applied Mathematics, 1989.
- Goldfarb, D. and Yin, W. Parametric maximum flow algorithms for fast total variation minimization. *SIAM J. Sci. Comput.*, 31:3712–3743, 2009.
- Goldstein, Tom and Osher, Stanley. The Split Bregman method for L1-regularized problems. *SIAM J. Imag. Sci.*, 2:323–342, 2009.
- He, B. and Yuan, X. On the  $o(1/n)$  convergence rate of the Douglas-Rachford alternating direction method. *SIAM J. Numer. Anal.*, 50:700–709, 2012.
- He, L., Chang, T. C., and Osher, S. MR image reconstruction from sparse radial samples by using iterative refinement procedures. In *13th Annual Meeting of ISMRM*, 2005.
- Huang, J., Zhang, S., and Metaxas, D. Efficient MR image reconstruction for compressed MR imaging. *Medical Image Analysis*, 15:670–679, 2011.
- Kim, D., Sra, S., and Dhillon, I. A scalable trust-region algorithm with application to mixed-norm regression. In *ICML*, 2010.
- Lustig, M., Lee, J. H., Donoho, D. L., and Pauly, J. M. Faster imaging with randomly perturbed undersampled spirals and  $l_1$  reconstruction. In *13th Scientific Meeting of the ISMRM*, 2005.
- Lustig, M., Donoho, D., and Pauly, J. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magn. Reson. Med.*, 58:1182–1195, 2007.
- Ma, S., W, Yin, Zhang, Y., and Chakraborty, A. An efficient algorithm for compressed MR imaging using total variation and wavelets. In *CVPR*, 2008.
- Monteiro, R. and Svaiter, B. F. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. manuscript, 2010.
- Osher, S., Burger, M., Goldfarb, D., Xu, J., and Yin, W. An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.*, 4:460–489, 2005.
- Parikh, N. and Boyd, S. Proximal algorithms. *Foundations and Trends in Optimization*, 1:123–231, 2013.
- Pustelnik, N., Chaux, C., and Pesquet, J.-C. Parallel proximal algorithm for image restoration using hybrid regularization. *IEEE Transactions on Image Processing*, 20:2450–2462, 2011.
- Rockafellar, R. *Convex Analysis*. Princeton University Press, 1970.
- Rudin, L. I., Osher, S. J., and Fatemi, E. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60:259–268, 1992.
- Ruszczynski, A. *Nonlinear Optimization*. Princeton University Press, 2006.
- Trzasko, J., Manduca, A., and Borisch, E. Sparse MRI reconstruction via multiscale  $L_0$ -continuation. In *14th IEEE/SP Workshop on Statistical Signal Processing*, 2007.
- Vert, J. P. and Bleakley, K. Fast detection of multiple change-points shared by many signals using group LARS. In *NIPS*, 2010.
- Vogel, C. and Oman, M. Fast, robust total variation-based reconstruction of noisy, blurred images. *IEEE Trans. Image Process.*, 7:813–824, 1998.
- Wahlberg, B., Boyd, S., Annergren, M., and Wang, Y. An ADMM algorithm for a class of total variation regularized estimation problems. In *IFAC Symposium on System Identification*, 2012.
- Yang, S., Wang, J., Fan, W., Zhang, X., Wonka, P., and Ye, J. An efficient ADMM algorithm for multidimensional anisotropic total variation regularization problems. In *SIGKDD*, 2013.