SIAM J. IMAGING SCIENCES
Vol. 9, No. 1, pp. 24–51

A New Algorithm Framework for Image Inpainting in Transform Domain*

Fang Li† and Tieyong Zeng‡

Abstract. In this paper, we focus on variational approaches for image inpainting in transform domain and propose two new algorithms, iterative coupled transform domain inpainting (ICTDI) and iterative decoupled transform domain inpainting. In the derivation of ICTDI, we use operator splitting and the quadratic penalty technique to get a new approximate problem of the basic model. By the alternating minimization method, the approximate problem can be decomposed as three relatively simple subproblems with closed-form solutions. However, ICTDI is not efficient when some adaptive regularization operator is used, such as the learned BM3D frame. To overcome this drawback, with some modifications, we decouple our framework into three relatively independent parts: denoising, linear combination in the transform domain, and linear combination in the image domain. Therefore, we can use any existing denoising method in the denoising step. We consider three choices for regularization operators in our approach: gradient operator, tight framelet transform, and learned BM3D frame. The numerical experiments and comparisons on various images demonstrate the effectiveness of the proposed methods. The convergence of the numerical algorithms is proved under some assumptions.

Key words. image inpainting, transform domain, shrinkage, BM3D frame

AMS subject classifications. 68U10, 94A08, 80M30, 65K10

DOI. 10.1137/15M1015169

1. Introduction.

Image inpainting is the procedure of recovering lost or deteriorated parts of images and videos. More precisely, the problem of image inpainting occurs when the observed data is incomplete in the sense that some pixels or coefficients of the target image under a certain transform are missing or corrupted [2, 48]. The main challenge of this important task is to find a solution having edges, structures, and texture patterns consistent with the given data [8].

Many useful techniques have been proposed in recent years to address the image inpainting task, which can be roughly classified into two categories: image domain inpainting and transform domain inpainting [48]. Here image domain inpainting means that some data in the image domain is missing or, equivalently, some pixels are missing. Image domain inpainting has wide application in text and scratch removal in ancient drawings or old pictures, removal of objects in photography or films for special effects, recovering lost pixels damaged in image coding and transmission, etc. [2, 17]. On the other hand, transform domain inpainting means that

*Received by the editors April 2, 2015; accepted for publication (in revised form) October 26, 2015; published electronically January 12, 2016. This work is supported by the 973 Program 2011CB707104, the Science and Technology Commission of Shanghai Municipality (STCSM) 13dz2260400, NSFC 11271049, RGC 211911, 12302714, and RFGs of HKBU.

http://www.siam.org/journals/siims/9-1/M101516.html

†Corresponding author. Department of Mathematics, East China Normal University, and Shanghai Key Laboratory of Pure Mathematics and Mathematical Practice, Shanghai, China (fli@math.ecnu.edu.cn).

‡Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (zeng@hkbu.edu.hk).

some coefficients in a certain transform domain are missing. Transformed domain inpainting arises in practical applications because images are usually formatted, transmitted, and stored in a transformed domain. For instance, JPEG standard images are encoded in terms of discrete cosine transform coefficients and JPEG 2000 standard images are encoded by wavelet transform coefficients, and in magnetic resonance (MR) imaging the acquired data is Fourier transform coefficients [12, 14]. Certain coefficients may be lost or corrupted during the process of storage and transmission and this leads to the transformed domain inpainting problem.

Image domain inpainting has been widely studied in literature in has past decades. One important stream for this issue is the pixel based method, including variational and partial differential equation (PDE) based methods, and the sparsity driven method. In the PDE method, the missing region is filled by diffusing the image information from the known region to the missing region [2, 1, 41, 42, 3, 43]. In the sparsity based method, the image is represented by a sparse combination of a set of transforms such as wavelet or tight frame, and then the missing pixels can be filled by adaptively updating the sparse representation [23, 27, 8, 24, 49, 39, 34]. The other category is the exemplar based inpainting method, in which the image information in the known region propagates into the missing region patch by patch [17, 46, 47].

Transform domain inpainting has also received remarkable attention in variational approaches. Indeed, currently the main existing works have been focused on wavelet domain inpainting and Fourier domain inpainting. The first variational wavelet inpainting method was proposed by Chan, Shen, and Zhou [13]. As a counterpart of the image domain total variation (TV) inpainting model in [41], Chan, Shen, and Zhou propose the TV wavelet inpainting model for filling in the missing coefficients in the wavelet domain. They use the terminology "wavelet inpainting" to address that the missing data is in the wavelet transform domain. Numerically, they evolve the associate PDE in the wavelet domain which converges rather slowly. In the literature, many fast numerical methods have been proposed to handle the TV denoising/deblurring problem—for instance, Chambolle's fast dual projection algorithm [9], the fast TV deconvolution algorithm [44], the split Bregman method [26], the primal dual method [10, 20], and so on [6]. They are applied to solve the TV wavelet inpainting problem and get efficient algorithms. Chan, Win, and Yip [11] propose a fast optimization transfer algorithm (OTA). Based on the variable splitting method, the problem is split into a standard TV denoising subproblem and another simple problem with a closed-form solution. Chambolle's fast dual projection algorithm [9] is then used to solve the TV denoising problem. Later Chan, Yang, and Yuan [12] proposed to use the alternating direction method (ADM) to solve the TV wavelet inpainting model in which two extra variables are introduced. It is reported that ADM is more efficient than OTA with similar image inpainting quality. Wen, Chan, and Yip [45] propose a primal-dual type numerical algorithm to solve this problem and convergence is proved. Another primal-dual hybrid gradient method is proposed in [48] by Ye, Sapiro, and Mallet. Nonlocal TV (NLTV) regularization is considered in wavelet inpainting by Zhang and Chan. [51].

Fourier domain inpainting has been widely addressed in the MR imaging problem such as partial parallel imaging. Sometimes, it is also termed compressed sensing. Since the wavelet transform and the Fourier transform have quite different properties, inpainting in these two domains is also different in many aspects. Hence, wavelet domain inpainting and Fourier domain inpainting are commonly not addressed simultaneously in a paper. Moreover,

the TV Fourier inpainting model is also a basic model for this issue. To solve this model, Goldstein and Osher [26] propose to use the split Bregman method and Chen et al. [14] propose two fast algorithms based on the primal-dual hybrid gradient method. Some other types of regularization have also been considered. Ma et al. [36] propose a model with both TV and wavelet regularization and an efficient numerical algorithm is deduced following the operator splitting technique. Guo, Qin, and Yin [28] propose a total generalized variation and shearlet based regularization scheme, and the alternating direction method of multiplier (ADMM) is used to derive the algorithm. Zhang et al. [50] propose to use NLTV as the regularization term and build the algorithm based on Bregmanized operator splitting.

In this paper, we generalize our previous work [34] built for pixel domain inpainting to transform domain inpainting. We extend the existing TV transform domain inpainting model and propose new numerical algorithms based on variable splitting and the quadratic penalty method. By utilizing the variable splitting method, we get a relaxed minimization problem with two auxiliary variables. The two variables and the latent image can be efficiently solved by the alternating minimization method. Then we get a general framework called the iterative coupled transform domain inpainting (ICTDI) algorithm. For some special transform such as the learned BM3D frame, ICTDI is not so efficient since an ill-conditioned subproblem is involved. Inspired by the decoupling idea in [19, 34], we make a similar modification and get the iterative decoupled transform domain inpainting (IDTDI) algorithm. In the decoupled algorithm, the iteration scheme is decoupled into three steps, which come from minimizing two different energies. One step is denoising. The other two steps are linear combinations in the image domain and the transform domain, respectively. In such a framework, the denoising step becomes crucial. As we know, image denoising is far more widely studied than transform domain inpainting. So we can use the existing best denoising method in this step such as the overcomplete dictionary learning method or learned BM3D filter. Henceforth, we can get the state-of-the-art inpainting results. An additional advantage of the framework of our algorithm is that it can be readily generalized to other important image processing tasks such as image decompression, which will be our future work.

The contribution of our paper is clear. First, we propose a new algorithm framework (ICTDI and IDTDI) for the transform domain image inpainting problem by considering many existing popular regularization operators. Second, the design of the decoupled algorithm provides a natural approach to integrate the BM3D filter which can greatly improve the image inpainting quality. The idea of decoupling can be readily extended to other image tasks.

The paper is organized as follows. In section 2, we give the general model and propose two new algorithms for transform domain inpainting. Some mathematical results are proved in section 3. The experiments and comparisons with the previous algorithms are performed in section 4. Finally, we conclude this work in section 5.

2. The proposed method.

In this section, we extend the existing model for transform domain inpainting by considering more general regularization and propose two new algorithms to solve it which depend on the choice of regularization operator.

2.1. The model.

We will denote images as vectors by stacking their columns. Suppose $u \in R^N$ is the image to be recovered, where N denotes the number of pixels. The general

image inpainting problem can be formulated as [48]

$$f = \mathcal{P}\mathcal{T}u + n, \tag{2.1}$$

where $\mathcal{T} \in R^{N\times N}$ is the transform under which the data is acquired, $\mathcal{P} \in R^{m\times N}$ is the binary selection/downsampling matrix containing $m < N$ rows of the identity matrix of order N, n is the additive noise, and $f \in R^m$ is the acquired data in transform domain which is incomplete and our aim is to recover the missing/damaged data. When $\mathcal{T}$ is the identity matrix, it becomes the image domain inpainting model. In this paper, we focus on two cases where $\mathcal{T}$ is a wavelet transform or Fourier transform which satisfies

$$\mathcal{T}^T\mathcal{T} = \mathcal{I}. \tag{2.2}$$

Here the superscript ^{T} denotes the conjugate operator. Note that the algorithm framework in this section also holds for the general transform domain. Assume that $\Phi \in R^{M\times N}$ is a given transform matrix corresponding to some regularization operator. To fill in the missing transform coefficients, we consider the general model

$$\min_u \|\Phi u\|_p + \frac{\lambda}{2}\|\mathcal{P}\mathcal{T}u - f\|_2^2, \tag{2.3}$$

where λ is a positive regularization parameter, $p = 0$ or 1. When $\Phi = \nabla, p = 1$, it becomes the widely studied TV inpainting model for transform domain inpainting.

The first term in model (2.3) is a regularization term which requires that the representation of Φu is sparse. The second term is a data fitting term, which requires that the subsampled transformed coefficient $\mathcal{P}\mathcal{T}u$ should be close to the given data f.

The model in (2.3) can be easily generalized to another problem such as deblurring and transform domain inpainting, in which the model is

$$\min_u \|\Phi u\|_p + \frac{\lambda}{2}\|\mathcal{P}\mathcal{T}\mathcal{A}u - f\|_2^2, \tag{2.4}$$

where $\mathcal{A}$ is the blur operator. Note that this version is not considered in the rest of the paper.

2.2. The algorithm ICTDI. To solve problem (2.3) efficiently, we use the variable splitting technique and quadratic penalty method in the algorithm design. First, we introduce two auxiliary variables and rewrite model (2.3) into the following equivalent formulation with constraints:

$$\begin{aligned} &\min_{d_1,d_2,u} \|d_1\|_p + \tfrac{\lambda}{2}\|\mathcal{P}d_2 - f\|_2^2 \\ &\text{s.t.} \quad \Phi u = d_1, \mathcal{T}u = d_2. \end{aligned} \tag{2.5}$$

Note that this variable splitting method is also used in [12], which has the advantage of separating the two operators P and T such that many existing methods can be used to handle this constrained minimization problem. The authors in [12] use ADM to solve the constrained model, which is also equivalent to ADMM or alternating split Bregman [40]. Many other efficient methods can also be used to solve this problem, including the accelerated primal-dual methods in various forms; we refer to [10, 15, 16, 31, 35] and the references therein for

more details. In this paper, we propose to use the quadratic penalty method to handle the constraints. This method has the advantage that it can be easily changed to another efficient algorithm, which will be addressed in the next subsection. Hence we can approximate model (2.5) by

$$\min_{d_1,d_2,u} E(d_1,d_2,u) = \left\{ \ \|d_1\|_p + \tfrac{\lambda}{2}\|\mathcal{P}d_2 - f\|_2^2 + \tfrac{\mu}{2}\|\Phi u - d_1\|_2^2 + \tfrac{\mu}{2}\|\mathcal{T}u - d_2\|_2^2 \ \right\}, \tag{2.6}$$

where μ is a positive parameter. When the penalty parameter $\mu \to +\infty$, according to the convergence of the quadratic penalty method [38, Theorem 17.1], the solution of (2.6) converges to (2.3). Then we solve the approximated model (2.6) by the alternating minimization method in the following. We set $\tau = 1/\mu$, $\gamma = \lambda/\mu$ to simplify the notation.

2.2.1. Solving d_1, d_2. Fixing u, the subproblems for d_1 and d_2 are separable. For d_1, the subproblem is

$$\min_{d_1} \|d_1\|_p + \frac{\mu}{2}\|\Phi u - d_1\|_2^2. \tag{2.7}$$

It is well known that when $p = 0$ [5], (2.7) is solved by hard shrinkage, i.e.,

$$d_1 = \begin{cases} 0 \text{ if } |\Phi u| < \sqrt{2\tau}, \\ \Phi u \text{ otherwise.} \end{cases} \tag{2.8}$$

When $p = 1$ [26], it is solved by soft shrinkage, that is,

$$d_1 = \max\{|\Phi u| - \tau, 0\}\operatorname{sign}(\Phi u). \tag{2.9}$$

For simplicity, we use the uniform notation as

$$d_1 = \operatorname{shrink}_p(\Phi u, \tau),$$

where $p = 0$ denotes hard shrinkage (2.8) and $p = 1$ denotes soft shrinkage (2.9).

The subproblem for d_2 is

$$\min_{d_2} \gamma\|\mathcal{P}d_2 - f\|_2^2 + \|\mathcal{T}u - d_2\|_2^2. \tag{2.10}$$

The closed-form solution for d_2 is given by

$$d_2 = \left(\gamma\mathcal{P}^T\mathcal{P} + \mathcal{I}\right)^{-1}\left(\gamma\mathcal{P}^T f + \mathcal{T}u\right). \tag{2.11}$$

Since $\mathcal{P}$ is the selection matrix, $\mathcal{P}^T\mathcal{P}$ is a diagonal matrix with diagonal elements 0 or 1, and $\left(\gamma\mathcal{P}^T\mathcal{P} + \mathcal{I}\right)^{-1}$ is a diagonal matrix with diagonal entries 1 or $\frac{1}{\gamma+1}$. Hence the formula of d_2 can be simplified as

$$d_2 = \begin{cases} \mathcal{T}u, & \text{missing coefficents,} \\ \frac{\gamma\mathcal{P}^T f + \mathcal{T}u}{\gamma+1}, & \text{selected coefficients.} \end{cases} \tag{2.12}$$

If we reshape $\mathcal{T}u$ and $\mathcal{P}^T f$ into an $\sqrt{N} \times \sqrt{N}$ matrix, denoted as $\widetilde{\mathcal{P}^T f}$ and $\widetilde{\mathcal{T}u}$, and use $\Lambda \in R^{\sqrt{N}\times\sqrt{N}}$ to denote the selection matrix, then $\widetilde{d_2}$ as an $\sqrt{N}\times\sqrt{N}$ matrix can be formulated as

$$\widetilde{d_2} = \frac{\gamma\Lambda\widetilde{\mathcal{P}^T f} + \widetilde{\mathcal{T}u}}{\gamma\Lambda + 1}, \tag{2.13}$$

where $\Lambda\widetilde{\mathcal{P}^T f}$ is a pointwise multiplication of matrices. Hence the solution of d_2 can be viewed as linear combination of given data $\mathcal{P}^T f$ and estimated data $\mathcal{T}u$. Actually, (2.13) is used in the numerical implementation for its simplicity,

2.2.2. Solving u. Fixing d_1 and d_2, the subproblem for u is

$$\min_u \|\Phi u - d_1\|_2^2 + \|\mathcal{T}u - d_2\|_2^2. \tag{2.14}$$

It is easy to get that the closed-form solution of u is

$$u = (\Phi^T\Phi + \mathcal{I})^{-1}(\Phi^T d_1 + \mathcal{T}^T d_2). \tag{2.15}$$

Finally, the algorithm is summarized in Algorithm 1, ICTDI.

ALGORITHM 1. ICTDI.

- Initialization: $u^0 = \mathcal{T}^T\mathcal{P}^T f, d_1^0 = \mathbf{0}, d_2^0 = \mathbf{0}$.
- For $k = 0, 1, 2, \ldots$, repeat until stopping criterion is reached

$$\begin{aligned} d_1^{k+1} &= \text{shrink}_p(\Phi u^k, \tau); \\ d_2^{k+1} &= \left(\gamma\mathcal{P}^T\mathcal{P} + I\right)^{-1}\left(\gamma\mathcal{P}^T f + \mathcal{T}u^k\right); \\ u^{k+1} &= (\Phi^T\Phi + I)^{-1}(\Phi^T d_1^{k+1} + \mathcal{T}^T d_2^{k+1}). \end{aligned}$$

- Output: u^{k+1}.

2.3. Algorithm IDTDI. Note that when Φ is set as the gradient operator, wavelet, or tight frame, Φ is a fixed matrix. Especially, when Φ is the gradient operator, $\Phi^T\Phi$ can be diagonalized by the fast Fourier transform, which makes the computation of updating u very fast. For wavelet and tight frame, Φ is an analysis matrix and Φ^T equals to the synthesis matrix such that the following equality holds:

$$\Phi^T\Phi = \mathcal{I}.$$

In this case the iteration of u becomes a simple linear combination. However, this equality does not hold for the image adaptive denoising method including dictionary learning [22, 21, 30] and the BM3D filter [18]. In the following, we consider the BM3D filter as an example.

Assume that the analysis and synthesis operators of the BM3D filter are Φ and Ψ, respectively. Let us review the BM3D filter process briefly. The algorithm of the BM3D filter can be split into three steps [18, 19]:

(1) Analysis. Similar image blocks are collected in groups. Then blocks in each group are stacked together to form three-dimensional (3D) data arrays which are decorrelated using an invertible 3D transform.
(2) Processing. The obtained 3D group spectra are filtered by hard/soft thresholding.
(3) Synthesis. The filtered spectra are inverted, providing estimates for each block in the group. These blockwise estimates are returned to their original positions, and the final image reconstruction is calculated as a weighted average of all the obtained blockwise estimates.

In fact, BM3D filter denoising can be expressed by applying the following operator on the given noisy data:

$$\Psi \circ \text{shrink}_p \circ \Phi.$$

Moreover, Danielyan, Katkovnik, and K. Egiazarian [19] give the explicit formulation of Φ, Ψ which is abbreviated here because of space limitations. They prove that Φ, Φ^T, and Ψ have the following relationships:

$$\Phi^T\Phi = \mathcal{W} > 0, \Psi = \mathcal{W}^{-1}\Phi^T, \Psi\Phi = \mathcal{I}. \tag{2.16}$$

Here $\mathcal{W}$ is a diagonal matrix whose entries are defined by the data grouping and counting the number of times each pixel appears in different groups. Experiments show that the range of these entries is very large (up to several hundred times). Hence the matrix $\Phi^T\Phi + \mathcal{I}$ is ill-conditioned, which leads to numerical difficulty. To overcome this drawback, following the methods in [19] and [34], we replace Φ^T by Ψ in Algorithm 1 in the iteration step of u, which yields a new algorithm, IDTDI, detailed in Algorithm 2. Here Φ and Ψ are image adaptive, i.e., $\Phi = \Phi(u^k)$ and $\Psi = \Psi(u^k)$.

ALGORITHM 2. IDTDI.

- Initialization: $u^0 = \mathcal{T}^T\mathcal{P}^T f, d_1^0 = \mathbf{0}, d_2^0 = \mathbf{0}$.
- For $k = 0, 1, 2, \ldots$, repeat until stopping criterion is reached

$$\begin{aligned} d_1^{k+1} &= \text{shrink}_p(\Phi u^k, \tau); \\ d_2^{k+1} &= \left(\gamma\mathcal{P}^T\mathcal{P} + \mathcal{I}\right)^{-1}\left(\gamma\mathcal{P}^T f + \mathcal{T}u^k\right); \\ u^{k+1} &= \frac{1}{2}\left(\Psi d_1^{k+1} + \mathcal{T}^T d_2^{k+1}\right). \end{aligned}$$

- Output: u^{k+1}.

We remark that when $\Psi = \Phi^T$, Algorithm 1 is equivalent to Algorithm 2. However, when $\Psi \neq \Phi^T$, the two algorithms are extremely different. All the updating formulas of Algorithm 1 are derived by minimizing the same energy functional in (2.6), and such that it is called the "coupled" algorithm. In contrast, the iteration formulas in Algorithm 2 come from minimizing the following decoupled functionals, respectively:

$$\begin{cases} (d_1, d_2) = \arg\min\limits_{d_1,d_2} \|d_1\|_1 + \frac{\lambda}{2}\|\mathcal{P}d_2 - f\|_2^2 + \frac{\mu}{2}\|\Phi u - d_1\|_2^2 + \frac{\mu}{2}\|\mathcal{T}u - d_2\|_2^2, \\ u = \arg\min\limits_{u} \|u - \Psi d_1\|_2^2 + \|\mathcal{T}u - d_2\|_2^2. \end{cases} \tag{2.17}$$

That is the reason Algorithm 2 is called "decoupled." Basically, (2.17) is obtained by replacing Φ^T as Ψ in the ICTDI algorithm of (2.6). As here Ψ can better describe the image features, this replacement has the potential to have better image reconstruction in our case.

In fact, the main updating formula in IDTDI can also be rewritten as

$$\begin{cases} \tilde{u}^{k+1} = \Psi \circ \text{shrink}_p \circ \Phi(u^k); \\ d_2^{k+1} = \left(\gamma \mathcal{P}^T \mathcal{P} + I\right)^{-1} \left(\gamma \mathcal{P}^T f + \mathcal{T} u^k\right); \\ u^{k+1} = \dfrac{1}{2}\left(\tilde{u}^{k+1} + \mathcal{T}^T d_2^{k+1}\right). \end{cases} \tag{2.18}$$

In this formulation Algorithm 2 is split into three simple iteration steps: denoising ($\tilde{u}$), linear combination in the transform domain (d_2), and linear combination in the image domain (u). Since the first denoising step is individual, we can use various denoising methods in this step—for example, the state-of-the-art denoising methods such as the overcomplete dictionary learning based denoising method and the BM3D denoising method. As the state-of-the-art denoising method is used in the first step, it is reasonable to expect the state-of-the-art inpainting results. This is the advantage of the decoupling process. We remark that the advantage of decoupling has also been shown in image deblurring and the denoising method [19] and the image domain inpainting method [34]. In [19] the original problem is decoupled into two individual steps: denoising and deblurring. In [34] the inpainting problem is decoupled into denoising and linear combination. Then the BM3D filter is used in the denoising step and state-of-the-art results are achieved.

3. Convergence analysis.

3.1. Convergence of algorithm ICTDI.

Assume that $p = 1$ in this section. We consider the convergence of both algorithms ICTDI and IDTDI. For simplicity, we use $\mathcal{S}_1$ to denote the soft shrinkage operator shrink_1, $\mathcal{S}_2$ to denote the operator $\left(\gamma \mathcal{P}^T \mathcal{P} + I\right)^{-1}$, and M to denote $\Phi^T \Phi + I$. Using these notations, algorithm ICTDI can be simplified as follows:

$$\begin{cases} d_1^{k+1} = \mathcal{S}_1(\Phi u^k), \\ d_2^{k+1} = \mathcal{S}_2\left(\gamma \mathcal{P}^T f + \mathcal{T} u^k\right), \\ u^{k+1} = M^{-1}(\Phi^T d_1^{k+1} + \mathcal{T}^T d_2^{k+1}). \end{cases} \tag{3.1}$$

Furthermore, by introducing the linear operators

$$\begin{aligned} h_1(d_1, d_2) &:= \Phi M^{-1}(\Phi^T d_1 + T^T d_2), \\ h_2(d_1, d_2) &:= \gamma \mathcal{P}^T f + \mathcal{T} M^{-1}(\Phi^T d_1 + T^T d_2), \end{aligned}$$

we can rewrite (3.1) as

$$\begin{cases} d_1^{k+1} = \mathcal{S}_1 \circ h_1(d_1^k, d_2^k), \\ d_2^{k+1} = \mathcal{S}_2 \circ h_2(d_1^k, d_2^k), \\ u^{k+1} = M^{-1}(\Phi^T d_1^{k+1} + \mathcal{T}^T d_2^{k+1}). \end{cases} \tag{3.2}$$

We claim that problem (2.6) has at least one solution under the assumption that the null space of operators Φ and $\mathcal{T}$ satisfies

$$\mathcal{N}(\Phi) \cap \mathcal{N}(T) = \{0\}. \tag{3.3}$$

Under condition (3.3), it is easy to deduce that the function $E(d_1, d_2, u)$ in (2.6) is coercive, that is, $E(d_1, d_2, u) \to \infty$ as $\|(d_1, d_2, u)\| \to \infty$. Moreover, $E(d_1, d_2, u)$ is convex and bounded from below. Therefore, by the standard argument in [4], we get that $E(d_1, d_2, u)$ has at least one minimizer pair (d_1^*, d_2^*, u^*) which satisfies the following Euler–Lagrange equations:

$$\begin{cases} d_1^* = \mathcal{S}_1 \circ h_1(d_1^*, d_2^*), \\ d_2^* = \mathcal{S}_2 \circ h_2(d_1^*, d_2^*), \\ u^* = M^{-1}(\Phi^T d_1^* + \mathcal{T}^T d_2^*). \end{cases} \tag{3.4}$$

We use the notation

$$d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \ H = \begin{pmatrix} \Phi \\ \mathcal{T} \end{pmatrix}$$

and define

$$h(d) := (h_1(d_1, d_2), h_2(d_1, d_2)), \quad \mathcal{S} \circ h(d) := (\mathcal{S}_1 \circ h_1, \mathcal{S}_2 \circ h_2)(d).$$

Then the first two equations of (3.4) imply that $d^* = (d_1^*, d_2^*)$ is a fixed point of $\mathcal{S} \circ h$. Hence the convergence analysis of ICTDI can be established based on the properties of nonexpansiveness of operators $\mathcal{S}$ and h. The nonexpansiveness of soft shrinkage operator $\mathcal{S}_1$ has been proved in [44, 34]. The nonexpansiveness of S_2 is obvious since

$$\|(\gamma P^T P + I)^{-1}\| \le 1.$$

The following proposition states the nonexpansiveness of operator h.

Proposition 3.1. *For any $d, \tilde{d}$ in the range of H, we have*

$$\|h(d) - h(\tilde{d})\| \le \|d - \tilde{d}\|. \tag{3.5}$$

The equality holds if and only if $h(d) - h(\tilde{d}) = d - \tilde{d}$.

Proof. By the definition of h, we have

$$\begin{aligned} \left\|h(d) - h(\tilde{d})\right\| &= \left\| \begin{pmatrix} h_1(d_1, d_2) - h_1(\tilde{d}_1, \tilde{d}_2) \\ h_2(d_1, d_2) - h_2(\tilde{d}_1, \tilde{d}_2) \end{pmatrix} \right\| \\ &= \left\| \begin{pmatrix} \Phi M^{-1}\Phi^T & \Phi M^{-1} T^T \\ T M^{-1}\Phi^T & T M^{-1} T^T \end{pmatrix} \begin{pmatrix} d_1 - \tilde{d}_1 \\ d_2 - \tilde{d}_2 \end{pmatrix} \right\| \\ &= \left\| H M^{-1} H^T \left(d - \tilde{d}\right) \right\| \\ &\le \max |\rho(Q)| \left\| d - \tilde{d} \right\|, \end{aligned} \tag{3.6}$$

where $Q = HM^{-1}H^T$ and $\rho(Q)$ denotes the eigenvalues of Q. We claim that $0 \le \rho(Q) \le 1$. Using the assumption (2.2), we get $H^T H = \Phi^T \Phi + T^T T = \Phi^T \Phi + I = M$. Meanwhile, by [32, Theorem 1.3.22], $Q = HM^{-1}H^T$ have the same nonzero eigenvalues as $M^{-1}H^T H$. Since

$$M^{-1}H^T H = M^{-1} M = I,$$

we conclude that $\rho(Q) \in \{0, 1\}$. Then from (3.6) we can deduce that h is nonexpansive, i.e.,

$$\|h(d) - h(\tilde{d})\| \le \|d - \tilde{d}\|.$$

Assume that $Q = U^T\Sigma U$ is the eigen-decomposition of Q, where U is an orthogonal matrix and Σ is a diagonal matrix with elements in $\{0,1\}$. If the equality holds, we have $\|U^T\Sigma U(d-\tilde{d})\| = \|d-\tilde{d}\|$, and then $\|\Sigma U(d-\tilde{d})\| = \|U(d-\tilde{d})\|$. By the property of Σ, we deduce that $\Sigma U(d-\tilde{d}) = U(d-\tilde{d})$. Multiplying both sides of the above equation by U^T, we get $h(d) - h(\tilde{d}) = d - \tilde{d}$. ■

Based on the nonexpansive property of $\mathcal{S}$ and h, by a similar argument as in Theorem 3.4 [44], we can prove the convergence of ICTDI in the following theorem. The proof is omitted.

Theorem 3.2. *Assume that condition* (3.3) *holds. Then for any fixed parameters* $\tau > 0, \gamma > 0$*, the sequence* $\{(d_1^k, d_2^k, u^k)\}$ *generated by algorithm ICTDI converges to a solution* (d_1^*, d_2^*, u^*) *of problem* (2.6).

3.2. Convergence of algorithm IDTDI. We use the same notation $\mathcal{S}_1$ and $\mathcal{S}_2$ as in subsection 3.1. By introducing the linear operators

$$g_1(d_1,d_2) := \frac{1}{2}\Phi(\Psi d_1 + T^T d_2),$$
$$g_2(d_1,d_2) := \gamma\mathcal{P}^T f + \frac{1}{2}\mathcal{T}(\Psi d_1 + T^T d_2),$$

we can simplify algorithm IDTDI as

$$\text{(3.7)} \qquad \begin{cases} d_1^{k+1} = \mathcal{S}_1 \circ g_1(d_1^k, d_2^k), \\ d_2^{k+1} = \mathcal{S}_2 \circ g_2(d_1^k, d_2^k), \\ u^{k+1} = \dfrac{1}{2}\left(\Psi d_1^{k+1} + \mathcal{T}^T d_2^{k+1}\right). \end{cases}$$

Denote

$$d = \begin{pmatrix} d_1 \\ d_2 \end{pmatrix}, \; G = \begin{pmatrix} \Phi \\ \mathcal{T} \end{pmatrix}, \; K = \begin{pmatrix} \Psi & \mathcal{T}^T \end{pmatrix}$$

and define

$$g(d) := (g_1(d_1,d_2), g_2(d_1,d_2)), \quad \mathcal{S}\circ g(d) := (\mathcal{S}_1 \circ g_1, \mathcal{S}_2 \circ g_2)(d).$$

Then the convergence analysis of IDTDI can be established based on the properties of nonexpansiveness of operators $\mathcal{S}$ and g. The nonexpansiveness of operator g is stated in the following proposition.

Proposition 3.3. *Assume that* GK *is normal. For any* $d, \tilde{d}$ *in the range of* G*, we have*

$$\text{(3.8)} \qquad \|g(d) - g(\tilde{d})\| \le \|d - \tilde{d}\|.$$

The equality holds if and only if $g(d) - g(\tilde{d}) = d - \tilde{d}$.

Proof. Since GK is normal, by the definition of g, we have

$$\text{(3.9)} \qquad \begin{aligned} \left\|g(d) - g(\tilde{d})\right\| &= \left\| \begin{pmatrix} g_1(d_1,d_2) - h_1(\tilde{d}_1,\tilde{d}_2) \\ g_2(d_1,d_2) - h_2(\tilde{d}_1,\tilde{d}_2) \end{pmatrix} \right\| \\ &= \frac{1}{2}\left\| \begin{pmatrix} \Phi\Psi & \Phi T^T \\ T\Psi & TT^T \end{pmatrix} \begin{pmatrix} d_1 - \tilde{d}_1 \\ d_2 - \tilde{d}_2 \end{pmatrix} \right\| \\ &= \frac{1}{2}\left\| GK\left(d - \tilde{d}\right) \right\| \\ &\le \frac{1}{2}\max|\rho(GK)| \left\| d - \tilde{d} \right\|. \end{aligned}$$

By [32, Theorem 1.3.22], GK has the same nonzero eigenvalues as

$$KG = \Psi\Phi + T^T T = 2I.$$

Hence the eigenvalues $\rho(GK) \in \{0, 2\}$. Together with (3.9), we conclude that g is nonexpansive, i.e.,

$$\|g(d) - g(\tilde{d})\| \leq \|g - \tilde{g}\|.$$

Following a similar argument as in Proposition 3.1, we can get that the quality in (3.8) holds if and only if $g(d) - g(\tilde{d}) = d - \tilde{d}$. ∎

Similarly as in Theorem 3.2, we can prove the following result for algorithm IDTDI when Φ is chosen as the BM3D frame.

Theorem 3.4. *Assume that the fixed point set of problem (2.17) is nonempty. Then for any fixed parameters $\tau > 0, \gamma > 0$, the sequence $\{(d_1^k, d_2^k, u^k)\}$ generated by algorithm IDTDI converges to a fixed point (d_1^*, d_2^*, u^*) of problem (2.17).*

3.3. Convergence of algorithms ICTDI/IDTDI for special case. If the operators Φ and Ψ satisfy condition

$$\Phi^T = \Psi, \Phi^T\Phi = \mathcal{I}, \tag{3.10}$$

we get that

$$\|\Phi^T\| = \|\Psi\| = \|\Phi\| = 1.$$

For example, if Φ and Ψ are synthesis and analysis operators of some tight framelet, the above condition is satisfied. Then algorithm ICTDI is equivalent to algorithm IDTDI. Meanwhile, problem (2.6) is equivalent to problem (2.17). In this case, the convergence can be proved in a much easier way in the following. We rewrite the iteration of u as

$$\begin{aligned} u^{k+1} &= \frac{1}{2}\left(\Psi d_1^{k+1} + \mathcal{T}^T d_2^{k+1}\right) \\ &= \frac{1}{2}\left(\Psi\mathcal{S}_1\Phi u^k + \gamma\mathcal{T}^T\mathcal{S}_2\mathcal{P}^T f + \mathcal{T}^T\mathcal{S}_2\mathcal{T}u^k\right) \end{aligned}$$

and define

$$u^{k+1} := \ell(u^k).$$

Then we can prove the following proposition, which states the nonexpansiveness of operator ℓ.

Proposition 3.5. *For any $u, \tilde{u}$, we have*

$$\|\ell(u) - \ell(\tilde{u})\| \leq \|u - \tilde{u}\|. \tag{3.11}$$

The equality holds if and only if $\ell(u) - \ell(\tilde{u}) = u - \tilde{u}$.

Proof. By the definition of ℓ, we can deduce that

$$\begin{aligned} \|\ell(u) - \ell(\tilde{u})\| &= \frac{1}{2}\|(\Psi\mathcal{S}_1\Phi + \mathcal{T}^T\mathcal{S}_2\mathcal{T})(u - \tilde{u})\| \\ &\leq \frac{1}{2}(\|\Psi\left(\mathcal{S}_1(\Phi u) - \mathcal{S}_1(\Phi\tilde{u})\right)\| + \|\mathcal{T}^T\mathcal{S}_2\mathcal{T}\|\|u - \tilde{u}\|) \\ &\leq \frac{1}{2}(\|\Psi\|\|\Phi\|\|u - \tilde{u}\| + \|\mathcal{T}^T\|\|\mathcal{S}_2\|\|\mathcal{T}\|\|u - \tilde{u}\|) \\ &\leq \|u - \tilde{u}\|. \end{aligned} \tag{3.12}$$

In the last equality we used the fact that $\|\mathcal{T}^T\| = \|\mathcal{T}\| = 1$, which is deduced from assumption (2.2). Following a similar argument as in Proposition 1, we can get that the quality in (3.12) holds if and only if $\ell(u) - \ell(\tilde{u}) = u - \tilde{u}$. ■

Then we have the following convergence result.

Theorem 3.6. *Assume that condition* (3.3) *holds. Then for any fixed parameters* $\tau > 0, \gamma > 0$, *the sequence* $\{(d_1^k, d_2^k, u^k)\}$ *generated by algorithm ICTDI or IDTDI converges to a solution* (d_1^*, d_2^*, u^*) *of problem* (2.6).

Particularly, in the case of a tight framelet, we can get a stronger result under the assumption that low frequency coefficients remain unchanged during the shrinkage process. This assumption is very common in image restoration problems since the noise is usually regarded as contained in the high frequency part. Following [7], we split the tight framelet transform matrix Φ as low frequency part $\mathcal{L}$ and high frequency part $\mathcal{H}$, which satisfies

$$\Phi^T\Phi = \mathcal{L}^T\mathcal{L} + \mathcal{H}^T\mathcal{H} = \mathcal{I}. \tag{3.13}$$

Define the new shrinkage operator as

$$\tilde{S}\Phi u = \begin{bmatrix} \mathcal{L}u \\ \mathcal{S}\mathcal{H}u \end{bmatrix} \tag{3.14}$$

in which the low frequency coefficients remain unchanged. Then we have the following convergence theorem.

Theorem 3.7. *If* Φ *is a tight framelet transform, and the smallest eigenvalue* η *of* $\mathcal{L}$ *satisfies* $\eta > 0$, *then the sequence* $\{u^k\}$ *generated by algorithm ICTDI/IDTDI where* $\mathcal{S}$ *is replaced by* $\tilde{S}$ *converges with a convergence factor* $1 - \eta^2/2 > 0$.

Proof. In (3.13) since $\mathcal{L}^T\mathcal{L}$ and $\mathcal{H}^T\mathcal{H}$ are both nonnegative definite, it is easy to get that the smallest eigenvalue η of $\mathcal{L}$ satisfies $\eta < 1$. Then the smallest eigenvalue of $\mathcal{L}^T\mathcal{L}$ is $\eta^2 < 1$. Using the equality (3.13), we derive that the largest eigenvalue of $\mathcal{H}^T\mathcal{H}$ is $1 - \eta^2$. Then $\|\mathcal{H}^T\| = \|\mathcal{H}\| = \sqrt{1-\eta^2}$. Equation (3.12) yields

$$\|\ell(u) - \ell(w)\| \le (1 - \eta^2/2)\|u - w\|.$$

The conclusion thus follows. ■

Note that as shown in [7], if N is properly chosen the assumption $\eta > 0$ can be satisfied. However, in experiments, we find that the image size has no obvious influence on the convergence.

4. Experimental results.

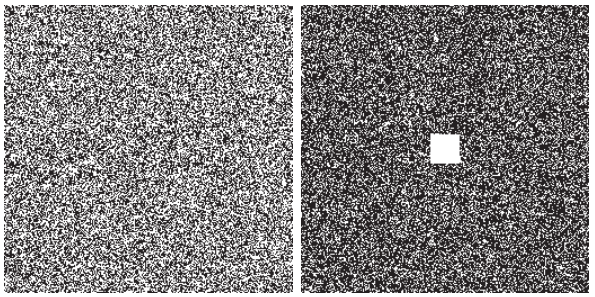
In this section, we apply the proposed algorithms on several standard test images in which some coefficients are missing in the wavelet transform domain or the Fourier transform domain. The results are compared with some closely related methods in the literature.

4.1. Experiments setting.

We choose three typical test images in our experiments, "Slope," "Cameraman," and "Barbara," displayed in Figure 1. The Slope image is piecewise smooth with sharp edges. The Cameraman image has both a large area of cartoon and some fine strutures. The Barbara image has many texture patterns. Three subregions inside the red rectangle will be zoomed to compare in detail.

(a) Slope (b) Cameraman (c) Barbara

Figure 1. *Test images.*



(a) sparsity = 50% (b) sparsity = 30.72%

Figure 2. *Inpainting masks in transform domain with missing data marked in black.* (a) *Wavelet domain mask;* (b) *Fourier domain mask.*

In Figure 2, we display two selection/downsampling masks for the transform domain. Figure 2(a) is the mask for wavelet domain inpainting in which 50% randomly chosen coefficients are known, and Figure 2(b) is the mask for Fourier domain inpainting in which 30.72% coefficients are known, including a rectangle in the low frequency area and some randomly chosen high frequency data. Note that for Fourier domain inpainting, we include some low frequency coefficients around the center in order to get better image visual quality. See [14] for other kinds of masks in the Fourier domain in which also a small area of low frequency data are included. Model (2.1) is used to simulate the transform domain inpainting data. For wavelet domain inpainting, $\mathcal{T}$ is set as the wavelet transform with daubcqf(6) basis and two levels of decomposition using Rice wavelet toolbox 2.4. Gaussian noise with zero mean and standard deviation 1 is added.

We mainly test three operators in the proposed algorithms: (i) ICTDI-∇, where $\Phi = \nabla$ is the gradient operator; (ii) ICTDI/IDTDI-W, where $\Phi = W$ is a tight framelet transform constructed as in [7]; and (iii) IDTDI-BM3D, where Φ and Ψ are BM3D analysis and synthesis operators [18]. Additionally, we test the ICTDI-BM3D algorithm for problem (2.6) and the ADMM algorithm for problem (2.5), abbreviated as ADMM-BM3D.

We compare our results with the following existing methods: back projection (BP) using the formula $\mathcal{T}^T\mathcal{P}^T f$; cubic interpolation in the wavelet domain based on Delaunay triangulation (implemented by the MATLAB routine griddata); the ADM wavelet domain inpainting

method [12] (we also apply it to Fourier domain inpainting for comparison); nonlocal TV regularization; and a Bregmanized operator splitting based Fourier domain inpainting method [50]. We remark that although there exist many numerical methods related to the TV based transform domain inpainting method, their results have a similar quality since they solve the same model [13, 11, 36, 12, 45]. Here we choose ADM in [12] as a representative. All the key parameters (regularization parameters) are tuned carefully in each method and the optimal result is chosen to compare.

The default parameters of the proposed methods are set as follows: for ICTDI-∇, $\gamma = 10$, $\tau = 5$ for wavelet domain inpainting and $\tau = 1$ for Fourier domain inpainting; for ICTDI/IDTDI-W, $\gamma = 10$, $\tau = 3$ for wavelet domain inpainting and $\tau = 0.5$ for Fourier domain inpainting. For ICTDI-BM3D and IDTDI-BM3D, we set $\gamma = 1000$, patch size $= 8$. To speed up the IDTDI-BM3D algorithm, first, we choose the result of BP after cubic interpolation in the low frequency in the wavelet domain as initialization of u^0. Second, since larger τ corresponds to faster diffusion speed and smaller τ corresponds to higher inpainting quality (which will be addressed in section 4.4), we choose to decrease the noise estimate τ during iteration. For wavelet domain inpainting, we set $\tau = [20, 15, 10, 5, 2, 1]$. Sixty iterations are performed for each $\tau > 1$ and $\tau = 1$ for the rest iterations. Meanwhile, for Fourier domain inpainting, we set $\tau = [10, 8, 5, 2, 1, 0.5]$. Sixty iterations are performed for each $\tau > 0.5$ and $\tau = 0.5$ for the rest iterations.

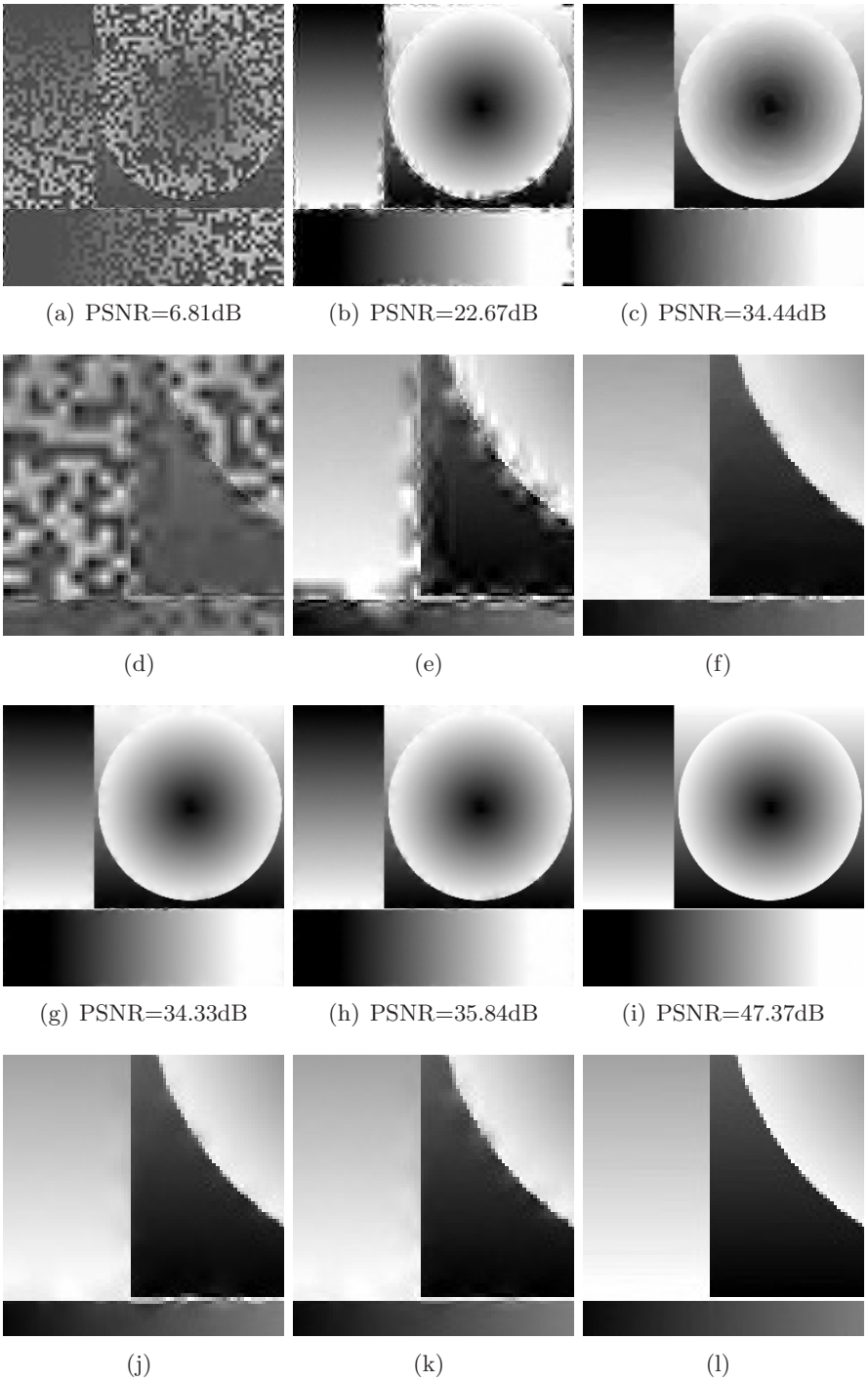
Following [12], for iterative methods ADM, ICTDI-∇, and ICTDI/IDTDI-W, we set that the stopping criterion as the relative error (ReErr) between the successive iterate of the restored image should satisfy the following inequality:

$$\text{ReErr} = \frac{\|u^{k+1} - u^k\|_2}{\|u^{k+1}\|_2} < 10^{-5}.$$

Note that this stopping criterion is widely used in image restoration problems [33, 37, 52]. For NLTV and IDTDI-BM3D, we set maximum iteration but we do not use ReErr. The first reason is the ReErr curves are oscillating for these two methods. The second reason is these patch based algorithms are slow compared with other methods, including ADM, ICTDI-∇, and ICTDI/IDTDI-W. By setting maximum iteration we can achieve a good balance of image reconstruction quality and computational time. The maximum iterations of NLTV and IDTDI-BM3D are set as 200 and 500, respectively, which can almost ensure the numerical convergence in our experiments.

All the experiments are performed under Windows 8 and MATLAB R2012a with Intel Core i7-4500 CPU@1.80GHz and 8GB memory. The programming language is mixed MATLAB and C for ICTDI-BM3D, IDTDI-BM3D, ADMM-BM3D, and NLTV [50], while it is MATLAB for all the other algorithms.

4.2. Wavelet domain inpainting. In Figures 3–5 we test the three images Slope, Cameraman, and Barbara in Figure 1 with 50% wavelet coefficient missing selected by mask in Figure 2(a). We display the wavelet inpainting results by eight methods: BP, cubic interpolation, ADM [12], ICTDI-∇, ICTDI/IDTDI-W, and IDTDI-BM3D. Note that in Figures 3–5, the second and fourth rows show the zoomed regions (marked in Figure 1 by red rectangles) in the first and third rows correspondingly.



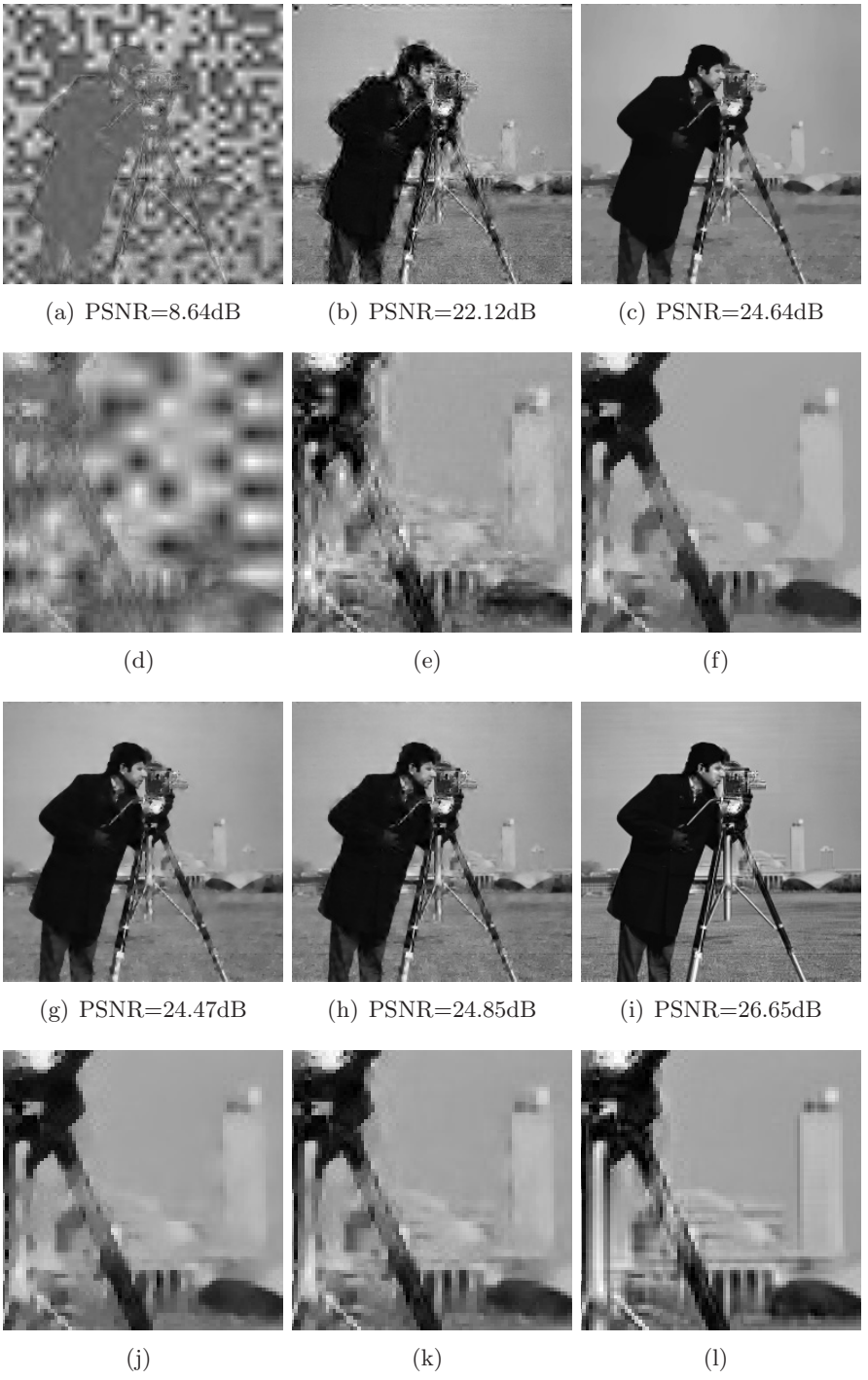
(a) PSNR=6.81dB (b) PSNR=22.67dB (c) PSNR=34.44dB

(d) (e) (f)

(g) PSNR=34.33dB (h) PSNR=35.84dB (i) PSNR=47.37dB

(j) (k) (l)

Figure 3. *Wavelet domain inpainting for Slope with mask in Figure* 2(a)*, sparsity* = 50%. (a) *BP;* (b) *cubic interpolation: time* = 0.02*s;* (c) *ADM* [12]*: iteration* = 485*, time* = 8.75*s;* (d)–(f) *the zoomed regions of* (a)–(c)*, respectively;* (g) *ICTDI-*∇*: iteration* = 414*, time* = 5.64*s;* (h) *ICTDI/IDTDI-W: iteration* = 1230*, time* = 58.78*s;* (i) *IDTDI-BM3D: iteration* = 500*, time* = 410.88*s;* (j)–(l) *the zoomed regions of* (g)–(i)*, respectively.*



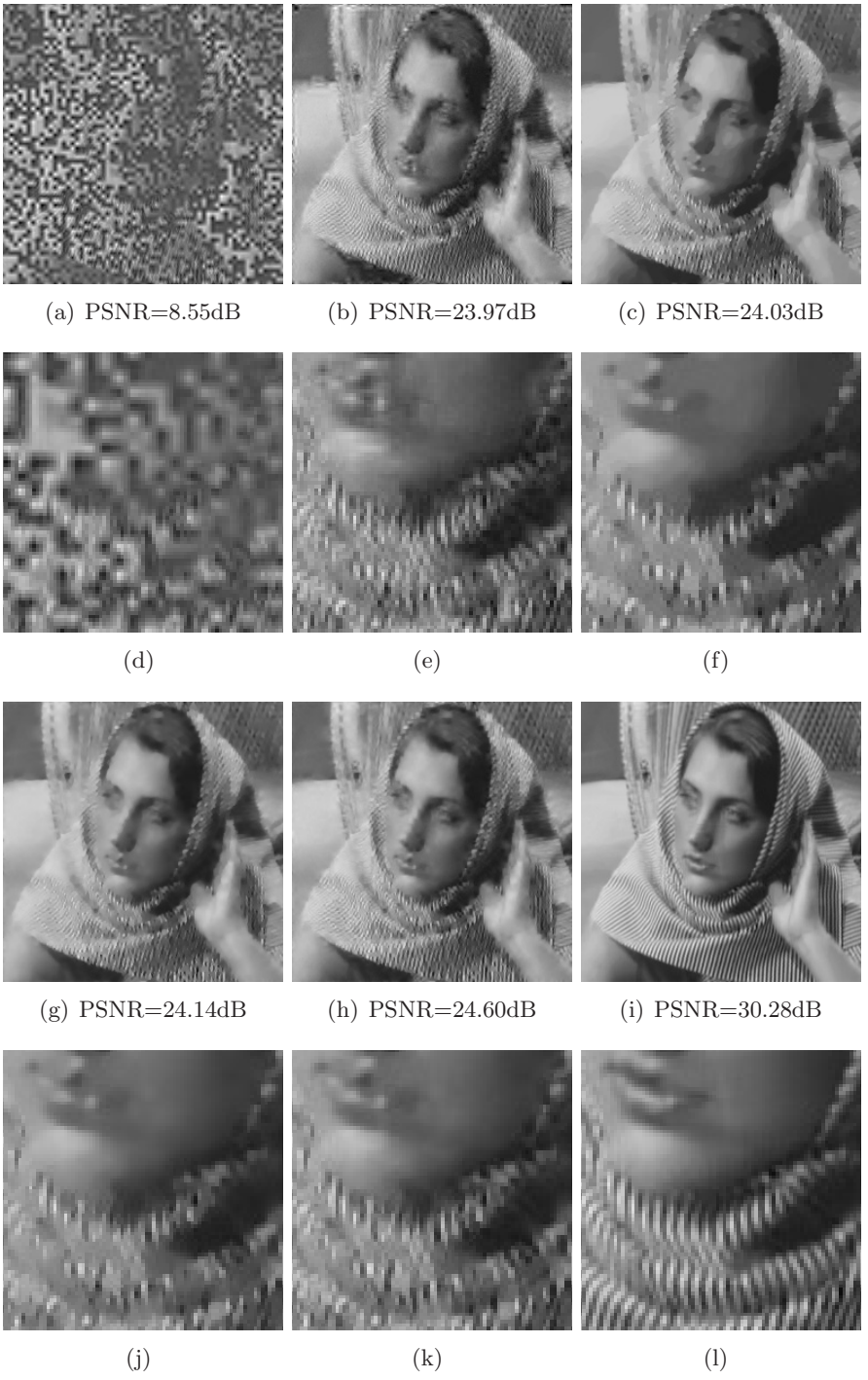
(a) PSNR=8.64dB (b) PSNR=22.12dB (c) PSNR=24.64dB

(d) (e) (f)

(g) PSNR=24.47dB (h) PSNR=24.85dB (i) PSNR=26.65dB

(j) (k) (l)

Figure 4. *Wavelet domain inpainting for Cameraman with mask in Figure* 2(a)*, sparsity* = 50%. (a) *BP;* (b) *cubic interpolation: time* = 0.02*s;* (c) *ADM* [12]: *iteration* = 596, *time* = 10.84*s;* (d)–(f) *the zoomed regions of* (a)–(c)*, respectively;* (g) *ICTDI-*∇*: iteration* = 518, *time* = 7.30*s;* (h) *ICTDI/IDTDI-W: iteration* = 1500, *time* = 73.17*s;* (i) *IDTDI-BM3D: iteration* = 500, *time* = 378.98*s;* (j)–(l) *the zoomed regions of* (g)–(i)*, respectively.*



(a) PSNR=8.55dB (b) PSNR=23.97dB (c) PSNR=24.03dB

(d) (e) (f)

(g) PSNR=24.14dB (h) PSNR=24.60dB (i) PSNR=30.28dB

(j) (k) (l)

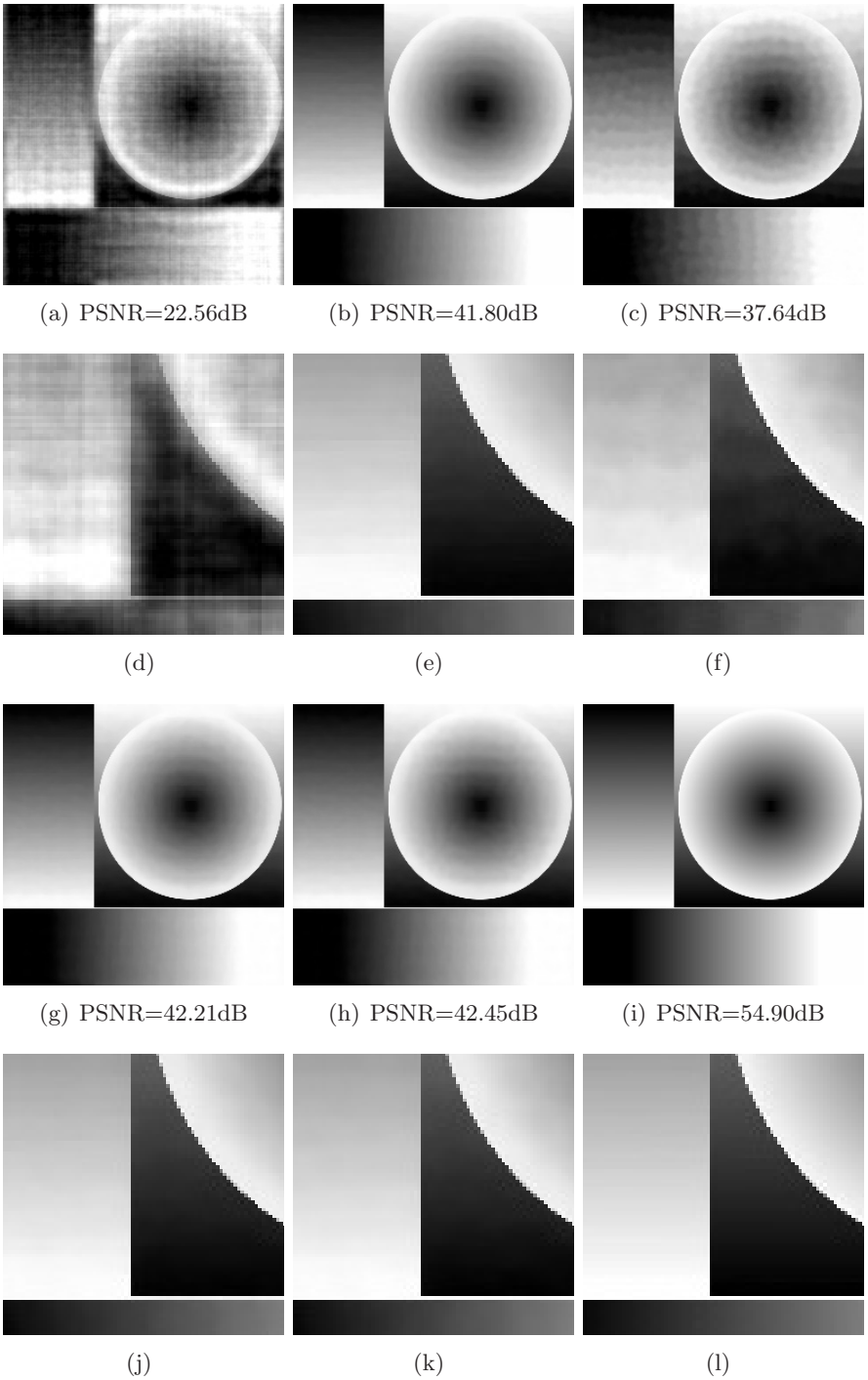
Figure 5. *Wavelet domain inpainting for Barbara with mask in Figure* 2(a)*, sparsity* = 50%*.* (a) *BP;* (b) *cubic interpolation: time* = 0.02*s;* (c) *ADM* [12]*: iteration* = 594*, time* = 10.70*s;* (d)–(f) *the zoomed regions of* (a)–(c)*, respectively;* (g) *ICTDI-*∇*: iteration* = 564*, time* = 8.16*s;* (h) *ICTDI/IDTDI-W: iteration* = 1608*, time* = 79.02*s;* (i) *IDTDI-BM3D: iteration* = 500*, time* = 391.86*s;* (j)–(l) *the zoomed regions of* (g)–(i)*, respectively.*

From Figures 3(a)–5(a), the results of BP, we observe that much information is lost in the pixel domain. In the cubic interpolation method displayed in Figures 3(b)–5(b), we only interpolate the low frequency in the wavelet domain, and then apply BP to reconstruct the image. The results seem much better than direct BP. Low frequency information is almost recovered. However, the high frequency information along the edges in Figure 3(b), the fine strutures in Figure 4(b), and the texture patterns in Figure 5(b) are poorly recovered. Since high frequency and low frequency components are very different, we remark that using cubic interpolation on the high frequency component is not helpful for recovering high frequency information. ADM and the proposed ICTDI-∇ are both solving the TV wavelet domain inpainting model, and so similar results with similar peak signal-to-noise ratio (PSNR) values are obtained; see Figures 3(c)–5(c) and Figures 3(g)–5(g). The high frequency parts are much better recovered by TV based methods than by cubic interpolation and also higher PSNR values are reported. However, the drawbacks are also apparent, especially through the zoomed rectangle regions. For example, the horizontal edge in Slope is not recovered, and the fine structures of Cameraman and the textures of Barbara are oversmoothed. As displayed in Figures 3(h)–5(h), the proposed ICTDI/IDTDI-W algorithm has slightly better restoration quality than TV based methods ADM and ICTDI-∇ with about 0.5 dB higher PSNR values on average. Observing the zoomed regions, we find that the horizontal edge in Slope is fully recovered and more textures in Barbara are recovered by ICTDI/IDTDI-W. The results of IDTDI-BM3D are displayed in Figures 3(i)–5(i) and seem quite good in the above mentioned clues. The edges, fine strutures, and textures are far better recovered than in other methods. In terms of PSNR, IDTDI-BM3D achieves about 11.53dB in Slope, 1.8dB in Cameraman, and 5.68dB in Barbara higher than the second best algorithm IDTDI-W. We remark that BM3D is a patch based method. So if more similar patches can be found in the image, the restoration quality will be better. That is why it enhances so much PSNR in Slope and Barbara.

4.3. Fourier domain inpainting. In Figures 6–8, we test three images Slope, Cameraman, and Barbara with 30.72% known Fourier coefficients selected by mask in Figure 2(b). We display the Fourier inpainting results by six methods: BP, ADM, NLTV and the proposed algorithms ICTDI-∇, ICTDI/IDTDI-W, and IDTDI-BM3D. Again, the second and fourth rows in Figures 6–8 show the zoomed regions of the results in the first and third rows correspondingly.

As shown in Figures 6–8, the results of direct BP have the poorest visual quality and lowest PSNR values. Among all, the proposed IDTDI-BM3D algorithm achieves the best visual quality and the highest PSNR values. In terms of PSNR, IDTDI-BM3D achieves about 12.45dB in Slope, 0.76dB in Cameraman, and 1.99dB in Barbara higher than the second best algorithm. Note that for Cameraman, ADM and ICTDI-∇ also have similar PSNR values and visual quality. ICTDI/IDTDI-W is slightly better than ADM and ICTDI-∇, in which the PSNR values are about 0.3dB higher on average. However, the artifacts of ADM, ICTDI-∇, and ICTDI/IDTDI-W are apparent in both Slope and Barbara, for instance, the "staircase" effect in Slope, oversmoothness of textures in Barbara, and texture-like artifacts on Barbaras face. For Barbara, NLTV performs much better than ADM, ICTDI-∇, and ICTDI/IDTDI-W; see the textures on Barbara, for example.

To further illustrate the effectiveness of IDTDI-BM3D, in Figure 9, we compare it with ICTDI-BM3D for solving problem (2.6) and ADMM-BM3D for solving problem (2.5). In all



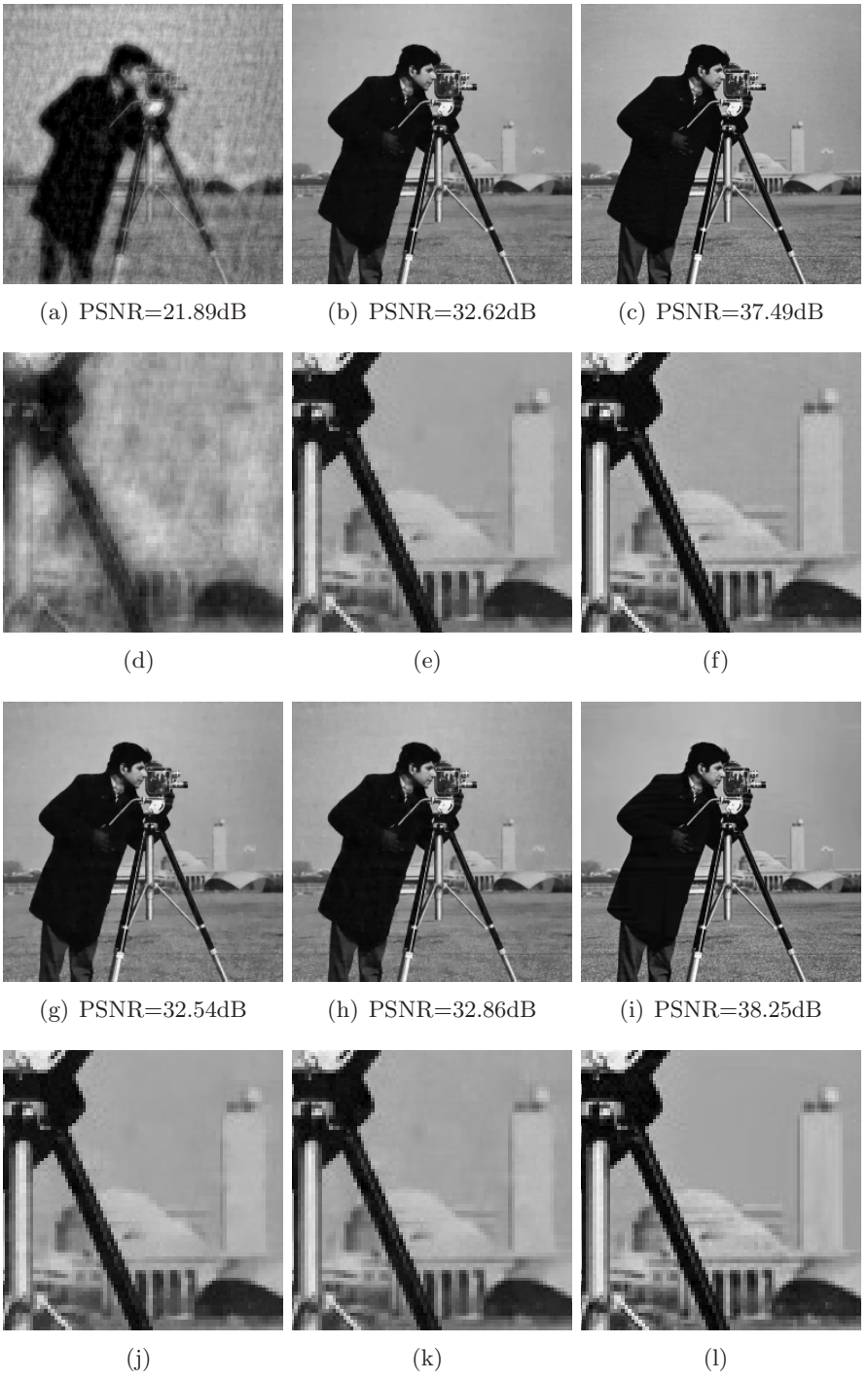
(a) PSNR=22.56dB (b) PSNR=41.80dB (c) PSNR=37.64dB

(d) (e) (f)

(g) PSNR=42.21dB (h) PSNR=42.45dB (i) PSNR=54.90dB

(j) (k) (l)

Figure 6. *Fourier domain inpainting for Slope with mask in Figure* 2(b)*, sparsity* = 30.72%. (a) *BP;* (b) *ADM* [12]: *iteration* = 280, *time* = 6.52*s;* (c) *NLTV* [50]: *iteration* = 200, *time* = 452.83*s;* (d)–(f) *the zoomed regions of* (a)–(c)*, respectively;* (g) *ICTDI-*∇*: iteration* = 250, *time* = 4.05*s;* (h) *ICTDI/IDTDI-W: iteration* = 591, *time* = 42.08*s;* (i) *IDTDI-BM3D: iteration* = 500, *time* = 426.78*s;* (j)–(l) *the zoomed regions of* (g)–(i)*, respectively.*



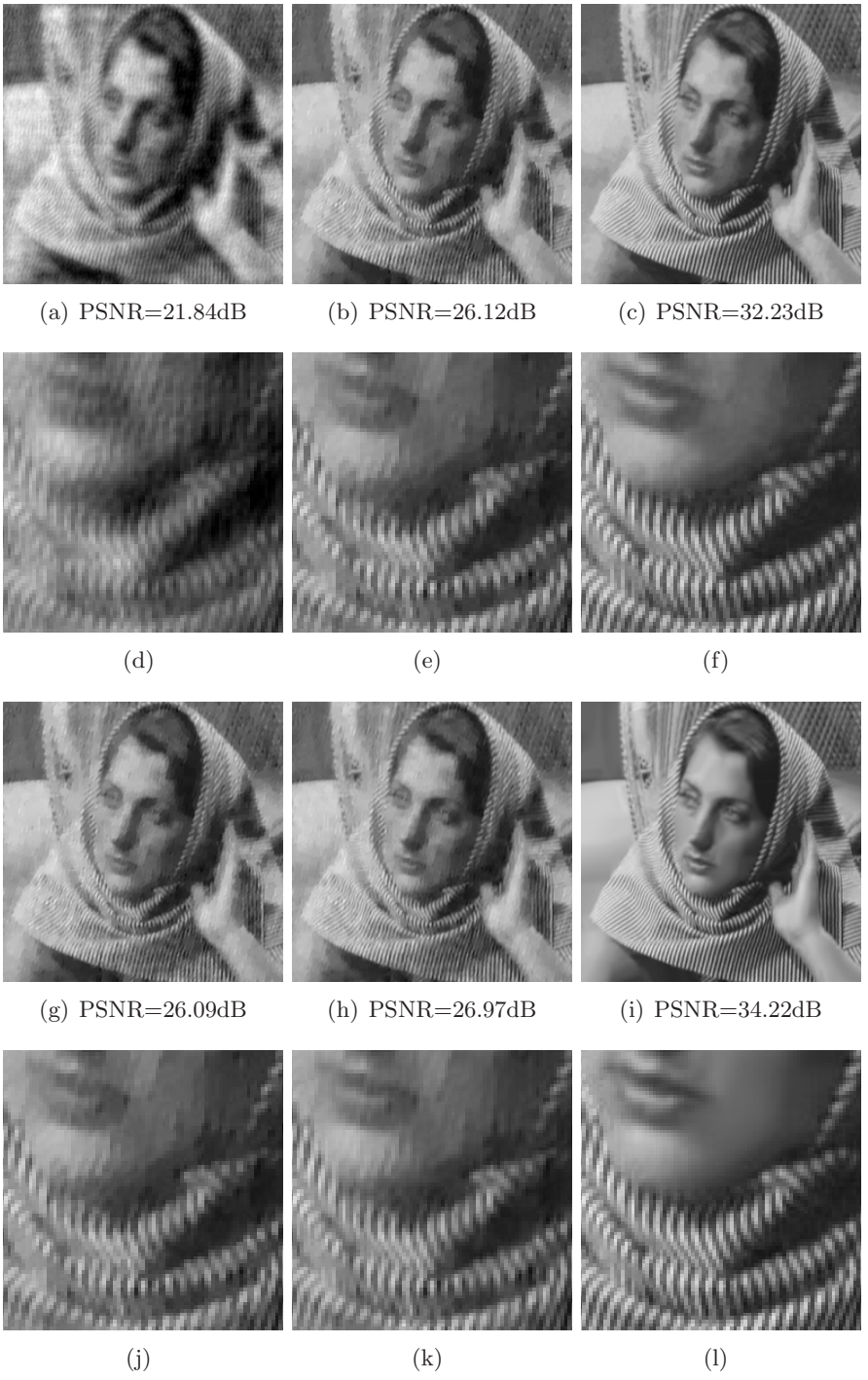
(a) PSNR=21.89dB (b) PSNR=32.62dB (c) PSNR=37.49dB

(d) (e) (f)

(g) PSNR=32.54dB (h) PSNR=32.86dB (i) PSNR=38.25dB

(j) (k) (l)

Figure 7. *Fourier domain inpainting for Cameraman with mask in Figure* 2(b)*, sparsity* = 30.72%. (a) *BP;* (b) *ADM* [12]: *iteration* = 356, *time* = 8.03*s;* (c) *NLTV* [50]: *iteration* = 200, *time* = 446.38*s;* (d)–(f) *the zoomed regions of* (a)–(c)*, respectively;* (g) *ICTDI-*∇*: iteration* = 402, *time* = 6.81*s;* (h) *ICTDI/IDTDI-W: iteration* = 768, *time* = 54.59*s;* (i) *IDTDI-BM3D: iteration* = 500, *time* = 399.75*s;* (j)–(l) *the zoomed regions of* (g)–(i)*, respectively.*



(a) PSNR=21.84dB (b) PSNR=26.12dB (c) PSNR=32.23dB

(d) (e) (f)

(g) PSNR=26.09dB (h) PSNR=26.97dB (i) PSNR=34.22dB

(j) (k) (l)

Figure 8. *Fourier domain inpainting for Barbara with mask in Figure* 2(b) *sparsity* $= 30.72\%$. (a) *BP;* (b) *ADM* [12]: *iteration* $= 488$, *time* $= 11.27s$; (c) *NLTV* [50]: *iteration* $= 200$, *time* $= 458.64s$; (d)–(f) *the zoomed regions of* (a)–(c), *respectively;* (g) *ICTDI-*∇*: iteration* $= 466$, *time* $= 7.98s$; (h) *ICTDI/IDTDI-W: iteration* $= 732$, *time* $= 53.58s$; (i) *IDTDI-BM3D: iteration* $= 500$, *time* $= 415.64s$; (j)–(l) *the zoomed regions of* (g)–(i), *respectively.*

(a) PSNR=32.69dB (b) PSNR=32.28dB (c) PSNR=34.26dB

(d) (e) (f)

Figure 9. *Fouier domain inpainting for Barbara with mask in Figure* 2(b)*, sparsity* = 30.72%*, iteration* = 3000*.* (a) *ICTDI-BM3D;* (b) *ADMM-BM3D;* (c) *IDTDI-BM3D;* (d)–(f) *the zoomed regions of* (a)–(c)*, respectively.*

three methods, the BM3D frame is used as a regularization operator. We iterate 3000 times for each method to ensure convergence. We observe that IDTDI-BM3D achieves higher PSNR than ICTDI-BM3D and ADMM-BM3D. In addition, IDTDI-BM3D takes less computational time at each iteration than the other two (about 1/2 of ICTDI-BM3D and 1/8 of ADMM-BM3D). In both algorithms ICTDI-BM3D and ADMM-BM3D, the operator $(\Phi^T\Phi + I)^{-1}$ is unavoidable. As is analyzed in subsection 2.3, $\Phi^T\Phi + I$ is ill-conditioned, which increases the computational complexity and numerical instability of both algorithms. In contrast, IDTDI-BM3D avoids this problem and the numerical implementation is more efficient and stable. Moreover, as is shown in (2.18), which is the equivalent form of algorithm IDTDI-BM3D, the BM3D filter is one step of the algorithm. Hence we can make use of the high efficient source code of the BM3D filter provided by the authors to speed up the whole algorithm. Therefore, we prefer IDTDI-BM3D in this paper. We remark that one advantage of ADMM over the quadratic penalty method is that the former avoids having to let the penalty parameter go to infinity. On the other hand, ADMM depends heavily on the right choice of the step size parameter and some step size strategies have been proposed which can speed up the computation dramatically; see [25] and references therein. Moreover, there exist many algorithms such as PDHG [10], related to ADMM, which avoid the matrix inversions. These speeding-up methods are not considered here.

To illustrate the convergence behavior of IDTDI-BM3D and the other six compared methods ADM, NLTV, ICTDI-∇, ICTDI/IDTDI-W, ICTDI-BM3D, and ADMM-BM3D, we plot

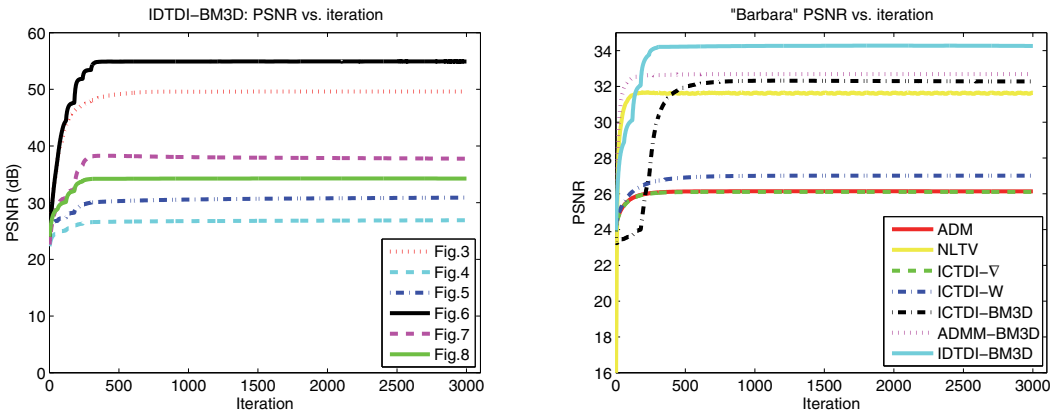
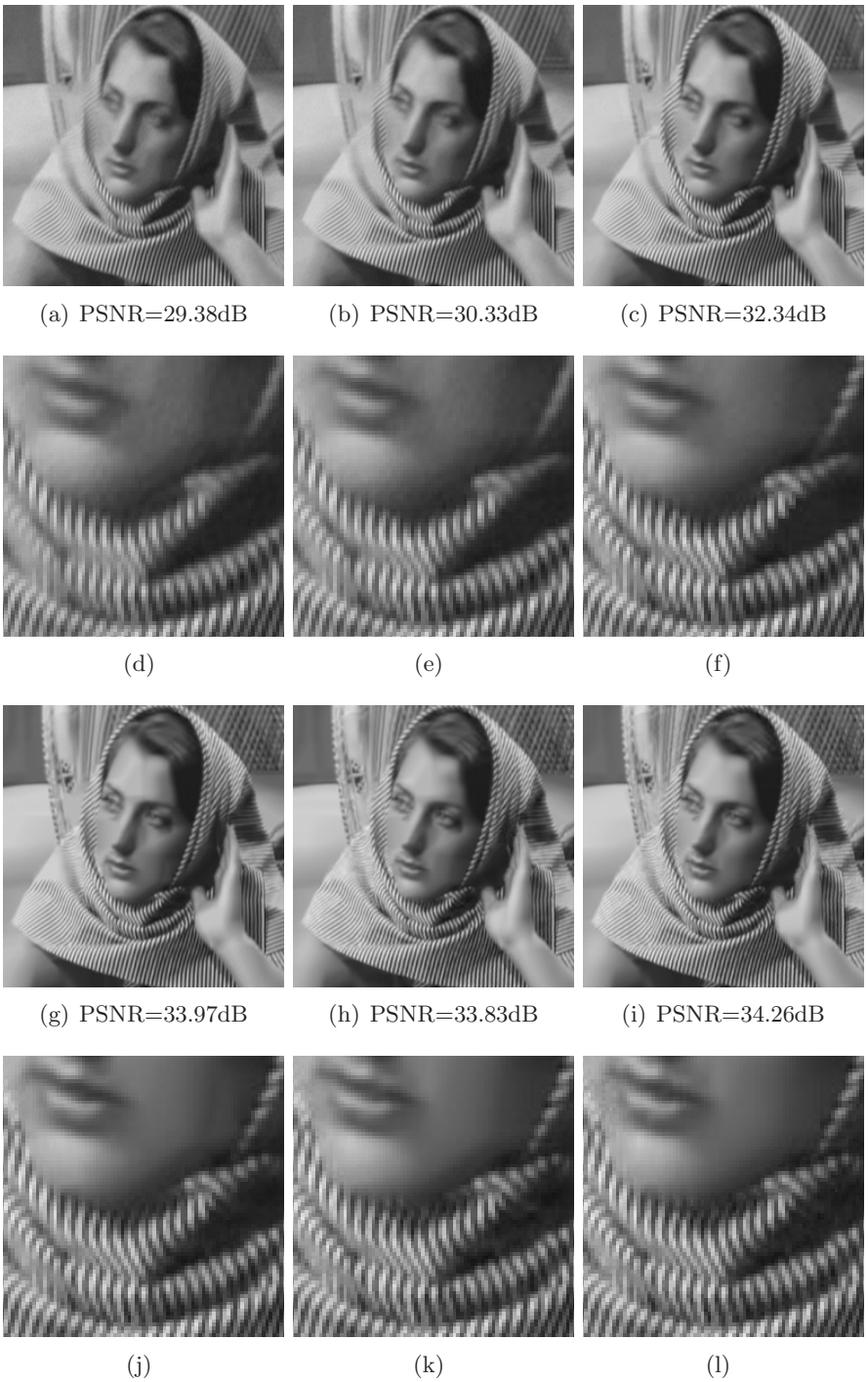


Figure 10. *Convergence behavior.* (a) *PSNR versus iteration of IDTDI-BM3D algorithm corresponding to the test in Figures 3–5 and Figures 6–8;* (b) *PSNR versus iteration of seven iterative methods for Fourier domain inpainting including ADM, NLTV, ICTDI-∇, ICTDI/IDTDI-W, ICTDI-BM3D, ADMM-BM3D, and IDTDI-BM3D.*

the curves of PSNR as functions of iteration number in Figure 10. For all the methods, the iteration is fixed as 3000, which is large enough to ensure the numerical convergence. In Figure 10(a), we display the PSNR curves of IDTDI-BM3D corresponding to the test images in Figures 3–5 and Figures 6–8. We observe that all the curves are increasing very fast at the beginning. The step of PSNR curves at the beginning iterations is caused by varying parameter τ. That is, when τ becomes smaller, there will be a sharp increment of PSNR. In Figure 10(b), the test image is Barbara as in Figure 8. Figure 10(b) shows that for all methods, the PSNR curves are almost increasing. The three BM3D involved algorithms (ICTDI-BM3D, ADMM-BM3D, and IDTDI-BM3D) and NLTV achieve much higher PSNR values than the other three (ADM, ICTDI-∇, and ICTDI/IDTDI-W). Among all, IDTDI-BM3D achieves the highest PSNR value.

4.4. Computational time and effect of parameters. In Figures 3–5 and Figures 6–8, we report the computational time of each method. For wavelet domain inpainting, cubic interpolation is not an iterative method and is very fast. About 0.02 second is taken for cubic interpolation for a 256×256 image. TV inpainting model based algorithms ADM and ICTDI-∇ are the fastest in the iteration type methods which take about 10 seconds for 500 iterations. Actually, ICTDI-∇ is slightly faster than ADM, since in each iteration ADM needs to update two extra Lagrangian multiplier variables. On average, ICTDI/IDTDI-W takes about 1400 iterations with 70 seconds in wavelet domain inpainting and it takes around 700 iterations with 50 seconds in Fourier domain inpainting for a 256×256 image. NLTV and IDTDI-BM3D are more time-consuming than others. On average, NLTV takes about 450 seconds for 200 iterations and IDTDI-BM3D takes about 410 seconds for 500 iterations. Note that the CPU time of our algorithms can still be reduced by including some parallel computing techniques or stop before convergence once the satisfactory result has been obtained.



(a) PSNR=29.38dB (b) PSNR=30.33dB (c) PSNR=32.34dB

(d) (e) (f)

(g) PSNR=33.97dB (h) PSNR=33.83dB (i) PSNR=34.26dB

(j) (k) (l)

Figure 11. *The effect of parameter τ in IDTDI-BM3D (Fourier domain inpainting for Barbara with mask in Figure 2(b), sparsity = 30.72%), iteration = 3000.* (a) $\tau = 10$; (b) $\tau = 8$; (c) $\tau = 5$; (d)–(f) *the zoomed regions of* (a)–(c), *respectively;* (g) $\tau = 2$; (h) $\tau = 1$; (i) *varying* τ; (j)–(l) *the zoomed regions of* (g)–(i), *respectively.*

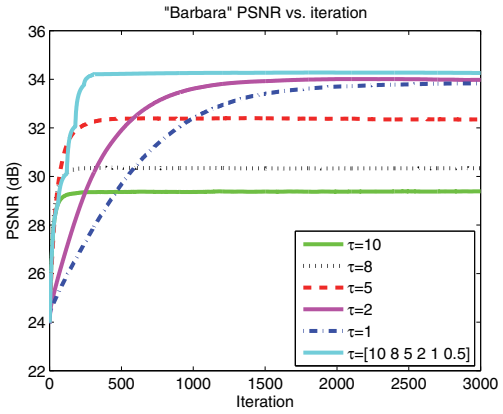


Figure 12. *PSNR versus iteration for different choice of parameter τ by applying IDTDI-BM3D on Barbara Fourier domain inpainting.*

In our algorithms, there are two parameters: τ and γ. The proposed algorithms are not sensitive to γ. Indeed, τ is the key parameter to control the restoration quality and convergence speed. For instance, we test IDTDI-BM3D for Fourier domain inpainting problem on Barbara with different choices of τ. We compare the fixed parameter setting $\tau = 10, 8, 5, 2, 1$ and varying parameter setting, respectively. The resulting images and PSNR curves are shown in Figures 11 and 12. We observe in Figure 11 that τ controls the smoothness of the restored image. When τ is bigger, the result is smoother. Figures 11(a)–11(b) show that bigger τ causes oversmoothing of textures. The results in Figures 11(c), 11(g)–11(i) are similar in visual aspect. However, with varying parameter, the proposed algorithm IDTDI achieves a higher PSNR value than others. In Figure 12, we display the PSNR versus iteration curves for different τ. It is clear that all the curves are almost increasing. We observe that there is a trade-off between the image inpainting quality and the convergence speed. In general, smaller τ leads to higher inpainting quality but slower inpainting speed. Meanwhile, we observe in the experiments that there is an upper limit of inpainting quality when μ is fixed. As an example, Figure 12 shows that the optimal performance is achieved for $\tau = 2$ and $\tau = 1$ when τ is fixed. It implies that $\tau = 2$ is small enough to gain the best inpainting quality, so it is not necessary to choose τ smaller than 2 (which will cost more computational time than $\tau = 2$). This optimal τ can be chosen by trail and error. Instead of choosing an optimal τ, another parameter choosing scheme is decreasing τ gradually. This varying τ setting works like this: at the beginning stage, larger τ leads to oversmoothing of the image with a fast speed; then by decreasing τ gradually, more and more details will be recovered; finally the high quality result will be obtained. On the whole, we can safely conclude that the setting of decreasing parameter τ gradually in the algorithm IDTDI-BM3D is helpful for enhancing convergence speed and at the same time enhancing image restoration quality. We remark that similar behavior of varying the parameter in the penalty method has been justified theoretically in [29].

5. Conclusion. In this paper, we extend an existing variational model for image inpainting in the transform domain and propose two general novel algorithms ICTDI and IDTDI. In the proposed framework, many existing regularization operators can be used, such as gradient, tight frame, and the adaptive BM3D frame. Experiments and comparisons show that the proposed IDTDI-BM3D can produce very promising results in transform domain inpainting problems. The key part of IDTDI is just the replacement of Φ^T by Ψ in ICTDI. This replacement can be considered when the subproblems involving $\Phi^T\Phi$ are not easy to solve. The effectiveness of this replacement needs to be justified in other applications in image processing. In future work, we will generalize our framework to other image processing problems, for instance, the image deblurring and inpainting problem (2.4) and the image segmentation problem. With the idea of decoupling, we can utilize the most efficient regularization schemes such that it is very possible to enhance the recovered image quality and segmentation accuracy.

Acknowledgments. The authors would like to thank X. Zhang for providing thes source code of [50] for comparison. We would like to thank all the anonymous referees for their useful comments and suggestions.

REFERENCES

[1] 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.
[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, ACM Press/Addison-Wesley, Reading, MA, 2000, pp. 417–424.
[3] A. L. Bertozzi, S. Esedoglu, and A. Gillette, *Inpainting of binary images using the Cahn–Hilliard equation*, IEEE Trans. Image Process., 16 (2007), pp. 285–291.
[4] D. P. Bertsekas, A. Nedi, A. E. Ozdaglar, et al., *Convex Analysis and Optimization*, Athena Scientific, Belmont, MA, 2003.
[5] K. Bredies and D. A. Lorenz, *Iterated hard shrinkage for minimization problems with sparsity constraints*, SIAM J. Sci. Comput., 30 (2008), pp. 657–683.
[6] M. Burger, A. Sawatzky, and G. Steidl, *First Order Algorithms in Variational Image Processing*, preprint, arXiv:1412.4237, 2014.
[7] J.-F. Cai, R. H. Chan, L. Shen, and Z. Shen, *Convergence analysis of tight framelet approach for missing data recovery*, Adv. Comput. Math., 31 (2009), pp. 87–113.
[8] J.-F. Cai, R. H. Chan, and Z. Shen, *A framelet-based image inpainting algorithm*, Appl. Comput. Harmon. Anal., 24 (2008), pp. 131–149.
[9] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.
[10] 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.
[11] R. H. 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.
[12] R. H. Chan, J. Yang, and X. Yuan, *Alternating direction method for image inpainting in wavelet domains*, SIAM J. Imaging Sci., 4 (2011), pp. 807–826.
[13] T. F. Chan, J. Shen, and H.-M. Zhou, *Total variation wavelet inpainting*, J. Math. Imaging Vision, 25 (2006), pp. 107–125.
[14] Y. Chen, W. Hager, F. Huang, D. Phan, X. Ye, and W. Yin, *Fast algorithms for image reconstruction with application to partially parallel MR imaging*, SIAM J. Imaging Sci., 5 (2012), pp. 90–118.
[15] Y. Chen, G. Lan, and Y. Ouyang, *Optimal primal-dual methods for a class of saddle point problems*, SIAM J. Optim., 24 (2014), pp. 1779–1814.

[16] E. Corman and X. Yuan, *A generalized proximal point algorithm and its convergence rate*, SIAM J. Optim., 24 (2014), pp. 1614–1638.
[17] A. Criminisi, P. Pérez, and K. Toyama, *Region filling and object removal by exemplar-based image inpainting*, IEEE Trans. Image Process., 13 (2004), pp. 1200–1212.
[18] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising by sparse 3-d transform-domain collaborative filtering*, IEEE Trans. Image Process., 16 (2007), pp. 2080–2095.
[19] A. Danielyan, V. Katkovnik, and K. Egiazarian, *BM3D frames and variational image deblurring*, IEEE Trans. Image Process., 21 (2012), pp. 1715–1728.
[20] Y. Dong, M. Hintermüller, and M. Neri, *An efficient primal-dual method for l^1 tv image restoration*, SIAM J. Imaging Sci., 2 (2009), pp. 1168–1189.
[21] M. Elad and M. Aharon, *Image denoising via sparse and redundant representations over learned dictionaries*, IEEE Trans. Image Process., 15 (2006), pp. 3736–3745.
[22] M. Elad, B. Matalon, and M. Zibulevsky, *Image denoising with shrinkage and redundant representations*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 1924–1931.
[23] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, *Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)*, Appl. Comput. Harmonic Analy., 19 (2005), pp. 340–358.
[24] M.-J. Fadili, J.-L. Starck, and F. Murtagh, *Inpainting and zooming using sparse representations*, Comput. J., 52 (2009), pp. 64–79.
[25] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, *Optimal parameter selection for the alternating direction method of multipliers (admm): Quadratic problems*, IEEE Trans. Automat. Control, 60 (2015), pp. 644–658.
[26] T. Goldstein and S. Osher, *The split Bregman method for l1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.
[27] O. G. Guleryuz, *Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising—Part* II: *Adaptive algorithms*, IEEE Trans. Image Process., 15 (2006), pp. 555–571.
[28] W. Guo, J. Qin, and W. Yin, *A new detail-preserving regularization scheme*, SIAM J. Imaging Sci., 7 (2014), pp. 1309–1334.
[29] E. T. Hale, W. Yin, and Y. Zhang, *Fixed-point continuation for \L_1-minimization: Methodology and convergence*, SIAM J. Optim., 19 (2008), pp. 1107–1130.
[30] S. Hawe, M. Kleinsteuber, and K. Diepold, *Analysis operator learning and its application to image reconstruction*, in Proceedings of the IEEE Trans. Image Process., 22 (2013), pp. 2138–2150.
[31] B. He, Y. You, and X. Yuan, *On the convergence of primal-dual hybrid gradient algorithm*, SIAM J. Imaging Sci., 7 (2014), pp. 2526–2537.
[32] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 2012.
[33] Y.-M. Huang, M. K. Ng, and Y.-W. Wen, *A new total variation method for multiplicative noise removal*, SIAM J. Imaging Sci., 2 (2009), pp. 20–40.
[34] F. Li and T. Zeng, *A universal variational framework for sparsity based image inpainting*, IEEE Trans. Image Process., 23 (2014), pp. 4242–4254.
[35] T. Lin, S. Ma, and S. Zhang, *On the Global Linear Convergence of the Admm with Multi-Block Variables*, preprint, arXiv:1408.4266, 2014.
[36] S. Ma, W. Yin, Y. Zhang, and A. Chakraborty, *An efficient algorithm for compressed mr imaging using total variation and wavelets*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[37] M. Nikolova, M. K. Ng, and C.-P. Tam, *Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction*, IEEE Trans. Image Process., 19 (2010), pp. 3073–3088.
[38] J. Nocedal and S. J. Wright, *Penalty and Augmented Lagrangian Methods*, Springer, New York, 2006.
[39] I. Ram, M. Elad, and I. Cohen, *Image processing using smooth ordering of its patches*, IEEE Trans. Image Process., 22 (2013), pp. 2764–2774.
[40] S. Setzer, *Operator splittings, bregman methods and frame shrinkage in image processing*, Int. J. Comput. Vision, 92 (2011), pp. 265–280.
[41] J. Shen and T. F. Chan, *Mathematical models for local nontexture inpaintings*, SIAM J. Appl. Math., 62 (2002), pp. 1019–1043.

[42] J. Shen, S. H. Kang, and T. F. Chan, *Euler's elastica and curvature-based inpainting*, SIAM J. Appl. Math., 63 (2003), pp. 564–592.
[43] X.-C. Tai, S. Osher, and R. Holm, *Image inpainting using a TV-Stokes equation*, in Image Processing Based on Partial Differential Equations, Springer, New York, 2007, pp. 3–22.
[44] 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.
[45] Y.-W. Wen, R. H. Chan, and A. M. Yip, *A primal–dual method for total-variation-based wavelet domain inpainting*, IEEE Trans. Image Process., 21 (2012), pp. 106–114.
[46] A. Wong and J. Orchard, *A nonlocal-means approach to exemplar-based inpainting*, in Proceedings of the 15th IEEE International Conference on Image Processing, 2008, pp. 2600–2603.
[47] Z. Xu and J. Sun, *Image inpainting by patch propagation using patch sparsity*, IEEE Trans. Image Process., 19 (2010), pp. 1153–1165.
[48] X. Ye and H. Zhou, *Fast total variation wavelet inpainting via approximated primal-dual hybrid gradient algorithm*, Inverse Problems Imaging, 7 (2013), pp. 1031–1050.
[49] G. Yu, G. Sapiro, and S. Mallat, *Solving inverse problems with piecewise linear estimators: From Gaussian mixture models to structured sparsity*, IEEE Trans. Image Process., 21 (2012), pp. 2481–2499.
[50] X. Zhang, M. Burger, X. Bresson, and S. Osher, *Bregmanized nonlocal regularization for deconvolution and sparse reconstruction*, SIAM J. Imaging Sci., 3 (2010), pp. 253–276.
[51] X. Zhang and T. F. Chan, *Wavelet inpainting by nonlocal total variation*, Inverse Problems Imaging, 4 (2010), pp. 191–210.
[52] X.-L. Zhao, W. Wang, T.-Y. Zeng, T.-Z. Huang, and M. K. Ng, *Total variation structured total least squares method for image restoration*, SIAM J. Sci. Comput., 35 (2013), pp. B1304–B1320.