SIAM J. IMAGING SCIENCES
Vol. 4, No. 3, pp. 807–826

Alternating Direction Method for Image Inpainting in Wavelet Domains*

Raymond H. Chan†, Junfeng Yang‡, and Xiaoming Yuan§

Abstract. Image inpainting in wavelet domains refers to the recovery of an image from incomplete and/or inaccurate wavelet coefficients. To reconstruct the image, total variation (TV) models have been widely used in the literature, and they produce high-quality reconstructed images. In this paper, we consider an unconstrained, TV-regularized, ℓ_2-data-fitting model to recover the image. The model is solved by the alternating direction method (ADM). At each iteration, the ADM needs to solve three subproblems, all of which have closed-form solutions. The per-iteration computational cost of the ADM is dominated by two Fourier transforms and two wavelet transforms, all of which admit fast computation. Convergence of the ADM iterative scheme is readily obtained. We also discuss extensions of this ADM scheme to solving two closely related constrained models. We present numerical results to show the efficiency and stability of the ADM for solving wavelet domain image inpainting problems. Numerical results comparing the ADM with some recent algorithms are also reported.

Key words. total variation, wavelet, inpainting, augmented Lagrangian method, alternating direction method, fast Fourier transform, fast wavelet transform

AMS subject classifications. 68U10, 65J22, 65K10, 65T50, 90C25

DOI. 10.1137/100807247

1. Introduction.

Image inpainting is an important image processing task in many applications, and it has been studied extensively in the literature; see, e.g., [19]. Image inpainting refers to the problem of filling in missing or damaged regions in images, either in the pixel domain or in a transformed domain, depending on how the image is damaged. Let u^* be an unknown image. Without loss of generality, we assume that u^* is an n-by-n square image; our discussions apply to nonsquare images as well. Following the standard treatment, we vectorize two-dimensional images into one-dimensional vectors. Therefore, throughout this paper we treat n-by-n images as vectors in $\mathbb{R}^{n^2}$. In general, the image inpainting problem can be viewed as recovering an unknown image u^* from

$$f = (P\mathcal{T}u^* + \omega) \in \mathbb{R}^p, \tag{1.1}$$

where $\mathcal{T} \in \mathbb{R}^{n^2 \times n^2}$ represents a transform matrix, $P \in \mathbb{R}^{p \times n^2}$ is a projection/downsampling matrix containing p ($< n^2$) rows of the identity matrix of order n^2, $\omega \in \mathbb{R}^p$ contains the

*Received by the editors September 1, 2010; accepted for publication (in revised form) June 13, 2011; published electronically September 1, 2011.
http://www.siam.org/journals/siims/4-3/80724.html
†Department of Mathematics, the Chinese University of Hong Kong, Shatin, Hong Kong (rchan@math.cuhk.edu.hk). This author's research was supported by HKRGC grant CUHK 400510 and DAG grant 2060408.
‡Department of Mathematics, Nanjing University, Nanjing, Jiangsu, 210093, China (jfyang@nju.edu.cn). This author's research was supported by National Science Foundation of China grant NSFC-11001123 and the Fundamental Research Funds for the Central Universities grant 1117020305.
§Corresponding author. Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (xmyuan@hkbu.edu.hk). This author's research was supported by HKRGC grant HKBU 202610.

contamination of noise introduced during both the process of observing the original image and the subsequent transmissions of the transformed and compressed coefficients, and $f \in \mathbb{R}^p$ denotes the remaining incomplete and inaccurate transformed coefficients. The positions of the remaining (and thus those missed) transformed coefficients are determined by P.

In [6], Bertalmio et al. first considered image inpainting in the pixel domain; i.e., $\mathcal{T}$ is the identity matrix, and f contains approximate values of the p remaining pixels. Inpainting in the pixel domain is, in some sense, an interpolation problem, where values of the missed pixels are approximated by using those of the remaining. Inpainting methods in the pixel domain usually take advantage of the property that the missed pixels are local and hence that their values can be estimated by utilizing neighboring information. We here list some existing approaches in the literature. The authors of [6] used partial differential equations to smoothly propagate information from the surrounding areas along the isophotes into the inpainting domain. Subsequently, Ballester et al. proposed a variational inpainting model based on a joint cost functional on the gradient vector field and gray values in [5]. Chan and Shen considered a total variational (TV) inpainting model in [18] and the curvature driven diffusion model in [17]. The TV inpainting model fills in the missing regions such that the TV is minimized, and its use is motivated by the wide applications of TV in image restoration. In [16], Chan, Kang, and Shen also introduced an inpainting technique using an Euler's elastica energy-based variational model. In [9], Cai, Chan, and Shen used a tight-frame approach for inpainting and showed that it is equivalent to using an ℓ_1 regularization on the tight-frame coefficients. All these works concentrate on image inpainting in the pixel domain.

Inpainting in transformed domains is totally different because each single corruption of data can, in general, affect the whole image, and thus an inpainting region in the pixel domain is not well defined. Transformed domain inpainting arises in practical applications because images are usually formatted, transmitted, and stored in a transformed domain. For examples, in the JPEG standard images are transformed by cosine transforms, while in the JPEG2000 standard images are transformed to a wavelet domain through wavelet transforms. In such situations, the transform $\mathcal{T}$ is either a discrete cosine transform (DCT) or a discrete wavelet transform (DWT). During storage and transmission, certain coefficients may be lost or corrupted, which naturally leads to the transformed domain inpainting problem. In this paper, we consider recovering the original image u^* from incomplete and inaccurate wavelet coefficients; i.e., $\mathcal{T}$ is an orthonormal wavelet transform.

In practical applications, the downsampling matrix P in (1.1) is usually determined according to a certain "thresholding" rule; i.e., those transformed coefficients with magnitudes bigger than some threshold value are kept, while the others are discarded. In this situation, P is also known as a thresholding compressing operator. We note that, in addition to thresholding based image compression, other compression approaches are certainly applicable in practical applications. For example, P can be a "quantization"-based compressing operator, in which case the binary representation of $\mathcal{T}u^*$ is quantized so that only bits with order higher than a prescribed value are kept, while the remaining bits of lower orders are discarded; see [10] for details about an iterative frame-based algorithm for image inpainting in the bits domain. In this paper, we concentrate on the case that P is a selection/downsampling operator.

1.1. Wavelet domain inpainting via TV regularization. Rane and coworkers [42, 43] considered wavelet domain inpainting in wireless networks, where separated reconstruction

techniques are used for image structures and textures. Inspired by the great success of TV in image restoration, Chan, Shen, and Zhou in [20] proposed to utilize TV regularization for image inpainting in wavelet domains. In [20], the following TV-regularized data-fitting model was solved to approximately reconstruct the original image:

$$\min_u \sum\nolimits_i \|D_i u\|_2 + \frac{\mu}{2}\|PWu - f\|_2^2, \tag{1.2}$$

where $W \in \mathbb{R}^{n^2\times n^2}$ represents a discrete orthonormal wavelet transform matrix, f contains those remaining inaccurate wavelet coefficients (given by (1.1) with $\mathcal{T} = W$), $\|\cdot\|_2$ represents the ℓ_2-norm, $\sum_i \|D_i u\|_2$ is a discretization of the TV of u where $\sum_i$ is taken over all pixels, and $\mu > 0$ is a scalar balancing regularization and fidelity. In this paper, we assume that the contamination of noise is random Gaussian, and thus the ℓ_2-norm square fidelity is used. In section 2.3, we will briefly address a constrained alternative of (1.2), as well as an equality constrained model which is suitable for noiseless data.

1.2. Some existing approaches for TV models. TV regularization was first proposed by Rudin, Osher, and Fatemi in [45, 44] for image denoising and deconvolution. Subsequently, TV regularization was studied extensively in image restoration; see, e.g., [11, 15] and references therein. The advantage of TV regularization is that it can exploit blocky image structures and preserve edges. The superiority of TV over Tikhonov regularization [50] is analyzed in [1, 26] for recovering images containing piecewise smooth objects. Despite these advantages, it is generally challenging to solve TV models efficiently in practice because imaging problems are usually large scale, ill-conditioned, and moreover contain nonsmooth TV. Here we review some existing approaches to TV models.

Early research on the numerical solution of TV models concentrated on smoothed TV models, where the TV is approximated by $\sum_i \sqrt{\|D_i x\|_2^2 + \epsilon}$ with small $\epsilon > 0$. As such, ordinary optimization methods for smooth function minimization can be applied, e.g., the time-marching scheme used in the pioneering work [45], or the linearized gradient method proposed in [51] for denoising and in [52] for deconvolution. Another class of algorithms for TV problems are those based upon the iterative shrinkage/thresholding (IST) operator; see, e.g., [24, 28, 30, 47, 23]. At each iteration of IST-based algorithms, a TV denoising problem needs to be solved, either exactly or approximately.

Recent approaches to TV models are based on appropriate splitting of the TV norm. A novel splitting of the TV norm was proposed in [53], where the authors utilized the quadratic penalty technique to derive an efficient alternating minimization algorithm. This splitting of TV allows the use of a multidimensional shrinkage operator and fast Fourier transform for deconvolution problems. Given its efficiency, the technique used in [53] was extended to multichannel image deconvolution in [56] and impulsive noise elimination in [58]. Goldstein and Osher [35] applied the classical augmented Lagrangian method [37, 41], where the authors derived the algorithm based on the Bregman distance [8]. In [35], the authors used an alternating strategy to minimize the augmented Lagrangian function; they proposed to solve the inner subproblem approximately by only one alternating step, which reduces to the classical ADM. Recently, the ADM has been applied to a set of imaging problems; see, e.g., [29, 46, 59, 2, 3, 39]. Also, recently, a first-order primal-dual algorithm was proposed in [13], which is shown to have attractive convergence results for uniformly convex problems and is

closely related to the Douglas–Rachford splitting algorithm [25] as well as the ADM [34, 31] to be discussed in this paper.

For solving problem (1.2), an explicit gradient descent scheme was used in [16]. An optimization transfer algorithm was recently proposed in [14], which is related to the half-quadratic technique [32]. Recently, model (1.2) was extended in [60] to recover textures and local geometry structures of natural images by utilizing a nonlocal TV regularizer, where the Bregman iteration [40] and the operator splitting algorithms [22, 61] were applied to solving the underlying problems. The main contribution of this paper is to introduce a simple yet very efficient algorithm for solving (1.2) and some related problems as well. In addition, we compare the proposed algorithm with the optimization transfer algorithm recently proposed in [14] and the two-step IST algorithm in [7].

1.3. Notation and organization. Let the superscript "$\top$" be the transpose operator for real 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 $(A_1; A_2) = (A_1^\top, A_2^\top)^\top$. As used in (1.2), for each i, D_i is a 2-by-n^2 matrix such that the two entries of $D_i u$ represent the horizontal and vertical local finite differences of u at pixel i (boundary conditions will be specified later). The corresponding horizontal and vertical global finite difference matrices are denoted by $D^{(1)}$ and $D^{(2)}$, respectively. As such, $D^{(1)}$ and $D^{(2)}$ contain, respectively, the first and second rows of D_i for all i's. For the rest of the paper, we let $\|\cdot\| = \|\cdot\|_2$. Additional notation will be defined when it occurs.

The paper is organized as follows. In section 2, we present the basic algorithm for solving (1.2) and discuss its extensions to solving some related problems. Section 3 reports experimental results in comparisons with the optimization transfer algorithm [14] and the two-step iterative shrinkage/thresholding algorithm [7]. Finally, some concluding remarks are given in section 4.

2. Basic algorithm and related work. The main difficulty in solving (1.2) is due to the nondifferentiability of the involved TV norm. Analogous to [53], we shall first reformulate (1.2) into a minimization problem with linear constraints. Then, we shall apply the influential ADM [31] to solve the reformulation. In the following, we first present the reformulation and briefly review the classical augmented Lagrangian method (ALM; see, e.g., [37, 41]), from which the ADM is motivated. Then, we delineate the iterative scheme of the ADM for solving (1.2). Finally, we discuss some extensions of the ADM to some other relevant models.

2.1. Reformulation and the ALM. By introducing auxiliary variables $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_{n^2}]$, where each $\mathbf{w}_i$ is a column vector in $\mathbb{R}^2$, the model (1.2) is equivalent to

$$\min_{u,\mathbf{w}} \left\{ \sum_i \|\mathbf{w}_i\| + \frac{\mu}{2}\|Pv - f\|^2 \;\middle|\; v = Wu,\ \mathbf{w}_i = D_i u\ \forall\, i \right\}. \tag{2.1}$$

Here $\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_{n^2}]$ is a 2-by-n^2 matrix. For convenience, the jth row of $\mathbf{w}$ is denoted by $w_j^\top$, $j = 1, 2$. Thus, the constraints $\{\mathbf{w}_i = D_i u\ \forall i\}$ are equivalent to $w_j = D^{(j)}u$, $j = 1, 2$, or, more compactly, $w = Du$, where $w = (w_1; w_2)$ and $D = (D^{(1)}; D^{(2)})$. Since $\mathbf{w}$ and w are the same variables with different ordering, in the following we use either $\mathbf{w}$ or w subject to convenience. The motivation for considering the reformulation problem (2.1) is that the selection operator P, the finite difference operator D, and the wavelet transform operator W

are separated in such a way that their special structures can be fully utilized. This is actually the key idea of the efficient implementation of the ADM algorithm to be presented.

The model (2.1) is a standard convex program with linear constraints, for which a classical approach is the ALM (see, e.g., [37, 41]). More specifically, let the augmented Lagrangian function of (2.1) be denoted by

$$\mathcal{L}_{\mathcal{A}}(u,v,w,\lambda,\eta) := \sum_i \left(\|\mathbf{w}_i\| - \lambda_i^\top(\mathbf{w}_i - D_i u) + \frac{\beta_1}{2}\|\mathbf{w}_i - D_i u\|^2 \right) + \frac{\mu}{2}\|Pv - f\|^2 - \eta^\top(v - Wu) + \frac{\beta_2}{2}\|v - Wu\|^2, \tag{2.2}$$

where $\lambda_i \in \mathbb{R}^2$ (for all i) and $\eta \in \mathbb{R}^{n^2}$ are multipliers, and $\beta_1, \beta_2 > 0$ are penalty parameters. The iterative scheme of the ALM for (2.1) is given by

$$\begin{cases} (u,v,w)^{k+1} = \arg\min \mathcal{L}_{\mathcal{A}}(u,v,w,\lambda^k,\eta^k), \\ \lambda_i^{k+1} = \lambda_i^k - \gamma\beta_1(\mathbf{w}_i^{k+1} - D_i u^{k+1}) \quad \forall i, \\ \eta^{k+1} = \eta^k - \gamma\beta_2(v^{k+1} - Wu^{k+1}), \end{cases} \tag{2.3}$$

where $\gamma > 0$ is a relaxation parameter. Note that at each iteration the ALM needs to solve a joint minimization problem with respect to (u,v,w), either exactly or approximately.

2.2. Solving (2.1) using the ADM. Let us revisit the minimization task in (2.3) which aims at solving all the variables (u,v,w) simultaneously. Note that the augmented Lagrangian function $\mathcal{L}_{\mathcal{A}}(u,v,w,\lambda,\eta)$ of (2.2) has favorable separable structures. First, variables $\mathbf{w}$ and v are completely separated from each other, and thus their minimization can be implemented in parallel. Second, with fixed u, the minimization for $\mathbf{w}$ reduces to the minimization with each $\mathbf{w}_i$, and the minimization for v is componentwise separable. Hence, the direct application of the ALM, which minimizes the variables (u,v,w) jointly, treats (2.1) as a generic convex program and completely ignores the favorable separable structure.

The influential ADM [31] takes full advantage of the separable structure and decomposes the joint minimization task of the ALM into some smaller and easier ones in the alternating order. In the literature, the ADM has been well studied especially in the context of convex programming and variational inequalities. We refer to [33, 46] for the connection between the ADM and the classical Douglas–Rachford splitting method [25], and to [27] for the relationship of the ADM with the renowned proximal point algorithm. More specifically, with the given $(u,\lambda,\eta)^k := (u^k,\lambda^k,\eta^k)$, the iterative scheme of the ADM for (2.1) is as follows:

$$\begin{cases} (v,w)^{k+1} = \arg\min_{v,w} \mathcal{L}_{\mathcal{A}}(u^k,v,w,\lambda^k,\eta^k), \\ u^{k+1} = \arg\min_u \mathcal{L}_{\mathcal{A}}(u,v^{k+1},w^{k+1},\lambda^k,\eta^k), \\ \lambda_i^{k+1} = \lambda_i^k - \gamma\beta_1(\mathbf{w}_i^{k+1} - D_i u^{k+1}) \quad \forall i, \\ \eta^{k+1} = \eta^k - \gamma\beta_2(v^{k+1} - Wu^{k+1}). \end{cases} \tag{2.4}$$

It is easy to see that the ADM inherits the spirit of the Gauss–Seidel iteration in the sense that the minimization for u uses the most up-to-date iterates of v and w.

Now, let us delineate the way of solving the subproblems of the ADM. Clearly, the v-subproblem in (2.4) is equivalent to

$$\min_v \frac{\mu}{2}\|Pv - f\|^2 + \frac{\beta_2}{2}\|v - \zeta^k\|^2, \tag{2.5}$$

where $\zeta^k = Wu^k + \eta^k/\beta_2$. We let v_P (resp., $v_{\bar{P}}$) be the subvector of v formed by those entries selected (resp., not selected) by P. Thus, $Pv = v_P$. It is easy to show that the minimizer v^{k+1} of (2.5) is given by

$$v_P^{k+1} = \frac{\beta_2\zeta_P^k + \mu f}{\beta_2 + \mu} \quad \text{and} \quad v_{\bar{P}}^{k+1} = \zeta_{\bar{P}}^k. \tag{2.6}$$

For the minimization of w (or equivalently $\mathbf{w}$) in (2.4), for each $i \in \{1, 2, \ldots, n^2\}$ we let $\xi_i^k = D_i u^k + \lambda_i^k/\beta_1$, and the minimization for $\mathbf{w}_i$ is given by (see, e.g., [53, 35])

$$\mathbf{w}_i^{k+1} = \arg\min_{\mathbf{w}_i} \|\mathbf{w}_i\| + \frac{\beta_1}{2}\|\mathbf{w}_i - \xi_i^k\|^2 = \max\left\{\|\xi_i^k\| - \frac{1}{\beta_1}, 0\right\} \times \frac{\xi_i^k}{\|\xi_i^k\|}, \tag{2.7}$$

where we follow the convention $0 \times (0/0) = 0$.

Finally, with fixed $v = v^{k+1}$ and $w = w^{k+1}$, the u-subproblem in (2.4) is a least-squares problem whose normal equations are given by

$$Au^{k+1} = b^k, \tag{2.8}$$

where, by noting the orthnormality of W,

$$\begin{cases} A = \beta_1 \sum_i D_i^\top D_i + \beta_2 I = \beta_1 D^\top D + \beta_2 I, \\ b^k = D^\top(\beta_1 w^{k+1} - \lambda^k) + W^\top(\beta_2 v^{k+1} - \eta^k). \end{cases} \tag{2.9}$$

Under the periodic boundary conditions, $D^\top D$ is a blockwise circulant matrix with circulant blocks and thus can be diagonalized by two-dimensional discrete Fourier transform. Therefore, (2.8) can be easily solved by two FFTs (fast Fourier transforms). Specifically, u^{k+1} is given by

$$u^{k+1} = \mathcal{F}^{-1}\left(\mathcal{F}(b^k)./\mathrm{diag}(\mathcal{F}A\mathcal{F}^{-1})\right), \tag{2.10}$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent, respectively, the two-dimensional forward and inverse Fourier matrix, $\mathrm{diag}(\cdot)$ takes the diagonal elements, and "./" denotes componentwise division. Alternatively, under the Neumann boundary conditions, $D^\top D$ is a block Toeplitz-plus-Hankel matrix (see, e.g., [38]), and system (2.8) can be solved efficiently by two-dimensional DCTs. Therefore, all the subproblems arising in the ADM for solving (2.1) have closed-form solutions. This feature is very beneficial for achieving attractive numerical performance, as we shall show in the next section.

Although one can circularly apply (2.6), (2.7), and (2.10) until $\mathcal{L}_\mathcal{A}(u, v, w, \lambda^k, \eta^k)$ is minimized jointly with respect to (u, v, w) and update the multipliers as in the ALM (2.3), we choose to update λ^k and η^k immediately after merely one round of minimizations in the alternating order. Now, we are ready to present the algorithm of ADM for solving (2.1) (or, equivalently, (1.2)).

Algorithm 1. *Input problem data* P, f *and model parameters* $\mu > 0$. *Given* $\beta_1, \beta_2 > 0$ *and* $\gamma \in \left(0, (\sqrt{5}+1)/2\right)$. *Initialize* $u = u^0$, $\lambda = \lambda^0$, *and* $\eta = \eta^0$. *Set* $k = 0$. *Compute* $\Lambda := \mathcal{F}\left(\beta_1 D^\top D + \beta_2 I\right)\mathcal{F}^{-1}$.

While *"not converged",* **Do**

(1) *Compute* v^{k+1} *and* w^{k+1}*:*

$$\begin{cases} \zeta^k = Wu^k + \eta^k/\beta_2, \\ \xi_i^k = D_i u^k + \lambda_i^k/\beta_1 \quad \forall i, \\ v_P^{k+1} = \frac{\beta_2 \zeta_P^k + \mu f}{\beta_2 + \mu}, \\ v_{\bar{P}}^{k+1} = \zeta_{\bar{P}}^k, \\ \mathbf{w}_i^{k+1} = \max\left\{\|\xi_i^k\| - \frac{1}{\beta_1}, 0\right\} \times \frac{\xi_i^k}{\|\xi_i^k\|} \quad \forall i. \end{cases}$$

(2) *Compute* u^{k+1}*:*

$$\begin{cases} b^k = D^\top(\beta_1 w^{k+1} - \lambda^k) + W^\top(\beta_2 v^{k+1} - \eta^k), \\ u^{k+1} = \mathcal{F}^{-1}\left(\mathcal{F}(b^k)./\mathrm{diag}(\Lambda)\right). \end{cases}$$

(3) *Update* λ^k *and* η^k *by*

$$\begin{cases} \lambda_i^{k+1} = \lambda_i^k - \gamma\beta_1(\mathbf{w}_i^{k+1} - D_i u^{k+1}) \quad \forall i, \\ \eta^{k+1} = \eta^k - \gamma\beta_2(v_i^{k+1} - Wu^{k+1}). \end{cases}$$

(4) $k \leftarrow k + 1$.

End Do

We note that the per iteration cost of this ADM framework is dominated by two DWTs (one forward for computing Wu^k, and one inverse DWT for computing b^k) and two FFTs (one forward and one inverse for computing u^{k+1}). The complexity of each such computation is $O(N \log N)$ for a vector of length N. All other computations have linear complexity. We also note that this ADM framework is applicable if the isotropic TV in (1.2) is replaced by an anisotropic TV of the form $\mathrm{TV}(u) = \sum_i \|D_i u\|_1$. In this case, the only modification of Algorithm 1 lies in the computation for $\mathbf{w}^{k+1}$, which is given by

$$\mathbf{w}_i^{k+1} = \max\left\{|\xi_i^k| - 1/\beta_1, 0\right\} \circ \mathrm{sgn}(\xi_i^k) \quad \forall i,$$

where "sgn$(\cdot)$" denotes the signum function, and $|\cdot|$ and "$\circ$" represent, respectively, componentwise absolute value and multiplication.

It is easy to see that the fast and exact minimization of the u-subproblem in the ADM framework (2.4) is attainable, provided that $\mathcal{T}$ is orthonormal and $D^\top D$ is diagonalizable by fast transforms. Therefore, the ADM framework (2.4) can be easily generalized to other problem scenarios including image inpainting from cosine transformed or tight-frame coefficients [9].

Implementation details of Algorithm 1 including stopping criteria, and choices of parameters will be specified in section 3. The convergence of the proposed algorithm can be found in the literature of the ADM, e.g., [31, 33]. For succinctness, we present only the convergence theorem without detailed proof.

Theorem 2.1. *For any* $\beta_1, \beta_2 > 0$ *and* $\gamma \in (0, (\sqrt{5}+1)/2)$, *the sequence* $\{(u^k, v^k, w^k)\}$ *generated by Algorithm 1 from any starting point* (u^0, λ^0, η^0) *converges to a solution of* (2.1).

2.3. Extensions and remarks. In this subsection, we discuss extensions of the ADM to solving two closely related TV models. The first one is the constrained alternative of (1.2), i.e.,

$$\min_u \left\{ \sum_i \|D_i u\| \;\middle|\; \|PWu - f\| \le \delta \right\}, \tag{2.11}$$

where $\delta > 0$ is related to the noise level. Note that, from optimization theory, it is easy to see that (1.2) and (2.11) are equivalent in the sense that solving one of them can determine an appropriate parameter for the other such that these two problems share common solutions. Compared with (1.2), the advantage of (2.11) is that δ can be properly selected whenever a reasonable estimation of the noise level is available. For noiseless data, we recover u^* via solving

$$\min_u \left\{ \sum_i \|D_i u\| \;\middle|\; PWu = f \right\}. \tag{2.12}$$

In the following, we briefly sketch the ADM scheme for solving (2.11), where $\delta = 0$ is also permitted and the resulting algorithm solves (2.12).

By introducing auxiliary variables, problem (2.11) is clearly equivalent to

$$\min_{u,v,\mathbf{w}} \left\{ \sum_i \|\mathbf{w}_i\| + I_{\mathcal{V}}(v) \;\middle|\; v = Wu,\ \mathbf{w}_i = D_i u \quad \forall\, i \right\}, \tag{2.13}$$

where $\mathcal{V} \triangleq \{v \in \mathbb{R}^{n^2} : \|Pv - f\| \equiv \|v_P - f\| \le \delta\}$ and $I_{\mathcal{V}}(v)$ is the indicator function of $\mathcal{V}$, i.e.,

$$I_{\mathcal{V}}(v) = \begin{cases} 0 & \text{if } v \in \mathcal{V}, \\ +\infty & \text{otherwise.} \end{cases} \tag{2.14}$$

The augmented Lagrangian function $\mathcal{L}^{\mathcal{I}}_{\mathcal{A}}(u, v, w, \lambda, \eta)$ of (2.13) is given by

$$\begin{aligned} \mathcal{L}^{\mathcal{I}}_{\mathcal{A}}(u, v, w, \lambda, \eta) := \sum\nolimits_i \left(\|\mathbf{w}_i\| - \lambda_i^\top (\mathbf{w}_i - D_i u) + \frac{\beta_1}{2} \|\mathbf{w}_i - D_i u\|^2 \right) \\ + I_{\mathcal{V}}(v) - \eta^\top (v - Wu) + \frac{\beta_2}{2} \|v - Wu\|^2. \end{aligned} \tag{2.15}$$

Given λ^k and η^k, the ADM applied to (2.13) is an iterative algorithm based on the iteration

$$\begin{cases} (v, w)^{k+1} = \arg\min_{v,w} \mathcal{L}^{\mathcal{I}}_{\mathcal{A}}(u^k, v, w, \lambda^k, \eta^k), \\ u^{k+1} = \arg\min_u \mathcal{L}^{\mathcal{I}}_{\mathcal{A}}(u, v^{k+1}, w^{k+1}, \lambda^k, \eta^k), \\ \lambda_i^{k+1} = \lambda_i^k - \gamma\beta_1 (\mathbf{w}_i^{k+1} - D_i u^{k+1}) \quad \forall i, \\ \eta^{k+1} = \eta^k - \gamma\beta_2 (v^{k+1} - Wu^{k+1}). \end{cases} \tag{2.16}$$

The only difference between (2.16) and (2.4) lies in the minimization for v. It is easy to see that the v-subproblem of (2.16) is equivalent to $\min_{v\in\mathcal{V}} \|v-\zeta^k\|$, where $\zeta^k = Wu^k + \eta^k/\beta_2$. Let $\mathcal{P}_\mathcal{V}$ be the projection onto $\mathcal{V}$. Clearly, $v^{k+1} = \mathcal{P}_\mathcal{V}(\zeta^k)$ or, equivalently, $v_{\bar{P}}^{k+1} = \zeta_{\bar{P}}^k$ and

$$v_P^{k+1} = f + \min\left(\|\zeta_P^k - f\|, \delta\right) \frac{\zeta_P^k - f}{\|\zeta_P^k - f\|}. \tag{2.17}$$

Therefore, by replacing the formula for computing v_P^{k+1} in Algorithm 1 by (2.17), we obtain an ADM algorithm for solving (2.13) or, equivalently, (2.11). We note that $v_P^{k+1} \equiv f$ when $\delta = 0$, and the resulting algorithm solves (2.12).

Based on previous discussions, it is easy to see that the ADM framework can be applied to solving (2.11) and (2.12) with minor modifications. In fact, ADM is also easily applicable to many other cases, e.g., local weighted and multichannel TV regularizations, nonnegativity, or simple bound constraints, as well as models with the ℓ_1-norm data-fitting; see, e.g., [56, 59]. The most influential feature of the ADM approach is perhaps its great versatility and universal effectiveness for a wide range of optimization problems in signal, image, and data analysis, particularly for those involving ℓ_1-like regularizations such as TV, ℓ_1-norm, and nuclear-norm (sum of singular values), which have been used in the literature to promote different properties of the original signal. Due to its simplicity and efficiency, very recently ADM has found many applications in diverse areas in addition to image restoration, such as compressive sensing [57], semidefinite programming [54], sparse and low-rank matrix separation [55, 49], and magnetic resonance imaging [59].

3. Numerical results.

In this section, we present experimental results to show the practical efficiency of the proposed ADM scheme. In particular, we compare Algorithm 1 with the optimization transfer algorithm [14] and the two-step IST algorithm [7]. We did not compare with the gradient descent method [20] since it was shown in [14] that it is much slower and less accurate than the optimization transfer algorithm. In the following, we abbreviate the optimization transfer algorithm [14] and the two-step IST algorithm [7] as OpT and TwIST, respectively, while Algorithm 1 will be refereed to as ADM. In all experiments, we initialized $u^0 = W^\top P^\top f$, and we call it "back projection", $\lambda^0 = 0$ and $\eta^0 = 0$. The quality of recovered images is measured by signal-to-noise ratio (SNR), defined by

$$\mathrm{SNR} = 20\log_{10}\frac{\|u^*\|}{\|u-u^*\|},$$

where u^* and u are respectively the original and the reconstructed images.

3.1. Comparison with OpT.

In [14], the authors proposed to solve (1.2) by an optimization transfer or majorization algorithm. First, problem (1.2) was converted to an equivalent problem of the form

$$\min_{u,v}\left\{\mathrm{TV}(u) + \frac{\mu(1+\tau)}{2\tau}\left(\|Pv-f\|^2 + \tau\|v-Wu\|^2\right)\right\}, \tag{3.1}$$

where $\tau > 0$ is a parameter. It can be shown that

$$\|PWu-f\|^2 = \min_v \frac{1+\tau}{\tau}\left(\|Pv-f\|^2 + \tau\|v-Wu\|^2\right). \tag{3.2}$$

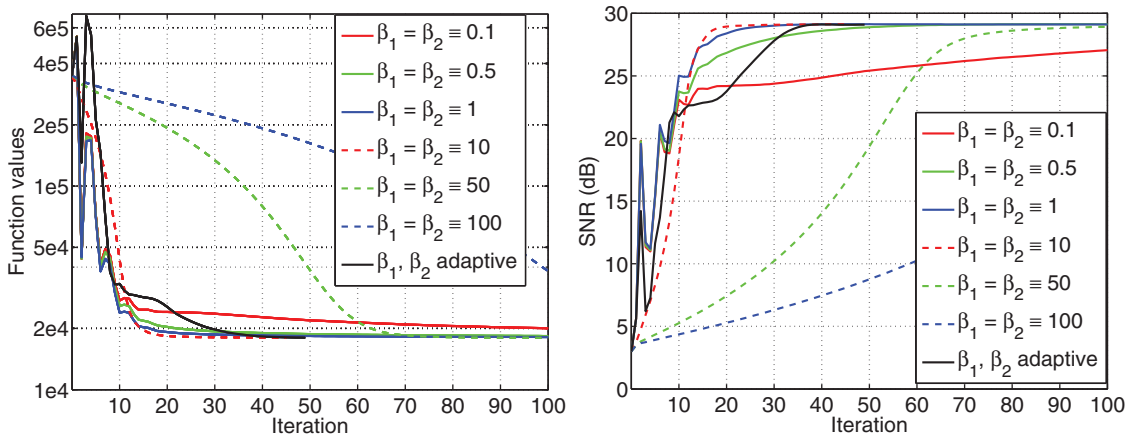

Figure 1. *Convergence of ADM with different β values. Test image: Boat (the second image in Figure 6).*

Therefore, (3.1) is equivalent to (1.2). The advantage of considering (3.1) is that alternating minimization can be applied. For fixed u, the minimization of (3.1) with respect to v has a closed-form solution. On the other hand, for fixed v the minimization of (3.1) with respect to u is a TV denoising problem (because $\|Wu - v\|^2 = \|u - W^\top v\|^2$), and therefore it can be solved efficiently by Chambolle's dual approach [12].

In this comparison, we tested three images of different sizes: Cameraman (256-by-256), Barbara (512-by-512), and Man (1024-by-1024). The intensity values of the original images are scaled into [0,1] before generating f. In all experiments, we corrupted each original image with random Gaussian noise of mean zero and standard deviation (std.) of 0.0392 (equivalent to std. = 10 for images with intensity values varying between 0 and 255), and the model parameter μ was set to be 50. We used the Daubechies 7-9 biorthogonal wavelets with symmetric extensions at the boundaries [4, 21]. For each image, we tested the algorithms with 30%, 50%, and 70% randomly selected wavelet coefficients.

The parameter settings of ADM are as follows. We set $\gamma = 1.618$ in all experiments because, based on our data, ADM is not sensitive to its choice. Although fixed values of β_1 and β_2 suffice for convergence, we chose to select them adaptively. Since β_1 and β_2 are penalty parameters, we determine them in such a way that the variations of the constraints $w = Du$ and $v = Wu$ are balanced. Specifically, for given β_1 we determine β_2 by

$$\beta_2 = \min\left\{\beta_1 \times \frac{\|w - Du\|}{\|v - Wu\|}, 2 \times 10^4\right\}. \tag{3.3}$$

We initialize $\beta_1^0 = 0.1$ and set $\beta_1^k = \min\{1.15\beta_1^{k-1}, 2 \times 10^3\}$ at the kth iteration. We note that this dynamic parameter selection rule does not spoil the convergence theory of ADM; see, e.g., [36] for a study of ADM with adaptive penalty parameters. In fact, the practical performance of ADM is not very sensitive to the choices of β_1 and β_2, provided that they are not extremely large or small. To illustrate this point, we tested several constant values of β_1 and β_2. The convergence of ADM with constant β-values, as well as the adaptive rule (3.3) described above, in terms of function values and SNR values are given in Figure 1.

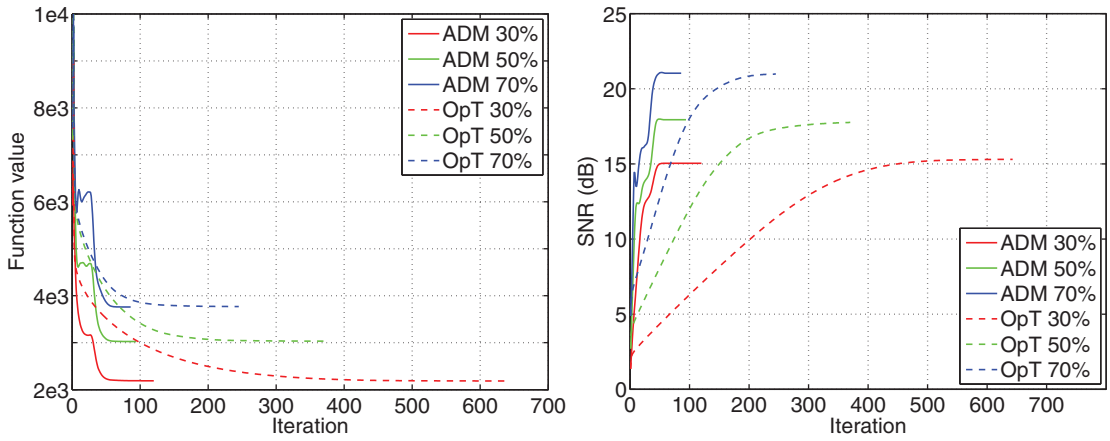

Figure 2. *Comparison results for the ADM and OpT algorithms about the convergence behavior of function values and SNR values. Test image: Cameraman;* $p/n^2 = 30\%, 50\%,$ *and* 70%.

It can be seen from Figure 1 that the convergence speed of ADM is not very sensitive to the values of β_1 and β_2 as long as they are relatively not too large or small. In particular, for the tested problem, β_1 and β_2 can be randomly selected between, say, 0.5 and 30, and the corresponding iteration numbers differ slightly. In fact, ADM with the self-adaptive rule (3.3) is less sensitive to the scale of problem data, and the suitable values for β_1 and β_2 can be found simultaneously.

In all experiments, we terminated Algorithm 1 when the relative change between two consecutive points became small, i.e.,

$$\frac{\|u^k - u^{k-1}\|}{\|u^{k-1}\|} \leq tol, \tag{3.4}$$

where $tol > 0$ is a tolerance. For the OpT algorithm, we used the same settings as in [14]. All the experiments were performed under Windows XP and MATLAB v7.9 (R2009b) running on a Dell desktop with an Intel Xeon CPU at 3GHz and 3Gb of memory.

To examine carefully the convergence behavior of ADM, we first tested the Cameraman image with different percentages of available data. The ADM is terminated by (3.4) with relatively stringent tolerance: $tol = 10^{-5}$. The decrease of function values and the increase of SNR values as functions of iteration numbers are illustrated in Figure 2, in comparison with the OpT algorithm [14]. Note that in this test we set the maximum allowed iteration number to be 10 for each call of Chambolle's dual algorithm for solving the TV denoising subproblem in OpT.

It can be seen form Figure 2 that ADM converges much faster than OpT. Specifically, for all three tests ADM requires about $50 \sim 60$ iterations to reach the lowest function values and the highest SNR values achievable by the model. In contrast, OpT improves the solution quality continuously but at a much slower speed. OpT takes about 250 iterations at least for the case of 70% available data. For the case of 30% available data, OpT takes more than 600 iterations to recover a solution of approximately the same quality as that obtained by the ADM.

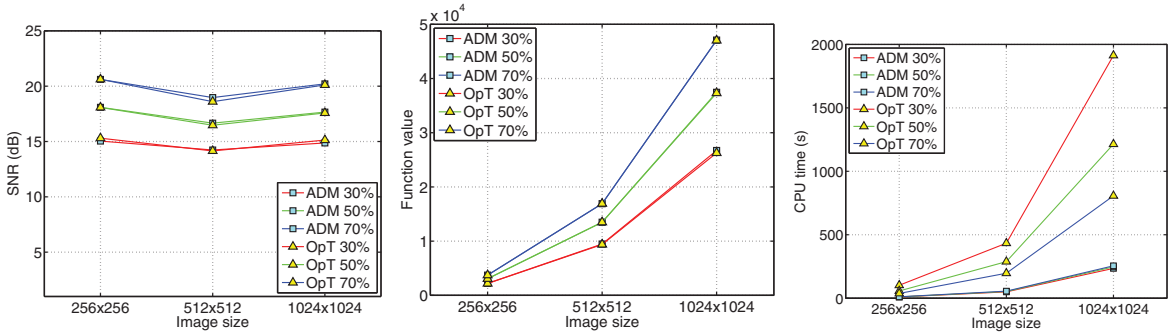

Figure 3. *Comparison results of SNR values, function values, and CPU times.*

Given the above observations, in the next set of experiments we terminated ADM with a looser tolerance: $tol = 5 \times 10^{-4}$. In this set of experiments, we tested all three images with three levels of available data (30%, 50%, and 70%). The results of final objective function values, SNR values, and the consumed CPU time are summarized in Figure 3.

From the first and the second plots in Figure 3, we can see that, for all tests, the final SNR values and function values obtained by both algorithms are approximately the same, because they solve the same model. However, from the third plot in Figure 3 it is clear that ADM converges much faster than OpT. The faster convergence of ADM over OpT becomes more evident when image size becomes large. For example, for the 1024-by-1024 "Man" image, OpT takes up to $1,900$ seconds for the test on 30% data, while ADM takes only takes about 250 seconds. Another advantage of ADM over OpT is that its speed seems less sensitive to the amount of available data. The CPU time consumed by ADM remains roughly the same for the tests on 30%, 50%, and 70% data, while that taken by OpT becomes longer for less data. These comparison results clearly demonstrate the superiority of the proposed ADM approach. Recall that OpT applies the method in [12] to solve the resulting TV denoising subproblem at each iteration. In this experiment, we set the maximal number of iterations as 5 for the inner iterations, as suggested in [14]. If the TV denoising subproblems are solved to higher accuracy, OpT becomes even slower. The original images, back projections ($W^\top P^\top f$), and those images recovered by both algorithms from 50% wavelet coefficients are presented in Figure 4 along with the resulting SNR values and the consumed CPU times.

Now, we dig into the reason why OpT is slower than ADM. It is easy to show that, for given $u = u^k$, the minimizer v^{k+1} of (3.2) is given by

$$\begin{cases} v_{\bar{P}}^{k+1} = W_{\bar{P}} u^k, \\ v_P^{k+1} = \frac{f + \tau W_P u^k}{1+\tau}, \end{cases} \tag{3.5}$$

where W_P is the submatrix of W containing those rows with indices selected by P, and $W_{\bar{P}}$ contains the remaining rows. By plugging v^{k+1} into (3.1) and ignoring constant quantities, we see that u^{k+1} generated by the OpT algorithm in [14] is the unique solution of

$$\min_u \left\{ \mathrm{TV}(u) + \frac{\mu(1+\tau)}{2} \|u - \xi^k\|^2 \right\}, \tag{3.6}$$

Figure 4. *Images recovered by both algorithms from 50% wavelet coefficients. First row: original images. Second row: back projections. Third row: recovered by OpT. Bottom row: recovered by ADM.*

where $\xi^k = W^\top v^{k+1}$. From (3.5) and $W_P^\top W_P + W_{\bar{P}}^\top W_{\bar{P}} = I$, it is easy to show that

$$\begin{aligned}\xi^k = W^\top v^{k+1} &= W_P^\top v_P^{k+1} + W_{\bar{P}}^\top v_{\bar{P}}^{k+1} \\ &= \frac{W_P^\top f + \tau W_P^\top W_P u^k}{1+\tau} + W_{\bar{P}}^\top W_{\bar{P}} u^k \\ &= \frac{1}{1+\tau}\left(W_P^\top f + \tau u^k + W_{\bar{P}}^\top W_{\bar{P}} u^k\right) \\ &= u^k - \left(u^k - \frac{1}{1+\tau}\left(W_P^\top f + \tau u^k + W_{\bar{P}}^\top W_{\bar{P}} u^k\right)\right) \\ &= u^k - \frac{1}{1+\tau} W_P^\top \left(W_P u^k - f\right) = u^k - \delta g^k, \end{aligned} \tag{3.7}$$

where $\delta = \frac{1}{1+\tau}$ and $g^k = W_P^\top (W_P u^k - f)$ is the gradient of $\frac{1}{2}\|PWu - f\|^2$ at $u = u^k$. From (3.6) and (3.7), it is clear that the optimization transfer algorithm proposed in [14] is essentially a proximal forward-backward operator splitting algorithm with the constant steplength $\frac{1}{1+\tau}$, which is strictly less than 1 since $\tau > 0$. We note that in general the proximal forward-backward operator splitting algorithm converges for varying steplengths δ^k, provided that the sequence $\{\delta^k\}_{k=1}^{+\infty}$ satisfies

$$0 < \inf_k\{\delta^k\} \le \sup_k\{\delta^k\} < 2/\lambda_{\max}(AA^\top),$$

where, under the circumstances, $A = PW$ for problem (1.2) and $\lambda_{\max}(\cdot)$ denotes the maximum eigenvalue. We also note that in general bigger steplength δ^k leads to faster convergence. For $A = PW$, it holds that $\lambda_{\max}(AA^\top) = \lambda_{\max}(PP^\top) = 1$. Therefore, according to the convergence theory of the proximal forward-backward operator splitting algorithm, the steplength δ in (3.7) can be relaxed to a value as large as 2 without spoiling its convergence. On the other hand, the ADM is a variant of the classical ALM, which has a close relationship with Newton's method. In fact, recent studies on semidefinite programs conducted in [48] show that the ALM can be locally regarded as an approximate generalized Newton method applied to a semismooth equation. This partially explains why the OpT algorithm converges more slowly than does ADM. For details about the proximal forward-backward operator splitting algorithm, we refer interested readers to [22] and the references therein.

3.2. Comparison with TwIST.

Given the above observations, we next compare the ADM with TwIST [7], which is actually a variant of the proximal forward-backward operator splitting algorithm. Note that TwIST solves the general problem:

$$\min_u \left\{\Phi_{\text{reg}}(u) + \frac{\mu}{2}\|Au - b\|^2\right\}, \tag{3.8}$$

where $\Phi_{\text{reg}}(\cdot)$ can be either TV or ℓ_1 regularization. Given u^k, we let $\xi^k = u^k - A^\top(Au^k - b)$ and

$$\Psi(\xi^k) := \arg\min_u \left\{\Phi_{\text{reg}}(u) + \frac{\mu}{2}\|u - \xi^k\|^2\right\}. \tag{3.9}$$

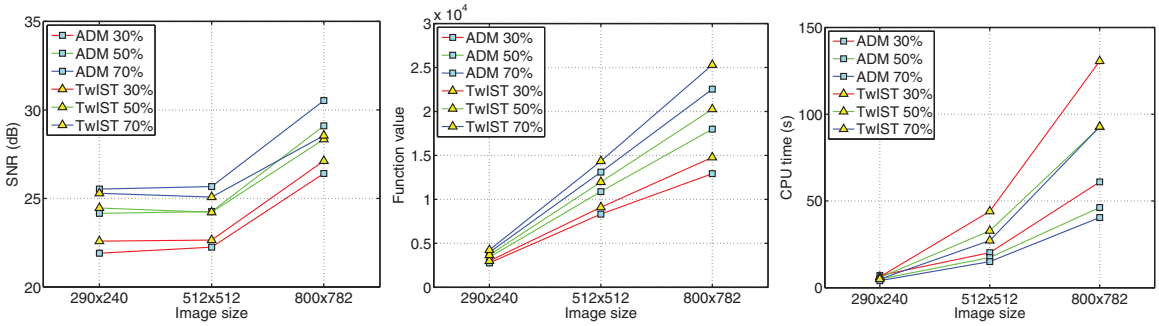

Figure 5. *Comparison results of SNR values, function values, and CPU times.*

Initialized at u^0 and u^1, the TwIST algorithm iterates as

$$u^{k+1} = (1-\alpha)u^{k-1} + (\alpha-\theta)u^k + \theta\Psi(\xi^k), \tag{3.10}$$

where $\alpha, \theta > 0$ are properly selected constant parameters. It is easy to see that (3.10) reduces to (3.6) (with $\delta = 1$ in (3.7)) if $\Phi_{\text{reg}}(u) = \text{TV}(u)$ and $\alpha = \theta = 1$. In the implementation of TwIST, the parameters α and θ were determined carefully based on the spectral distribution of $AA^\top$. In our case, $A = PW$, and the minimum and the maximum eigenvalues of $AA^\top$ are obviously 0 and 1, respectively. Therefore, we assigned a relatively small value 10^{-4} to the TwIST parameter `lam1` (which is used to compute α and θ), as recommended in the TwIST_v2 documentation. In the TwIST codes, problem (3.9) is also solved by Chambolle's denoising algorithm [12]. To speed up TwIST, we set the maximum iteration number to be 5 for each call of Chambolle's algorithm to solve (3.9). TwIST is terminated when the relative change in function values of two consecutive iterations falls below `tolA`$= 10^{-5}$. For ADM, we set $tol = 10^{-4}$ and kept other parameter settings as used in section 3.1.

In this set of experiments, we used the Haar wavelet transform provided by the Rice Wavelet Toolbox [62] with its default settings. The test results on three other images of different sizes (Boy: 290×240, Boat: 512×512, and Bird: 800×782) with 30%, 50%, and 70% data are reported in Figure 5.

It can be seen from the left plot in Figure 5 that ADM reaches lower SNR values than does TwIST for four of the nine tests. However, from the middle and the right plots, ADM obtains smaller function values within less CPU times for all nine tests. Also, by comparing the last plot in Figure 5 with that in Figure 3, we can see that TwIST is faster than OpT since the gap in CPU time is narrowed to a large extent. The original images, back projections, and images recovered by ADM and TwIST from 50% Haar wavelet coefficients are given in Figure 6.

The convergence of ADM and TwIST in terms of function values and SNR values for the 512×512 sized Boat image are plotted in Figure 7. It can be seen from the left plot in Figure 7 that ADM is much faster than TwIST in decreasing function values. For TwIST to obtain function values as good as those results of ADM, much stringent tolerance must be enforced. In that case, the consumed iteration numbers by TwIST will increase greatly, which is not required since the SNR values obtained by both methods are, roughly speaking, equally good, as shown by the plot on the right-hand side in Figure 7.

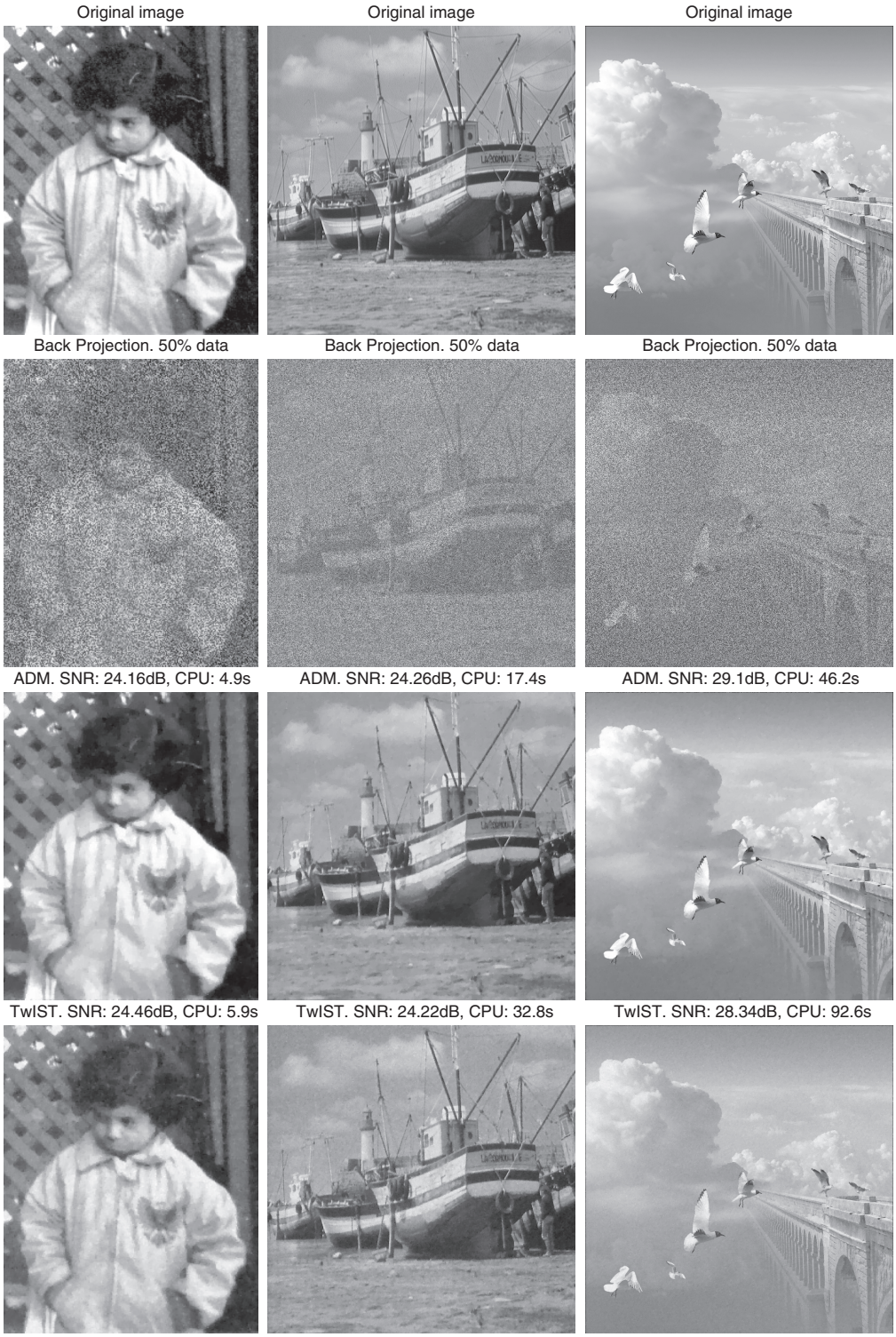

Figure 6. *Results recovered by ADM and TwIST from* 50% *Haar wavelet coefficients. First row: original images (from left to right: Boy, Boat, and Bird). Second row: back projections. Third row: recovered by ADM. Bottom row: recovered by TwIST.*

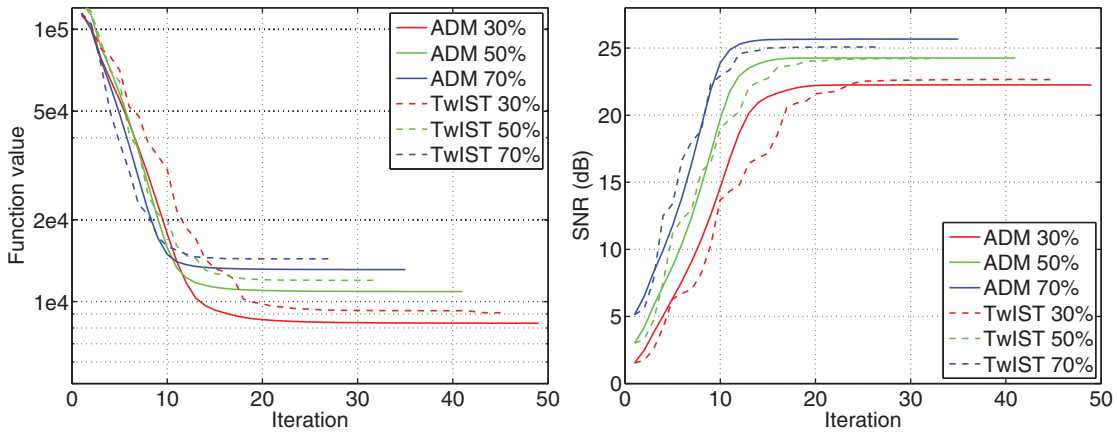

Figure 7. *Convergence of ADM and TwIST in terms of function values and SNR values for the Boat image;* $p/n^2 = 30\%, 50\%$ *and* 70%.

4. Concluding remarks. In this paper, we proposed a fast alternating direction algorithm for solving the TV wavelet domain inpainting problem. The proposed algorithm is a variant of the classical augmented Lagrangian method which takes full advantage of the separable structures of the problem. At each iteration, the main computational cost of the proposed algorithm is dominated by two DWTs and two FFTs. Upon profiling the ADM algorithm in MATLAB, we found that about 60% of CPU time was consumed by the computation of fast transforms (DWT, FFT, and their inverse transforms). The remaining time was consumed by the computation of finite differences, function values, SNR values, overhead, etc. The CPU time consumed by fast (forward and inverse) wavelet transforms versus that by (forward and inverse) Fourier transforms is determined by the average speed of these transforms, since the number of calls to each of the transforms is equal (ignoring overhead computations). Our experimental results clearly demonstrate that the proposed ADM is stable, efficient, and much faster than some existing methods, including the optimization transfer algorithm [14] and the two-step iterative/shrinkage algorithm [7].

Acknowledgment. We are grateful to three anonymous referees for their valuable comments and suggestions which have helped us improve the presentation of this paper.

REFERENCES

[1] R. ACAR AND C. R. VOGEL, *Analysis of total variation penalty methods*, Inverse Problems, 10 (1994), pp. 1217–1229.

[2] M. V. AFONSO, J. BIOUCAS-DIAS, AND M. FIGUEIREDO, *Fast Image Recovery Using Variable Splitting and Constrained Optimization*, IEEE Trans. Image Processing, 19 (2010), pp. 2345–2356.

[3] M. V. AFONSO, J. BIOUCAS-DIAS, AND M. FIGUEIREDO, *A Fast Algorithm for the Constrained Formulation of Compressive Image Reconstruction and Other Linear Inverse Problems*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP'2010, Dallas, TX, 2010.

[4] M. ANTONINI, M. BARLAUD, P. MATHIEU, AND I. DAUBECHIES, *Image coding using wavelet transform*, IEEE Trans. Image Process., 1 (1992), pp. 205–220.

[5] C. BALLESTER, M. BERTALMIO, V. CASELLES, G. SAPIRO, AND J. VERDERA, *Filling-in by joint interpolation of vector fields and gray levels*, IEEE Trans. Image Process., 10 (2001), pp. 1200–1211.

[6] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in SIGGRAPH 2000, Computer Graphics Proceedings, K. Akeley, ed., ACM Press/ACM SIGGRAPH/Addison Wesley Longman, New York, 2000, pp. 417–424.
[7] J. Bioucas-Dias and M. Figueiredo, *A new TwIST: Two-step iterative thresholding algorithm for image restoration*, IEEE Trans. Image Process., 16 (2007), pp. 2992–3004.
[8] L. Bregman, *The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex optimization*, USSR Comput. Math. Math. Phys., 7 (1967), pp. 200–217.
[9] J.-F. Cai, R. Chan, and Z. Shen, *A framelet-based image inpainting algorithm*, Appl. Comput. Harmon. Anal., 24 (2008), pp. 131–149.
[10] J.-F. Cai, H. Ji, F. Shang, and Z. Shen, *Inpainting for compressed images*, Appl. Comput. Harmon. Anal., 29 (2010), pp. 368–381.
[11] A. Chambolle and P. L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.
[12] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.
[13] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.
[14] R. Chan, Y. W. Wen, and A. M. Yip, *A fast optimization transfer algorithm for image inpainting in wavelet domains*, IEEE Trans. Image Process., 18 (2009), pp. 1467–1476.
[15] T. Chan, S. Esedoglu, F. Park, and A. Yip, *Recent Developments in Total Variation Image Restoration*, CAM Report TR05–01, Los Angeles, CA, 2005.
[16] T. F. Chan, S. H. Kang, and J. Shen, *Euler's elastica and curvature-based inpainting*, SIAM J. Appl. Math., 63 (2002), pp. 564–592.
[17] T. Chan and J. Shen, *Nontexture inpainting by curvature-driven diffusions*, J. Vision Commun. Image Represent., 12 (2001), pp. 436–449.
[18] T. F. Chan and J. Shen, *Mathematical models for local nontexture inpaintings*, SIAM J. Appl. Math., 62 (2002), pp. 1019–1043.
[19] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM, Philadelphia, 2005.
[20] T. Chan, J. Shen, and H. Zhou, *Total variation wavelet inpainting*, J. Math. Imaging Vision, 25 (2006), pp. 107–125.
[21] A. Cohen, I. Daubeches, and J. C. Feauveau, *Biorthogonal bases of compactly supported wavelets*, Comm. Pure Appl. Math., 45 (1992), pp. 485–560.
[22] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4 (2005), pp. 1168–1200.
[23] I. Daubechies, M. Defriese, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. Math., LVII (2004), pp. 1413–1457.
[24] C. De Mol and M. Defrise, *A note on wavelet-based inversion algorithms*, Contemp. Math., 313 (2002), pp. 85–96.
[25] J. Douglas and H. Rachford, *On the numerical solution of heat conduction problems in two and three space variables*, Trans. Amer. Math. Soc., 82 (1956), pp. 421–439.
[26] D. C. Dobson and F. Santosa, *Recovery of blocky images from noisy and blurred data*, SIAM J. Appl. Math., 56 (1996), pp. 1181–1198.
[27] J. Eckstein and D. Bertsekas, *On the Douglas-Rachford Splitting Method and the Proximal Point Algorithm for Maximal Monotone Operators*, Math. Program. 55, North–Holland, Amsterdam, 1992.
[28] M. Elad, *Why simple shrinkage is still relevant for redundant representations?*, IEEE Trans. Inform. Theory, 52 (2006), pp. 5559–5569.
[29] E. Esser, *Applications of Lagrangian-based Alternating Direction Methods and Connections to Split Bregman*, CAM Report 09–31, UCLA, Los Angeles, CA, 2009.
[30] M. Figueiredo and R. Nowak, *An EM algorithm for wavelet-based image restoration*, IEEE Trans. Image Process., 12 (2003), pp. 906–916.
[31] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976), pp. 17–40.

[32] D. Geman and C. Yang, *Nonlinear image recovery with half-quadratic regularization*, IEEE Trans. Image Process., 4 (1995), pp. 932–946.
[33] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator Splitting Methods in Nonlinear Mechanics*, SIAM Stud. Appl. Math. 9, SIAM, Philadelphia, 1989.
[34] R. Glowinski and A. Marrocco, *Sur lapproximation par elements finis dordre un, et la resolution par penalisation-dualite dune classe de problemes de Dirichlet nonlineaires*, Rev. Francaise Automat. Inform. Rech. Oper., R-2 (1975), pp. 41–76.
[35] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.
[36] B. S. He, S. L. Wang, and H. Yang, *A modified variable-penalty alternating directions method for monotone variational inequalities*, J. Comput. Math., 21 (2003), pp. 495–504.
[37] M. R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.
[38] M. K. Ng, R. H. Chan, and W.-C. Tang, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM J. Sci. Comput., 21 (1999), pp. 851–866.
[39] M. K. Ng, P. Weiss, and X. Yuan, *Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods*, SIAM J. Sci. Comput., 32 (2010), pp. 2710–2736.
[40] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489.
[41] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1969, pp. 283–298.
[42] S. Rane, J. Remus, and G. Sapiro, *Wavelet-domain reconstruction of lost blocks in wireless image transmission and packet-switched networks*, in Proceedings of the IEEE International Conference in Image Processing, Rochester, NY, 2002, pp. 309–312.
[43] S. Rane, G. Sapiro, and M. Bertalmio, *Structure and texture filling-in of missing image blocks in wireless transmission and compression applications*, IEEE Trans. Image Process., 12 (2003), pp. 296–303.
[44] L. Rudin and S. Osher, *Total variation based image restoration with free local constraints*, in Proceedings of the 1st IEEE International Conference on Image Processing, Austin, TX, 1994, pp. 31–45.
[45] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[46] S. Setzer, *Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage*, in Proceedings of the 2nd International Conference on Scale Space Methods and Variational Methods in Computer Vision, Lecture Notes in Comput. Sci. 5567, Springer, New York, 2009, pp. 464–476.
[47] J. L. Starck, M. Nguyen, and F. Murtagh, *Wavelets and curvelets for image deconvolution: A combined approach*, Signal Process., 83 (2003), pp. 2279–2283.
[48] D. F. Sun, J. Sun, and L. W. Zhang, *The rate of convergence of the augmented Lagrangian method for nonlinear semidefinite programming*, Math. Program., 114 (2008), pp. 349–391.
[49] M. Tao and X. Yuan, *Recovering low-rank and sparse components of matrices from incomplete and noisy observations*, SIAM J. Optim., 21 (2011), pp. 57–81.
[50] A. Tikhonov and V. Arsenin, *Solution of Ill-Posed Problems*, Winston, Washington, DC, 1977.
[51] C. R. Vogel and M. E. Oman, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput., 17 (1996), pp. 227–238.
[52] C. R. Vogel and M. E. Oman, *A fast, robust total variation based reconstruction of noisy, blurred images*, IEEE Trans. Image Process., 7 (1998), pp. 813–824.
[53] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 248–272.
[54] Z. Wen, W. Yin, and D. Goldfarb, *Alternating direction augmented Lagrangian methods for semidefinite programming*, Math. Program. Comput., 2 (2010), pp. 203–230.
[55] Z. Lin, M. Chen, L. Wu, and Y. Ma, *The augmented Lagrange multiplier method for exact recovery of a corrupted low-rank matrices*, Math. Prog., submitted.
[56] J. Yang, W. Yin, Y. Zhang, and Y. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, SIAM J. Imaging Sci., 2 (2009), pp. 569–592.
[57] J. Yang and Y. Zhang, *Alternating direction algorithms for ℓ_1-problems in compressive sensing*, SIAM J. Sci. Comput., 33 (2011), pp. 250–278.

[58] J. Yang, Y. Zhang, and W. Yin, *An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise*, SIAM J. Sci. Comput., 31 (2009), pp. 2842–2865.

[59] J.-F. Yang, Y. Zhang, and W. Yin, *A fast alternating direction method for TVL1-L2 signal reconstruction from partial Fourier data*, IEEE J. Sel. Top. Signal Process., 4 (2010), pp. 288–297.

[60] X. Q. Zhang and T. Chan, *Wavelet inpainting by nonlocal total variation*, Inverse Problems and Imaging, 4 (2010), pp. 191–210.

[61] X. Zhang, M. Burger, X. Bresson, and S. Osher, *Bregmanized nonlocal regularization for deconvolution and sparse reconstruction*, SIAM J. Imag. Sci., 3 (2010), pp. 253–276.

[62] R. Baraniuk, H. Choi, R. Neelamani, V. Ribeiro, J. Romberg, H. Guo, F. Fernandes, B. Hendricks, R. Gopinath, M. Long, J. E. Odegard, and D. Wei, *Rice Wavelet Toolbox*, Version 2.4, 2002, http://www.dsp.rice.edu/software/rice-wavelet-toolbox.