LINEARIZED BREGMAN ITERATIONS FOR FRAME-BASED IMAGE DEBLURRING

JIAN-FENG CAI*, STANLEY OSHER†, AND ZUOWEI SHEN‡

Abstract. Real images usually have sparse approximations under some tight frame systems derived from framelets, an oversampled discrete (window) cosine, or a Fourier transform. In this paper, we propose a method for image deblurring in tight frame domains. It is reduced to finding a sparse solution of a system of linear equations whose coefficient matrix is rectangular. Then, a modified version of the linearized Bregman iteration proposed and analyzed in [10, 11, 44, 51] can be applied. Numerical examples show that the method is very simple to implement, robust to noise, and effective for image deblurring.

Key words. linearized Bregman iteration, image deblurring, tight frame

AMS subject classifications. 94A08, 65T60, 90C90

1. Introduction. Image deblurring is a fundamental problem in image processing, since many real life problems can be modeled as deblurring problems; see, for instance, [20]. This paper proposes a deblurring algorithm by modifying the linearized Bregman iteration given and analyzed in [10, 11, 23, 44, 51].

For simplicity of notation, we shall denote images as vectors in $\mathbb{R}^N$ by concatenating their columns. Let $\mathbf{f} \in \mathbb{R}^N$ be the underlying image. Then the observed blurred image $\mathbf{g} \in \mathbb{R}^N$ is given by

$$\mathbf{g} = B\mathbf{f} + \mathbf{n}, \tag{1.1}$$

where $\mathbf{n} \in \mathbb{R}^N$ is a vector of noise and $B \in \mathbb{R}^{N\times N}$ is a linear blurring operator. This problem is ill-posed due to the large condition number of the matrix B. Any small perturbation on the observed blurred image $\mathbf{g}$ may cause the direct solution $B^{-1}\mathbf{g}$ to be very far away from the original image $\mathbf{f}$; see, for instance, [2, 49]. This is a widely studied subject, and there are many different approaches.

One of the main ideas is to find the solution that minimizes some cost functionals; see [20] and its references. The simplest method is Tikhonov regularization, which minimizes an energy consisting of a data fidelity term and an ℓ_2 norm regularization term. When B is a convolution, one can solve the problem in the Fourier domain. In this case, the method is called a Wiener filter [1]. This is a linear method, and the edges of restored image are usually smeared. To overcome this, a regularization based on total variation (TV) was proposed by Rudin, Osher, and Fatemi in [47]; it is well known as the ROF model. Due to its virtue of preserving edges, it is widely used in the many applications of image processing, such as blind deconvolution, inpainting and superresolution; see [20] for an overview. However, it is well known that TV yields staircasing (see [28, 42]); hence these TV-based methods do not preserve the

*Temasek Laboratories, National University of Singapore, 2 Science Drive 2, Singapore 117543. Email: tslcaij@nus.edu.sg. Research supported by the Wavelets and Information Processing Programme under a grant from DSTA, Singapore.

†Department of Mathematics, UCLA, 520 Portola Plaza, Los Angeles, CA 90095, USA. Email:sjo@math.ucla.edu. Research partially supported by ONR grant N000140710810, and by U.S. Department of Defense.

‡Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543. Email: matzuows@nus.edu.sg. Research supported in part by Grant R-146-000-113-112 at the National University of Singapore.

fine structures, details, and textures. To avoid these drawbacks, nonlocal methods were proposed for denoising in, e.g. [5, 48], and then extended to deblurring in [6]. Also, Bregman iteration, introduced to image science in [43], was shown to improve TV-based blind deconvolution in [35, 40, 41]. Recently, inspired by graph theory, a nonlocal TV regularization was invented in [32] and applied to image deblurring in [38]. We will compare the algorithm given in this paper with deblurring algorithms based on the nonlocal operator in [38]. As a consequence, it also gives comparisons with other algorithms in [1, 6, 47], as they are already compared with the nonlocal operator approach in [38]. The experiments in [38] were recently redone using the split Bregman approach of [33] rather than gradient descent, with apparently better results. Our comparisons here were taken with the original results in [38].

Another approach for deblurring is the wavelet-based method. This approach is based on the fact that most real images usually have sparse approximations under some tight frame systems, which include, for example, translation invariant orthonormal wavelets (see, e.g., [22]), discrete Fourier transforms, Gabor transforms, oversampled discrete cosine transforms, discrete local cosine transforms (see, e.g., [24, 39]), framelets (see, e.g., [26, 46]), and curvelets (see, e.g., [13]). One can solve (1.1) in a tight frame domain that has a sparse approximation of the underlying solution. The redundancy of systems leads to robust signal representation in which partial loss of the data can be tolerated without adverse effects. In order to obtain the sparse approximation, we minimize the weighted ℓ_1 norm of the tight frame coefficients. This is a relatively new approach in this well-studied subject. There are a few papers on solving inverse problems, in particular deblurring problems, by using the fact that the underlying image has a sparse approximation in a certain transform domain; see, for instance, [4, 7, 12, 16–18, 25, 27, 31]. In [7, 12, 16–18], the deconvolution problem is converted to an inpainting problem in a frame domain by designing a tight wavelet frame system via the unitary extension principle of [46], with the convolution kernel being one of the masks of the tight frames. We further note that the approach taken by [27] can be understood as solving a Lagrange relaxed minimization problem in frame domains. Some numerical comparisons of the approach here and these two approaches are given.

In this paper, we use a modified version of the linearized Bregman iteration in [10, 11, 23, 44, 51], which is very efficient in finding a sparse solution of a system of underdetermined linear equations, to solve the deblurring problem (1.1) in a tight frame domain. The algorithm given in this paper solves the deblurring problem directly in the frame domain, and there is no need to construct a tight frame system so that one of the masks is the given blurring kernel. It is, in particular, useful when it is impossible to construct a tight frame system with the blurring kernel as one of the masks.

The rest of the paper is organized as follows. In Section 2, we modify the linearized Bregman iteration in order to suit image deblurring for invertible B. Then we extend the results to noninvertible B in Section 3. Section 4 is devoted to algorithms and numerical experiments.

2. Frame-based deblurring. We start with a short review of the linearized Bregman iteration given and analyzed in [10, 11, 23, 44, 51]. Then, we reformulate the linearized Bregman iteration from the frame point of view. This, in turn, leads to our modified algorithm to solve (1.1) in frame domain.

2.1. Linearized Bregman iteration. Iterative algorithms involving Bregman distance were introduced to image and signal processing by [14,15] and by many other

authors. See [43] for an overview. In [43], a Bregman iteration was proposed for the nondifferentiable TV energy for image restoration. Then, in [51], it was shown to be remarkably successful for ℓ_1 norm minimization problems in compressive sensing. To further improve the performance of the Bregman iteration, a linearized Bregman iteration was invented in [23]; see also [51]. More details and an improvement called "kicking" of the linearized Bregman iteration was described in [44], and a rigorous theory was given in [10, 11]. Recently, a new type of iteration based on Bregman distance, called split Bregman iteration, was introduced in [33], which extended the utility of the Bregman iteration and the linearized Bregman iteration to minimizations of more general ℓ_1-based regularizations including TV, Besov norms, and sums of such things. Wavelet-based denoising using the Bregman iteration was introduced in [50], and it was further extended by using translation invariant wavelets in [37]. Here we focus on the linearized Bregman iteration.

Let A be an $N \times M$, $M \geq N$, surjective matrix. The aim of the linearized Bregman iteration is to find an approximation of a solution of $A\mathbf{u} = \mathbf{g}$ that has a minimal ℓ_1 norm among all the solutions, i.e., to approximately solve

$$\min_{\mathbf{u}\in\mathbb{R}^M} \{\|\mathbf{u}\|_1 : A\mathbf{u} = \mathbf{g}\}. \tag{2.1}$$

The iteration is

$$\begin{cases} \mathbf{v}^{k+1} = \mathbf{v}^k - A^T(A\mathbf{u}^k - \mathbf{g}), \\ \mathbf{u}^{k+1} = \delta T_\mu(\mathbf{v}^{k+1}), \end{cases} \tag{2.2}$$

where $\mathbf{u}^0 = \mathbf{v}^0 = 0$, and

$$T_\mu(\mathbf{w}) := [t_\mu(w_1), t_\mu(w_2), \ldots, t_\mu(w_M)]^T \tag{2.3}$$

with

$$t_\mu(\xi) = \begin{cases} 0 & \text{if } |\xi| \leq \mu, \\ \operatorname{sgn}(\xi)(|\xi| - \mu) & \text{if } |\xi| > \mu. \end{cases} \tag{2.4}$$

Iteration (2.2) is very simple to program, involving only matrix-vector multiplications and scalar shrinkages. It was applied to basis pursuit problems arising in compressed sensing in [11, 44]. The algorithm works based on the facts that the matrix-vector multiplications have fast algorithms in most applications in compressed sensing and the solution is sparse. The shrinkage operator makes the solution sparse and removes noises. Hence, this algorithm converges to a sparse solution and is robust to noise.

It is proven in [11] that, if $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ generated by (2.2) converges, its limit is the unique solution of (2.5), which is given below. It was also shown in [11] that the limit of (2.2) becomes a solution of the basis pursuit problem (2.1) as $\mu \to \infty$. Furthermore, it was shown in [11] that the corresponding linearized Bregman iteration converges when a smoothed ℓ_1 norm is used. However, the convergence of the sequence $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ of (2.2) is given in [10]. We summarize the above results from [10,11] into the following proposition. Throughout this paper, the notation $\|\cdot\|$ is always used for the ℓ_2 norm and $\|\cdot\|_1$ for the ℓ_1 norm of vectors.

PROPOSITION 2.1. *Assume that $A \in \mathbb{R}^{N\times M}$, $M \geq N$, is surjective. Then the sequence $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ generated by (2.2) with $0 < \delta < \frac{1}{\|AA^T\|}$ converges to the unique*

solution of

$$\min_{\mathbf{u}\in\mathbb{R}^M}\left\{\mu\|\mathbf{u}\|_1+\frac{1}{2\delta}\|\mathbf{u}\|^2 : A\mathbf{u}=\mathbf{g}\right\}. \tag{2.5}$$

Furthermore, when $\mu\to\infty$, the limit of (2.2) *tends to the solution of* (2.1) *that has a minimal ℓ_2 norm among all the solutions of* (2.1).

In addition, for general A, the following lemma was proved in [10].

LEMMA 2.2. *Let $A\in\mathbb{R}^{N\times M}$, $M\geq N$, be any matrix. Assume that $0<\delta<\frac{1}{\|AA^T\|}$. Then, the sequences $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ and $\{\mathbf{v}^k\}_{k\in\mathbb{N}}$ generated by* (2.2) *are bounded, and*

$$\lim_{k\to\infty} A^T(A\mathbf{u}^k-\mathbf{g})=0.$$

Proposition 2.1 does not reveal the convergence speed of the linearized Bregman iteration (2.2). In fact, the convergence speed depends on the condition number of A, as shown in [11] by a smoothed version of (2.2). Here the condition number of A is defined by $\mathrm{cond}(A)=\sqrt{\frac{\lambda_1(AA^T)}{\lambda_N(AA^T)}}$, where $\lambda_1(AA^T)$ and $\lambda_N(AA^T)$ are the first (maximum) and Nth (minimum) eigenvalues of the symmetric positive definite matrix AA^T, respectively. A smaller condition number of A leads to a faster convergence; see [11] for details.

In some applications of compressed sensing, A is usually a rectangular matrix consisting of partial rows of a fast transform matrix such as discrete cosine transform. In this case, $AA^T=I$. In (2.2), since $\mathbf{v}^0=0$ is in the range of A^T, by induction, $\mathbf{v}^k$ is in the range of A^T. Therefore, we can write $\mathbf{v}^k=A^T\mathbf{g}^k$. By substituting this into (2.2), we obtain

$$A^T\mathbf{g}^{k+1}=A^T\mathbf{g}^k-A^T(A\mathbf{u}^k-\mathbf{g}).$$

Since A is surjective, this leads to that the linearized Bregman iteration (2.2) is equivalent to

$$\begin{cases}\mathbf{g}^{k+1}=\mathbf{g}^k+(\mathbf{g}-A\mathbf{u}^k),\\ \mathbf{u}^{k+1}=\delta T_\mu(A^T\mathbf{g}^{k+1}),\end{cases} \tag{2.6}$$

where $\mathbf{u}^0=\mathbf{g}^0=0$. Formulation (2.6) leads to the following explanation: The first step is to add the error of the constraint $A\mathbf{u}=\mathbf{g}$ to the current data $\mathbf{g}^k$ to get the new data $\mathbf{g}^{k+1}$. The second step can be divided into two substeps. The first substep derives the minimal ℓ_2 norm solution $A^T\mathbf{g}^{k+1}$ of the equation $A\mathbf{u}=\mathbf{g}^{k+1}$ by using $AA^T=I$. This substep removes part of the noise contained in the kernel of A. The second substep shrinks the minimal ℓ_2 norm solution by applying the shrinkage operator. This substep removes noise as well. Furthermore, this step, together with the first step of updating the residues, keeps big entries and removes small entries from the solution. Altogether, the linearized Bregman iteration leads to a sparse limit and is robust to noise.

2.2. Modified linearized Bregman iteration. In this subsection, we modify (2.6) to accelerate its convergence in the case of $AA^T\neq I$. When $AA^T\neq I$, but A is still surjective, i.e., AA^T is positive definite, the minimal ℓ_2 norm solution of $A\mathbf{u}=\mathbf{g}^{k+1}$ is given by $A^\dagger\mathbf{g}^{k+1}$, where $A^\dagger=A^T(AA^T)^{-1}$. Notice that $A^\dagger$ is the

pseudoinverse of A. Therefore, we modify iteration (2.6) by replacing $A^T\mathbf{g}^{k+1}$ in (2.6) by $A^\dagger\mathbf{g}^{k+1}$. This leads to the following modified iteration:

$$\begin{cases}\mathbf{g}^{k+1} = \mathbf{g}^k + (\mathbf{g} - A\mathbf{u}^k),\\ \mathbf{u}^{k+1} = \delta T_\mu(A^\dagger\mathbf{g}^{k+1}),\end{cases} \tag{2.7}$$

where $\mathbf{u}^0 = \mathbf{g}^0 = 0$. Notice that (2.7) is not the same as (2.6) if $AA^T \neq I$. At each iteration, the second step shrinks the minimum ℓ_2 norm solution. This makes the solution sparse and removes noises. This also introduces the errors to the minimal ℓ_2 solution. The first step is to correct it by updating the residues generated from the second step. The following theorem says that this simple iteration converges to a sparse solution of the original equation $A\mathbf{u} = \mathbf{g}$.

THEOREM 2.3. *Assume that $A \in \mathbb{R}^{N\times M}$, $M \geq N$, is surjective. Then the sequence $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ generated by iteration (2.7) with $0 < \delta < 1$ converges to the unique solution of (2.5), or, equivalently,*

$$\min_{\mathbf{u}\in\mathbb{R}^M}\left\{\mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|\mathbf{u} - A^\dagger\mathbf{g}\|^2 : A\mathbf{u} = \mathbf{g}\right\}. \tag{2.8}$$

Furthermore, as $\mu \to \infty$, the limit of (2.7) is the solution of (2.1) that is closest to the minimal ℓ_2 norm solution of $A\mathbf{u} = \mathbf{g}$ among all the solutions of (2.1).

Proof. The theorem is proven by applying Proposition 2.1. By multiplying both sides of the first equation in (2.7) by $A^\dagger$, we obtain

$$\begin{cases}A^\dagger\mathbf{g}^{k+1} = A^\dagger\mathbf{g}^k + A^\dagger(\mathbf{g} - A\mathbf{u}^k),\\ \mathbf{u}^{k+1} = \delta T_\mu(A^\dagger\mathbf{g}^{k+1}).\end{cases}$$

Let $\mathbf{v}^{k+1} = A^\dagger\mathbf{g}^{k+1}$. Then, the above formulation leads to

$$\begin{cases}\mathbf{v}^{k+1} = \mathbf{v}^k - A^T(AA^T)^{-1}(A\mathbf{u}^k - \mathbf{g}),\\ \mathbf{u}^{k+1} = \delta T_\mu(\mathbf{v}^{k+1}),\end{cases} \tag{2.9}$$

where $\mathbf{u}^0 = \mathbf{v}^0 = 0$. This is the linearized Bregman iteration (2.2) for the system of linear equations

$$(AA^T)^{-1/2}A\mathbf{u} = (AA^T)^{-1/2}\mathbf{g}. \tag{2.10}$$

In fact, by replacing A and $\mathbf{g}$ in (2.2) by $(AA^T)^{-1/2}A$ and $(AA^T)^{-1/2}\mathbf{g}$, respectively, we obtain (2.9).

Since (2.7) leads to the linearized Bregman iteration for (2.10), Proposition 2.1 can be applied to prove the convergence of (2.7). Because A is surjective, the matrix $(AA^T)^{-1/2}A$ is rectangular and surjective. We have

$$\begin{aligned}\|\big((AA^T)^{-1/2}A\big)\big((AA^T)^{-1/2}A\big)^T\|_2 &= \|\big((AA^T)^{-1/2}A\big)^T\big((AA^T)^{-1/2}A\big)\|_2\\ &= \|A^T(AA^T)^{-1}A\|_2 = 1.\end{aligned}$$

Therefore, with $0 < \delta < 1$, Proposition 2.1 guarantees the convergence of (2.7) to the solution of

$$\min_{\mathbf{u}\in\mathbb{R}^M}\left\{\mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|\mathbf{u}\|^2 : (AA^T)^{-1/2}A\mathbf{u} = (AA^T)^{-1/2}\mathbf{g}\right\}. \tag{2.11}$$

Because $(AA^T)^{-1/2}$ is invertible, (2.10) is equivalent to $A\mathbf{u} = \mathbf{g}$; hence (2.11) is equivalent to (2.5).

It remains to show the equivalence between (2.5) and (2.8). Note that $A^\dagger A$ is a projection. The equality in (2.8) simply follows from

$$\|\mathbf{u}\|_2^2 = \|A^\dagger A\mathbf{u}\|_2^2 + \|(I - A^\dagger A)\mathbf{u}\|_2^2 = \|A^\dagger \mathbf{g}\|_2^2 + \|\mathbf{u} - A^\dagger \mathbf{g}\|_2^2 \tag{2.12}$$

and the fact that $\|A^\dagger \mathbf{g}\|_2^2$ is a constant. □

Here we make some remarks. First, the system of linear equations (2.10) is actually a "preconditioned" version of the system of linear equations $A\mathbf{u} = \mathbf{g}$. As a result, the condition number of the matrix $(AA^T)^{-1/2}A$ is one, since all the eigenvalues of $A^T(AA^T)^{-1}A$ are one. Hence, it can be used when the condition number of A is large, since the convergence rate depends on the condition number of the coefficient matrix as shown in [11] by a smoothed version.

Second, the term $\frac{1}{2}\|\mathbf{u} - A^\dagger \mathbf{g}\|_2^2$ in (2.8) is the distance from $\mathbf{u}$ to the minimal ℓ_2 norm solution $A^\dagger \mathbf{g}$ of $A\mathbf{u} = \mathbf{g}$. Therefore, the modified linearized Bregman iteration (2.7) leads to a solution that is less sparse than if there were no quadratic term. This is particularly useful in frame-based image restorations, as shown in [7,12,16–18,27,31].

Finally, the modified linearized Bregman iteration (2.7) is very cheap whenever there is a fast algorithm for finding the minimal ℓ_2 norm solution of $A\mathbf{u} = \mathbf{g}$. However, this is not true in general. In the next section, we will discuss how to make it happen for the image deblurring in frame domain.

2.3. Deblurring with full rank blurring matrix. In this subsection, we apply the linearized Bregman iteration (2.7) to the frame-based image deblurring. Recall that the column vectors of

$$A := [\mathbf{a}_1, \ldots, \mathbf{a}_M] \in \mathbb{R}^{N\times M}$$

form a frame in $\mathbb{R}^N$ if there exist two positive numbers a and b such that

$$a\|\mathbf{h}\|_2^2 \le \sum_{i=1}^{M} |\langle \mathbf{h}, \mathbf{a}_i\rangle|^2 \le b\|\mathbf{h}\|_2^2 \quad \forall\ \mathbf{h} \in \mathbb{R}^N. \tag{2.13}$$

It is obvious that A is a frame if and only if it is surjective. When $a = b = 1$, the column vectors of A form a tight frame, for which (2.13) is equivalent to

$$AA^T = I.$$

When $N = M$, then A is a basis in $\mathbb{R}^N$. When $N < M$, the system is redundant, since the rank of A is N. In this case, for a given vector $\mathbf{g}$, there are infinitely many representations of $\mathbf{g}$ that correspond to solutions of $A\mathbf{u} = \mathbf{g}$. Among all the solutions, the one $\mathbf{u}^* = A^T(AA^T)^{-1}\mathbf{g}$, called the canonical coefficients of $\mathbf{g}$, has a minimal ℓ_2 norm. However, in many applications, it is more important to find coefficients $\mathbf{u}$ of the vector $\mathbf{g}$ that minimize the ℓ_1 norm. Frame-based imaging restorations are widely used, and they can be found in, e.g., [7,12,16–18,27,30,31].

Now we turn to the deblurring problem (1.1). In this section, we assume that the blurring matrix B is invertible. We solve (1.1) in a tight frame domain. Let F be an $N \times M$ matrix whose column vectors form a tight frame in $\mathbb{R}^N$, i.e., $FF^T = I$. Further, we assume that the underlying solution $\mathbf{f}$ has a sparse approximation under F. Then, in the tight frame domain, (1.1) becomes

$$\mathbf{g} = BF\mathbf{u} + \mathbf{n}, \qquad \text{where} \qquad F\mathbf{u} = \mathbf{f}. \tag{2.14}$$

We start our discussion with the simple case when $\mathbf{n} = 0$. Then the model (2.14) becomes

$$BF\mathbf{u} = \mathbf{g}. \tag{2.15}$$

Since $\mathbf{f}$ has a sparse approximation under F, we would like to find a sparse solution of (2.15), i.e., to solve

$$\min_{\mathbf{u}\in\mathbb{R}^M} \{\|\mathbf{u}\|_1 : BF\mathbf{u} = \mathbf{g}\}. \tag{2.16}$$

Write $A = BF$. Because B is invertible and F is a tight frame, the matrix A is rectangular and surjective. Hence the column vectors of A form a frame in $\mathbb{R}^N$. Therefore, we use the modified linearized Bregman iteration (2.7) to solve the deblurring problem. Since $AA^T = BF(BF)^T = BB^T$, the pseudoinverse is $A^\dagger = A^T(AA^T)^{-1} = F^TB^T(BB^T)^{-1}$.

It is generally hard to find $(BB^T)^{-1}$ because usually B has a bad condition number. We use a symmetric positive definite, invertible matrix P to approximate it, i.e., $P \approx (BB^T)^{-1}$. Then, the modified linearized Bregman iteration for the frame-based deblurring becomes

$$\begin{cases} \mathbf{g}^{k+1} = \mathbf{g}^k + (\mathbf{g} - BF\mathbf{u}^k), \\ \mathbf{u}^{k+1} = \delta T_\mu(F^TB^TP\mathbf{g}^{k+1}), \end{cases} \tag{2.17}$$

where $\mathbf{u}^0 = \mathbf{g}^0 = 0$, and P is a symmetric positive definite matrix. In the first step, we update the data by adding the error of $BF\mathbf{u} = \mathbf{g}$ into the current data $\mathbf{g}^k$ to get the data $\mathbf{g}^{k+1}$; in the second step, we threshold the coefficient $F^TB^TP\mathbf{g}^{k+1}$, which approximates $F^TB^T(BB)^{-1}\mathbf{g}^{k+1}$, the minimal ℓ_2 norm solution of $BF\mathbf{u} = \mathbf{g}^{k+1}$.

Applying Proposition 2.1, we have the following result for (2.17).

THEOREM 2.4. *Assume that P is symmetric positive definite, F is a tight frame, and B is invertible. Let δ be such that $0 < \delta < \frac{1}{\|B^TPB\|}$. Then the sequence $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ generated by iteration* (2.17) *converges to the unique solution of*

$$\min_{\mathbf{u}\in\mathbb{R}^M} \left\{ \mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|\mathbf{u}\|^2 : BF\mathbf{u} = \mathbf{g} \right\}, \tag{2.18}$$

Furthermore, as $\mu \to \infty$, the limit of (2.17) *is the solution of* (2.16) *that has a minimum ℓ_2 norm among all the solutions of* (2.18).

Proof. Because P is symmetric positive definite, its square root $P^{1/2}$ exists. Similar to the proof of Theorem 2.3, the linearized Bregman iteration (2.2) for the system of linear equations

$$P^{1/2}BF\mathbf{u} = P^{1/2}\mathbf{g} \tag{2.19}$$

is, by replacing A and $\mathbf{g}$ in (2.2) by $P^{1/2}BF$ and $P^{1/2}\mathbf{g}$, respectively,

$$\begin{cases} \mathbf{v}^{k+1} = \mathbf{v}^k - F^TB^TP(BF\mathbf{u}^k - \mathbf{g}), \\ \mathbf{u}^{k+1} = \delta T_\mu(\mathbf{v}^{k+1}), \end{cases}$$

where $\mathbf{u}^0 = \mathbf{v}^0 = 0$. Since $\mathbf{v}^0 = 0$ is in the range of F^TB^TP, by induction, $\mathbf{v}^k$ is also in the range of F^TB^TP. Let $\mathbf{v}^k = F^TB^TP\mathbf{g}^k$. Then, one derives (2.17).

By the assumptions, the matrix $P^{1/2}BF$ is rectangular and surjective. By applying Proposition 2.1 and the invertibility of $P^{1/2}$, (2.17) converges to the unique solution of (2.18) provided that $0 < \delta < \frac{1}{\|(P^{1/2}BF)(P^{1/2}BF)^T\|}$, which is satisfied by $\|(P^{1/2}BF)(P^{1/2}BF)^T\| = \|P^{1/2}BB^TP^{1/2}\| = \|B^TPB\|$. □

Again, in the above theorem, the system of linear equations (2.19) is a "preconditioned" version of $BF\mathbf{u} = \mathbf{g}$. Since $P \approx (BB^T)^{-1}$, and the condition number of $(P^{1/2}BF)(P^{1/2}BF)^T = P^{1/2}BB^TP^{1/2}$ equals the square root of the ratio of the maximum to the minimum eigenvalues of PBB^T, the condition number of $(P^{1/2}BF)(P^{1/2}BF)^T$ is close to 1. On the other hand, since B is ill-conditioned, the matrix $(BF)(BF)^T = BB^T$ has a large condition number. Therefore, iteration (2.17) converges faster than the original linearized Bregman iteration (2.2) with $A = BF$.

The choice of P can be done, for example, as follows. The blurring operator B is usually a convolution operator from $\mathbb{R}^N$ to $\mathbb{R}^N$, and it is a Toeplitz-like or block-Toeplitz-like matrix with a suitable boundary condition. Hence, B can be efficiently approximated by a circulant matrix or a fast transform (e.g., the discrete cosine transform (DCT)) based matrix C; see, e.g., [19, 36]. This means that we use convolution matrices with circular or Neumann boundary conditions to approximate B. Therefore, $P = (CC^T)^{-1}$ is a good approximation for $(BB^T)^{-1}$. In order for the approximation to be numerically stable and robust to noise, we choose $P = (CC^T + \theta GG^T)^{-1}$, where θ is a small positive number and G is the identity or a difference matrix with a circular or Neumann boundary condition. To make sure that P is well defined, we require that either C is invertible or $\mathrm{Ker}(CC^T) \cap \mathrm{Ker}(GG^T) = \{0\}$. Here Ker denotes the kernel space of a matrix. Notice that $(CC^T + \theta GG^T)^{-1}\mathbf{h}$ for any $\mathbf{h}$ can be computed efficiently by fast Fourier transforms (FFTs) or DCTs; see [19]. This enables iteration (2.17) to be implemented quickly.

Since B is invertible, we have

$$\|\mathbf{u}\|^2 = \|F^TF\mathbf{u}\|^2 + \|(I - F^TF)\mathbf{u}\|^2 = \|F^TB^{-1}\mathbf{g}\|^2 + \|(I - F^TF)\mathbf{u}\|^2. \tag{2.20}$$

This implies that (2.18) is equivalent to

$$\min_{\mathbf{u}\in\mathbb{R}^M} \left\{ \mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|(I - F^TF)\mathbf{u}\|^2 : BF\mathbf{u} = \mathbf{g} \right\}. \tag{2.21}$$

Therefore, by Theorem 2.4, the modified iteration (2.17) converges to the solution of (2.18). The first term in (2.18), $\mu\|\mathbf{u}\|_1$, is to penalize the ℓ_1 norm of $\mathbf{u}$. This keeps the solution $\mathbf{u}$ sparse. The second term is to penalize the distance of $\mathbf{u}$ to the range of F, so it makes $\mathbf{u}$ close to its canonical tight frame coefficient. By the theory in [3, 34], the weighted ℓ_1 norm of the canonical coefficient is related to the Besov norm of the underlying solution when F is a tight wavelet frame (or so-called framelet) system. Therefore, these two terms together balance the sparsity of the tight frame coefficient and the regularity (the smoothness) of the underlying solution that has the same flavor of the algorithm given in [7, 8, 12].

Similar to the linearized Bregman iteration (2.6), iteration (2.17) is robust to noise. Indeed, there are two procedures in (2.17) to suppress the noise. In the first step of (2.17), we add the error of the equation $BF\mathbf{u} = \mathbf{g}$ to the current data $\mathbf{g}^k$ to obtain the new data $\mathbf{g}^{k+1}$. The second step consists of two substeps. The first substep is to get $P\mathbf{g}^{k+1}$. If we choose $P = (CC^T + \theta GG^T)^{-1}$, then $P\mathbf{g}^{k+1}$ is a solution of

$$P\mathbf{g}^{k+1} = \arg\min_{\mathbf{f}\in\mathbb{R}^N} \left\{ \|\mathbf{g}^{k+1} - CC^T\mathbf{f}\|^2_{(CC^T)^{-1}} + \theta\|G^T\mathbf{f}\|_2^2 \right\},$$

where $\|\mathbf{a}\|^2_{(CC^T)^{-1}} = \mathbf{a}^T(CC^T)^{-1}\mathbf{a}$ is a weighted norm. Therefore, this sub-step is to solve the approximated deblurring problem $CC^T\mathbf{f} = \mathbf{g}^{k+1}$ with a Tikhonov regularization. Hence, this substep can remove noise. The second substep is the shrinkage of $F^TB^TP\mathbf{g}^{k+1}$, the approximated minimal ℓ_2 norm solution of $BF\mathbf{u} = \mathbf{g}^{k+1}$. The purpose is to keep big entries and remove small entries to keep the solution sparse. Therefore, this substep removes noise as well.

In summary, iteration (2.17) is robust to noise due to the noise removals embedded. Therefore, when the data $\mathbf{g}$ contains noise, i.e., $\mathbf{n} \neq 0$, we can still use algorithm (2.17) but stop it before it converges. The stopping criteria is important. If we stop the iteration early, the details and the edges may not be recovered well; if we stop it late, the artifacts caused by the noise show up. A natural choice of the stopping criteria is that we stop the iteration whenever

$$\|\mathbf{g} - BF\mathbf{u}^k\|^2 \lessapprox \sigma^2, \tag{2.22}$$

where σ^2 is the variance of the noise $\mathbf{n}$. With this stopping criteria, we get deblurred images before the artifacts show up, as pointed in [43]. We remark that, given a noisy blurred image, σ^2 can be estimated by, for example, the method in [29].

3. Generalization to noninvertible blurring matrices. When the blurring matrix B is not invertible, then the matrix $A = BF$ is not surjective; hence the results presented in the previous section are not applicable. In this section, we show that iterations (2.2), (2.7), and (2.17) can be applied to the case that the matrix A is no longer surjective to find a (weighted) least square solution with sparsity. Therefore, even when B is not invertible, we can still use the linearized Bregman iterations to find a (weighted) least square solution with sparsity. Although it is motivated by the deblurring when the blurring matrix B is not invertible, it extends to many other general settings, i.e., to finding a sparse solution of the least square problem. In the first part of this section, we will extend (2.1) to the case that A is not a surjection, and the second part of this section is devoted to the deblurring for noninvertible blurring matrices.

3.1. Least Square Solutions. As before, we first assume that there is no noise in $\mathbf{g}$, i.e., $\mathbf{n} = \mathbf{0}$, and we need to solve $A\mathbf{u} = \mathbf{g}$. However, when A is not surjective, there may not necessarily exist a solution for the system of linear equations $A\mathbf{u} = \mathbf{g}$. Alternatively, it is natural to minimize the square error $\|A\mathbf{u} - \mathbf{g}\|^2$. Since A is a rectangular matrix, there are infinitely many solutions of

$$\min_{\mathbf{u}\in\mathbb{R}^M} \|A\mathbf{u} - \mathbf{g}\|^2. \tag{3.1}$$

Among all the solutions of (3.1), the minimal ℓ_2 norm solution $\mathbf{u}^* := A^\dagger\mathbf{g}$, which is generally not sparse, is commonly used. Note that here $A^\dagger$ is the pseudoinverse of A and is not $A^T(AA^T)^{-1}$ since $(AA^T)^{-1}$ does not exist. However, in our deblurring problem, we would like to find a sparse solution of (3.1). This leads to a minimal ℓ_1 norm solution of (3.1), i.e., to solving

$$\min_{\mathbf{u}\in\mathbb{R}^M} \{\|\mathbf{u}\|_1 : \mathbf{u} = \arg\min_{\mathbf{u}\in\mathbb{R}^M} \|A\mathbf{u} - \mathbf{g}\|^2\}. \tag{3.2}$$

Notice that, when A is surjective, (3.2) is equivalent to (2.1).

It is interesting to know that, in order to solve (3.2), we are still able to use the linearized Bregman iteration (2.2). The next theorem says that Proposition 2.1 still holds for an arbitrary A.

THEOREM 3.1. *Let $A \in \mathbb{R}^{N\times M}$, $M \geq N$, be an arbitrary matrix. Then the sequence $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ generated by (2.2) with $0 < \delta < \frac{1}{\|AA^T\|}$ converges to the unique solution of*

$$\min_{\mathbf{u}\in\mathbb{R}^M}\left\{\mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|\mathbf{u}\|^2 : \mathbf{u} = \arg\min_{\mathbf{u}\in\mathbb{R}^M}\|A\mathbf{u}-\mathbf{g}\|^2\right\}. \tag{3.3}$$

Furthermore, when $\mu\to\infty$, the limit of (2.2) tends to the solution of (3.2) that has a minimal ℓ_2 norm among all the solutions of (3.2).

Proof. Since the least square problem (3.1) is equivalent to solving the system of linear equations $A^TA\mathbf{u} = A^T\mathbf{g}$, the minimization problem (3.3) is equivalent to

$$\min_{\mathbf{u}\in\mathbb{R}^M}\left\{\mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|\mathbf{u}\|^2 : A^TA\mathbf{u} = A^T\mathbf{g}\right\}. \tag{3.4}$$

The set of all of the solutions of $A^TA\mathbf{u} = A^T\mathbf{g}$ is a nonempty convex set. Since the energy $\mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|\mathbf{u}\|^2$ is strongly convex, the solution of (3.4) is unique. Denote the unique solution by $\mathbf{u}^*_\mu$. Therefore, we need only to prove that $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ converges to $\mathbf{u}^*_\mu$.

By Lemma 2.2, the sequence $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ is bounded. Hence, there exists at least one convergent subsequence. We prove the first part of the theorem, i.e., $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ converges to the solution of (3.3), by showing that each convergent subsequence converges to $\mathbf{u}^*_\mu$. By the uniqueness of $\mathbf{u}^*_\mu$, we conclude that $\lim_{k\to\infty}\mathbf{u}^k = \mathbf{u}^*_\mu$. Let $\{\mathbf{u}^{k_i}\}_{i\in\mathbb{N}}$ be an arbitrary convergent subsequence. Let $\widetilde{\mathbf{u}} := \lim_{i\to\infty}\mathbf{u}^{k_i}$. Next we show that $\widetilde{\mathbf{u}} = \mathbf{u}^*_\mu$.

Since $\mathbf{v}^0 = \mathbf{u}^0 = 0$, by summing up the first equation of (2.2) from 0 to k, we have

$$\mathbf{v}^k = A^T\mathbf{g}^k, \quad \text{where} \quad \mathbf{g}^k = \sum_{j=0}^{k-1}(\mathbf{g} - A\mathbf{u}^j). \tag{3.5}$$

We decompose $\mathbf{g}^k$ into the direct sum of

$$\mathbf{g}^k = \mathbf{g}^k_{\mathrm{Ker}(A^T)} + \mathbf{g}^k_{\mathrm{Ran}(A)}, \tag{3.6}$$

where $\mathbf{g}^k_{\mathrm{Ker}(A^T)}$ is in the kernel of A^T, and $\mathbf{g}^k_{\mathrm{Ran}(A)}$ is in the range of A. It is obvious that these two components are orthogonal to each other. Moreover, since $\{A^T\mathbf{g}^k\}_{k\in\mathbb{N}}$ is bounded by Lemma 2.2 and A^T is one-to-one from $\mathrm{Ran}(A)$ to $\mathbb{R}^M$, we have that $\{\mathbf{g}^k_{\mathrm{Ran}(A)}\}_{k\in\mathbb{N}}$ is bounded, i.e.,

$$\|\mathbf{g}^k_{\mathrm{Ran}(A)}\| \leq C \quad \forall k. \tag{3.7}$$

Since $T_\mu(\mathbf{v}^k)$ is the solution of the minimization problem $\min_{\mathbf{u}}\frac{1}{2}\|\mathbf{v}^k-\mathbf{u}\|_2^2+\mu\|\mathbf{u}\|_1$, the first order optimality condition yields that

$$\mathbf{v}^k - T_\mu(\mathbf{v}^k) \in \partial\mu\|\mathbf{u}^k\|_1, \tag{3.8}$$

where ∂ denotes the subdifferential. Furthermore, the second equation of (2.2) gives $T_\mu(\mathbf{v}^k) = \frac{1}{\delta}\mathbf{u}^k$, which leads to

$$\mathbf{v}^k = \frac{1}{\delta}\mathbf{u}^k + (\mathbf{v}^k - T_\mu(\mathbf{v}^k)). \tag{3.9}$$

Define $H(\mathbf{u}) = \mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|\mathbf{u}\|^2$. Equations (3.8) and (3.9) imply that $\mathbf{v}^k \in \partial H(\mathbf{u}^k)$ for all k. This, together with the convexity of $H(\mathbf{u})$, leads to

$$H(\mathbf{u}^{k_i}) \le H(\mathbf{u}_\mu^*) - \langle \mathbf{u}_\mu^* - \mathbf{u}^{k_i}, \mathbf{v}^{k_i}\rangle.$$

By (3.5) and (3.6), we have

$$\begin{aligned} H(\mathbf{u}^{k_i}) &\le H(\mathbf{u}_\mu^*) - \langle \mathbf{u}_\mu^* - \mathbf{u}^{k_i}, A^T(\mathbf{g}^{k_i}_{\mathrm{Ker}(A^T)} + \mathbf{g}^{k_i}_{\mathrm{Ran}(A)})\rangle \\ &= H(\mathbf{u}_\mu^*) - \langle A(\mathbf{u}_\mu^* - \mathbf{u}^{k_i}), \mathbf{g}^{k_i}_{\mathrm{Ran}(A)}\rangle. \end{aligned} \tag{3.10}$$

Next, we estimate the second term of the right-hand side of (3.10). By the Cauchy-Schwarz inequality and (3.7), we have

$$|\langle A(\mathbf{u}_\mu^* - \mathbf{u}^{k_i}), \mathbf{g}^{k_i}_{\mathrm{Ran}(A)}\rangle| \le \|A(\mathbf{u}_\mu^* - \mathbf{u}^{k_i})\|\|\mathbf{g}^{k_i}_{\mathrm{Ran}(A)}\| \le C\|A(\mathbf{u}_\mu^* - \mathbf{u}^{k_i})\|. \tag{3.11}$$

By Lemma 2.2, $\lim_{k\to\infty} A^T(A\mathbf{u}^k - \mathbf{g}) = 0$. Hence, $\lim_{k\to\infty} A\mathbf{u}^k = \mathbf{g}_{\mathrm{Ran}(A)}$. Similarly, since $A^TA\mathbf{u}_\mu^* = A^T\mathbf{g}$, we have $A\mathbf{u}_\mu^* = \mathbf{g}_{\mathrm{Ran}(A)}$. Therefore, $\lim_{i\to\infty} A\mathbf{u}^{k_i} = A\widetilde{\mathbf{u}} = \mathbf{g}_{\mathrm{Ran}(A)} = A\mathbf{u}_\mu^*$. Hence, $\lim_{i\to\infty}\|A(\mathbf{u}_\mu^* - \mathbf{u}^{k_i})\| = 0$. In view of (3.11), we have $\lim_{i\to\infty}|\langle A(\mathbf{u}_\mu^* - \mathbf{u}^{k_i}), \mathbf{g}^{k_i}_{\mathrm{Ran}(A)}\rangle| = 0$.

Thus, (3.10) gives $H(\widetilde{\mathbf{u}}) = \lim_{i\to\infty} H(\mathbf{u}^{k_i}) \le H(\mathbf{u}_\mu^*)$. This, together with $A^TA\widetilde{\mathbf{u}} = A^T\mathbf{g}$ and the uniqueness of $\mathbf{u}_\mu^*$, implies that $\widetilde{\mathbf{u}} = \mathbf{u}_\mu^*$.

The proof of the second part of the theorem, i.e., the limit of (2.2) tends to the solution of (3.2), is essentially the same as that of the corresponding result of [11] (see the proof of Theorem 4.4 in [11]), and we omit the details here. □

3.2. Deblurring with noninvertible blurring matrices. Now we turn to the deblurring problem (1.1) with noninvertible B. Again, we solve it in the coefficient domain of a tight frame F, where we assume that there is a sparse approximation of the underlying image. Then (1.1) is transferred to (2.14). However, in the case of non-invertible B, the matrix $A = BF$ is no longer surjective. Therefore, there may not necessarily exist a solution of $BF\mathbf{u} = \mathbf{g}$. Hence there may not exist a solution of (2.16) either. This means that we cannot recover the image by solving (2.16). Alternatively, we find a sparse solution of the weighted least square solution of $BF\mathbf{u} = \mathbf{g}$. It reaches the minimization problem

$$\min_{\mathbf{u}\in\mathbb{R}^M}\left\{\|\mathbf{u}\|_1 : \mathbf{u} = \arg\min_{\mathbf{u}\in\mathbb{R}^M}\|BF\mathbf{u} - \mathbf{g}\|_P^2\right\}, \tag{3.12}$$

where $\|\cdot\|_P$ is a weighted seminorm defined by

$$\|\mathbf{a}\|_P = \sqrt{\mathbf{a}^TP\mathbf{a}} \tag{3.13}$$

for a symmetric positive definite matrix P. Once again, we remark that, when B is invertible, (3.12) is equivalent to (2.16).

We use the modified iteration (2.17) to solve (3.12). As before, the matrix P also serves as a preconditioner. Since $A^\dagger = A^T(AA^T)^\dagger = F^TB^T(BB^T)^\dagger$, and it is generally hard to find $(BB^T)^\dagger$, the matrix P is to approximate $(BB^T)^\dagger$, i.e., $P \approx (BB^T)^\dagger$. By applying Theorem 3.1, we have the following theorem.

THEOREM 3.2. *Assume that P is symmetric positive definite and F is a tight frame. Let δ be such that $0 < \delta < \frac{1}{\|B^TPB\|}$. Then the sequence $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ generated by iteration* (2.17) *converges to the unique solution of*

$$\min_{\mathbf{u}\in\mathbb{R}^M}\left\{\mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|\mathbf{u}\|^2 : \mathbf{u} = \arg\min_{\mathbf{u}\in\mathbb{R}^M}\|BF\mathbf{u} - \mathbf{g}\|_P^2\right\}. \tag{3.14}$$

Furthermore, as $\mu \to \infty$, the limit of (2.17) *is the solution of* (3.12) *that has a minimal ℓ_2 norm among all the solutions of* (3.12).

Proof. Following the proof of Theorem 2.4, iteration (2.17) is the linearized Bregman iteration (2.2) for the system of linear equations

$$P^{1/2}BF\mathbf{u} = P^{1/2}\mathbf{g}.$$

Since $\|(P^{1/2}BF)(P^{1/2}BF)^T\| = \|P^{1/2}BB^TP^{1/2}\| = \|B^TPB\|$, $0 < \delta < \frac{1}{\|B^TPB\|}$ implies $0 < \delta < \frac{1}{\|(P^{1/2}BF)(P^{1/2}BF)^T\|}$. Therefore, by Theorem 3.1, with $0 < \delta < \frac{1}{\|B^TPB\|}$, $\{\mathbf{u}^k\}_{k\in\mathbb{N}}$ converges to the solution of

$$\min_{\mathbf{u}\in\mathbb{R}^M}\left\{\mu\|\mathbf{u}\|_1 + \frac{1}{2\delta}\|\mathbf{u}\|^2 : \mathbf{u} = \arg\min_{\mathbf{u}\in\mathbb{R}^M}\|P^{1/2}BF\mathbf{u} - P^{1/2}\mathbf{g}\|^2\right\}. \tag{3.15}$$

By the definition of the weighted ℓ^2 norm in (3.13), (3.15) is equivalent to (3.14). □

4. Numerical experiments. In this section, we illustrate the effectiveness of the frame-based deblurring algorithm by experiments. We start with an algorithm based on iteration (2.17), and then we give simulation results. The comparisons with other methods are given at the end.

4.1. Algorithms. Our objective here is to design a simple and relatively fast tight frame-based algorithm that is, hopefully, also efficient in deblurring. The first algorithm is essentially based on (2.17).

ALGORITHM 1 (Frame-Based Deblurring Algorithm).

1. *Set* $\mathbf{u}^0 = 0$ *and* $\mathbf{g}^0 = 0$.
2. *Iterate*

$$\begin{cases}\mathbf{g}^{k+1} = \mathbf{g}^k + (\mathbf{g} - BF\mathbf{u}^k),\\ \mathbf{u}^{k+1} = \delta T_\mu(F^TB^TP\mathbf{g}^{k+1}),\end{cases} \tag{4.1}$$

until

$$\|\mathbf{g} - BF\mathbf{u}^k\|^2 \le \sigma^2, \tag{4.2}$$

where σ^2 is the estimated variance of the noise.

3. *Set $\widetilde{\mathbf{f}} = F\mathbf{u}^k$ as the deblurred image.*

As has been discussed, this algorithm converges whether the blurring matrix B is invertible or not.

To make everything simple, we use the simplest tight frame F system generated from the filters of the piecewise linear B-spline framelet derived in [46] by applying the unitary extension principle of [46] in numerical simulations and comparisons. The three filters are

$$\mathbf{h}_0 = \frac{1}{4}[1,2,1], \quad \mathbf{h}_1 = \frac{\sqrt{2}}{4}[-1,0,1], \quad \mathbf{h}_2 = \frac{1}{4}[-1,2,-1].$$

There is a standard way to generate an $N \times M$ matrix F whose columns form a tight frame corresponding to a set of given framelet filters. Furthermore, the matrix F is the synthesis (or reconstruction) operator and F^T is the analysis (or decomposition) operator of the underlying tight framelet system. We omit the details here, since the details can be found in [8]. We finally remark that one may choose other tight

framelets constructed in [26, 46], but our experiments show that the simple filters as given above from piecewise linear spline tight framelets are already good enough to illustrate the effectiveness of the frame-based deblurring algorithms. This fact, by itself, already shows the power of the algorithm proposed here.

Throughout the experiments, the level of the framelet decomposition is 4. The preconditioner is chosen as $P = (CC^T + \theta GG^T)^{-1}$, where C is either a circulant matrix for an asymmetric kernel or a DCT-based preconditioner matrix for a symmetric kernel (see [19, 36]), G is the first order difference matrix, and θ is a parameter depending on the image, the noise level, and the algorithm.

As shown in the numerical simulations, this algorithm restores the main features of images first, and it adds details from the residues as the iteration continues. This leads to the question of the stopping criteria. As iteration (4.1) converges, the noise level in the residues becomes relatively high. Hence, while continuing the iteration adds the details, noise also comes in. When the iteration is stopped early, the features are not restored well; when it is stopped late, the artifacts caused by the noise come out. One has to carefully balance getting more small features and making the artifacts at a low level. In Algorithm 1, a natural criteria (4.2) is used. But it still depends on the exactness of the noise level estimation which can be tedious sometimes. In the spirit of the simplicity of our algorithm, we build in a simple postprocessing procedure to remove possible artifacts from noise instead of carefully estimating the stopping point of the iteration. We use a bilateral filter [48] as a post processing procedure to remove possible artifacts. The bilateral filter is chosen, because it is edge-preserving, simple to implement, and efficient at removing the artifacts generated by noise. Furthermore, the bilateral filters have the same nonlocal flavor as those approaches in [38], whose numerical results will be compared here.

Recall that given an image $\mathbf{x}$, the bilateral filter outputs the filtered image $\mathbf{y} = L(\mathbf{x})$ defined by its entry as

$$\mathbf{y}[i,j] = \frac{1}{\mathbf{w}[i,j]} \sum_{p,q} G_{\sigma_s}(\sqrt{|i-p|^2 + |j-q|^2}) G_{\sigma_r}(|\mathbf{f}[i,j] - \mathbf{f}[p,q]|)\mathbf{f}[p,q], \qquad (4.3)$$

where G_{σ_s} and G_{σ_r} are the Gaussian functions with variance σ_s and σ_r respectively, and

$$\mathbf{w}[i,j] = \sum_{p,q} G_{\sigma_s}(\sqrt{|i-p|^2 + |j-q|^2}) G_{\sigma_r}(|\mathbf{f}[i,j] - \mathbf{f}[p,q]|).$$

In fact, the code of [45] provided in [21] for a bilateral filter, which we use in the postprocessing, is indeed very efficient. It takes less than half a second in one run. Altogether, we have the following algorithm.

ALGORITHM 2 (Frame-Based Deblurring Algorithm + Bilateral Filter).

1. *Set* $\mathbf{u}^0 = 0$ *and* $\mathbf{g}^0 = 0$.
2. *Iterate as* (4.1) *until* (4.2) *is satisfied.*
3. *Set* $\widetilde{\mathbf{f}} = L(F\mathbf{u}^k)$, *where* L *is the bilateral filter operator defined in* (4.3), *as the deblurred image.*

4.2. Simulations. We first demonstrate the performance of Algorithms 1 and 2. Figures 4.1, 4.2, and 4.3 give the results of the noisy blurred image and deblurred image by Algorithms 1 and 2. The blurring kernels are a 7×7 disk kernel, a 9×15 motion kernel, and a 15×15 Gaussian kernel with $\sigma_b = 2$, respectively. As shown in the figures, both Algorithms 1 and 2 produce good results. They are effective, since

(a) Noise $\sigma = 2$ (b) 11 iters, 27.6dB (c) 9 iters, 28.3dB

(d) Noise $\sigma = 5$ (e) 6 iters, 25.5dB (f) 5 iters, 25.9dB

(g) Noise $\sigma = 10$ (h) 6 iters, 24.3dB (i) 4 iters, 24.6dB

Noisy Blurred Image Algorithm 1 Algorithm 2

FIG. 4.1. *Deblurring results of Algorithms 1 and 2 for* 256×256 *cameraman image convolved by a* 7×7 *disk kernel generated by the MATLAB command* `fspecial('disk',3)` *and contaminated by a Gaussian white noise of variance* σ^2.

the deblurred image is given in a few steps of iterations. Note that at each iteration, only two fast transforms (FFTs or DCTs), two convolutions, and one tight frame decomposition and reconstruction are used. It is also shown that both algorithms are robust to noise; e.g., when the noise is as high as $\sigma = 10$, it still gives good restorations. By comparing these two algorithms, it is clear that Algorithm 2 performs better than Algorithm 1 as the restored images by Algorithm 2 have higher peak signal to noise ratios (PSNRs) than those by Algorithm 1. Moreover, there are fewer artifacts in the images as expected by Algorithm 2. Recall that the PSNR is defined by $20 \log_{10} \frac{255 \times N}{\|\mathbf{f} - \widetilde{\mathbf{f}}\|_2}$ with $\mathbf{f}$ and $\widetilde{\mathbf{f}}$ being the original and restored images, respectively.

In the iteration in Algorithms 1 and 2, small entries of the frame coefficients are thresholded by a very large thresholding. This leads to only big enough entries coming out in each step. Hence, it is expected that edges will come out first and stay as the iteration continues in Algorithms 1 and 2; i.e., the main features of images are restored first, and details are added in as the iteration continues. This is exactly shown in Figure 4.4, where intermediate iteration steps of the iteration in Algorithm 1 are plotted. We see that edges come first and stay as the iteration goes on, and

(a) Noise $\sigma = 2$ (b) 31 iters, 26.7dB (c) 31 iters, 26.9dB

(d) Noise $\sigma = 5$ (e) 12 iters, 24.5dB (f) 14 iters, 25.2dB

(g) Noise $\sigma = 10$ (h) 6 iters, 23.2dB (i) 6 iters, 23.6dB

Noisy Blurred Image Algorithm 1 Algorithm 2

FIG. 4.2. *Deblurring results of Algorithms 1 and 2 for* 256×256 *cameraman image convolved by a* 9×15 *motion kernel generated by the MATLAB command* `fspecial('motion',15,30)` *and contaminated by a Gaussian white noise of variance* σ^2.

the details of the underlying image are added in as the iteration runs. Therefore, we can stop the iteration whenever enough details are recovered. As the main objective in image deblurring is recovering edges, this shows that the iteration in Algorithm 1 (hence Algorithm 2) takes this as the first priority. As a side product, Algorithm 1 (hence Algorithm 2) can be robust to noise. When the noise level is low, more details can be obtained by iterating more steps. When the noise level is high, one can stop the iteration early, while it still restores edges without much loss. This explains why the number of steps of the iteration decreases as the noise level increases in our experiments.

Next, we apply our methods to a real image in Figure 4.5. The blurring kernel is estimated from two different images taken by a shaking camera on the same scene by the method in [9]. With the estimated blurring kernel, Algorithm 1 is applied. Though there are errors in the estimated kernel, Algorithm 1 gives a very good result for the real image. Moreover, the numbers of iterations are very small: it takes only 14, 16, and 17 steps of the iterations for the three channels of the color image respectively. Since the recovered images are already very good, the postprocessing is not needed. Hence Algorithm 2 is not required in this case.

(a) Noise $\sigma = 2$ (b) 12 iters, 25.4dB (c) 11 iters, 25.6dB

(d) Noise $\sigma = 5$ (e) 6 iters, 24.6dB (f) 6 iters, 24.8dB

(g) Noise $\sigma = 10$ (h) 5 iters, 23.8dB (i) 5 iters, 24.1dB

Noisy Blurred Image Algorithm 1 Algorithm 2

FIG. 4.3. *Deblurring results of Algorithms 1 and 2 for* 256×256 *cameraman image convolved by a* 15×15 *Gaussian kernel of* $\sigma_b = 2$ *generated by the MATLAB command* `fspecial('Gaussian',15,2)` *and contaminated by a Gaussian white noise of variance* σ^2.

4.3. Comparison with Wavelet-Based Methods. In this subsection, we compare Algorithms 1 and 2 with other wavelet-based algorithms. The compared algorithms seek sparse approximations of images under frame systems for linear inverse problems. Rather than solving constrained minimization (2.16) or (3.12) with a proper stopping criteria as done in this paper, the algorithms to be compared solve Lagrangian relaxed minimization problems, i.e., minimize energies consisting of a data fidelity term and a regularization term.

The first algorithm to which we compare our method is the iterative thresholding method in [27], which extends the results in [25] from orthonormal systems to frames. It solves

$$\min_{\mathbf{u}\in\mathbb{R}^N} \frac{1}{2}\|BF\mathbf{u} - \mathbf{g}\|_2^2 + \lambda\|\mathbf{u}\|_1$$

by the iteration

$$\mathbf{u}^{k+1} = T_\lambda(\mathbf{u}^k + F^T B^T(\mathbf{g} - BF\mathbf{u}^k)). \tag{4.4}$$

We compare this iteration with ours in Figure 4.7. The parameter λ is chosen such that the error $\|BF\widetilde{\mathbf{u}} - \mathbf{g}\|_2^2 \approx \sigma^2$, where $\widetilde{\mathbf{u}}$ is the limit of (4.4) and σ^2 is the variance

FIG. 4.4. *Intermediate iteration steps of the iteration in Algorithm 1 for the image deblurring with noise $\sigma = 2$. The top row contains the 2nd, 5th, and 8th iteration steps for the restored image in Figure 4.1. The middle row contains the 5th, 10th, and 15th iteration steps for the restored image in Figure 4.2. The bottom row contains the 2th, 5th, and 8th iteration steps for the restored image in Figure 4.3.*

of the noise. When $\|\mathbf{u}^{k+1} - \mathbf{u}^k\|_2/\|\mathbf{g}\|_2 \leq 10^{-5}$, the iteration is stopped. It takes thousands of iterations for the convergence of (4.4), while Algorithms 1 and 2 are stopped in, generally, tens of iterations. The performances of Algorithms 1 and 2 are also better than those of (4.4) in terms of PSNRs of the deblurred images.

Some intermediate steps of the iteration in Algorithm 1, and iteration (4.4), are shown in Figure 4.8. It shows that edges at earlier steps are already sharp by Algorithm 1, and the iteration adds the details in as it continues. However, for (4.4), edges at earlier steps are blurred, and edges become sharp until enough steps of iteration are performed. Therefore, Algorithm 1 gains more flexibility in the sense that one can stop it whenever enough features are obtained. The same is also true for Algorithm 2, since it is essentially Algorithm 1 followed by a bilateral filter.

Another frame based algorithm that we compare ours with is the deblurring algorithm in [7,12,16–18], where the deconvolution problem is converted to an inpainting problem in a frame domain by designing a tight wavelet frame system via the unitary extension principle of [46], with the convolution kernel being one of the masks of the tight frames. For this, we design a tight wavelet frame system from the convolution

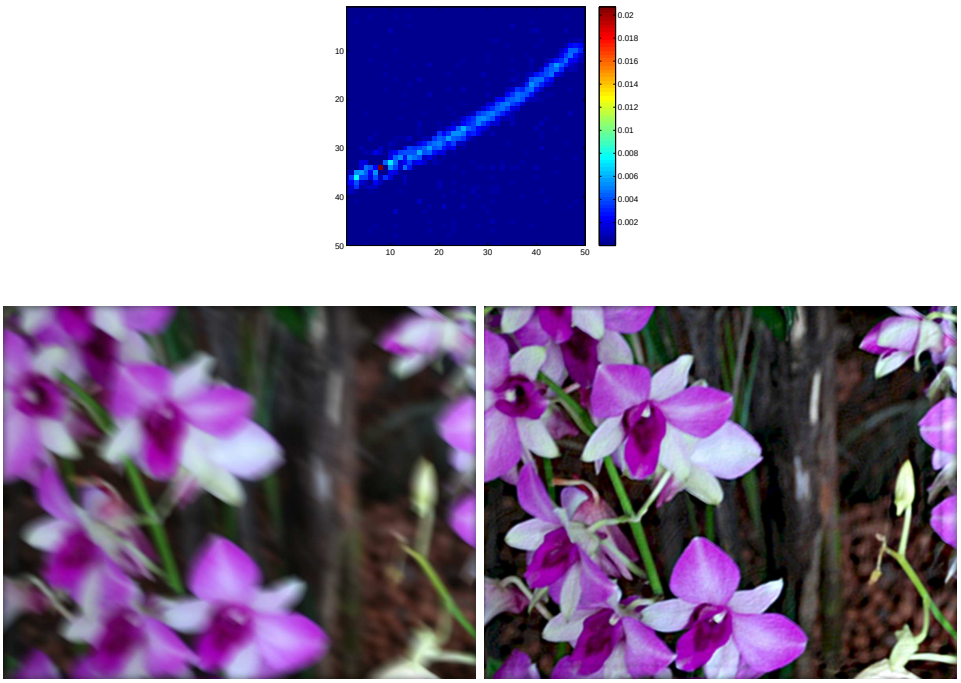


FIG. 4.5. *The results of Algorithm 1 applied to real images. The top row is the estimated blurring kernel from the bottom left blurred image by the method in [9]. On the bottom row, the left is a real image taken from a shaking camera, and the right is the restored image by Algorithm 1. The numbers of iterations are 14, 16, and 17 for the RGB channels, respectively.*

kernel. For the comparison, we choose the convolution kernel to be

$$\mathbf{h}_0 = \frac{1}{4}\left[\frac{1}{2}, 1, 1, 1, \frac{1}{2}\right], \tag{4.5}$$

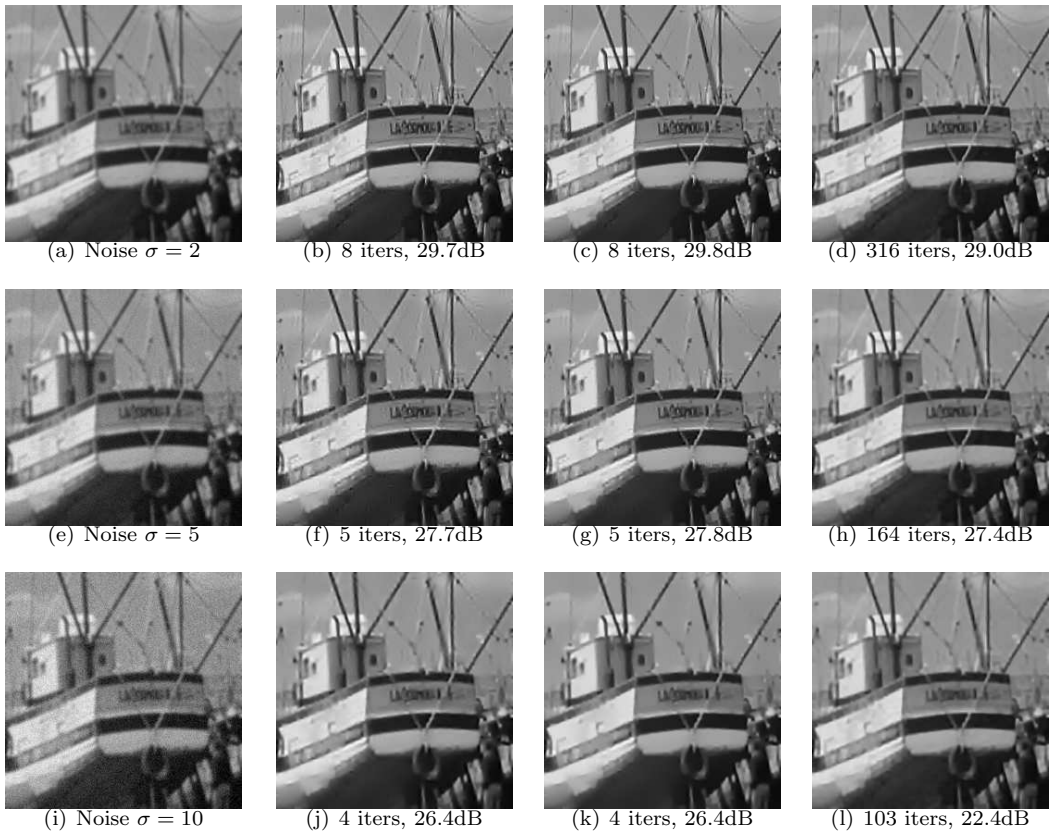
which arises in high-resolution image restoration [17, 18]. From this kernel, we construct a framelet system with the low-pass filters being $\mathbf{h}_0$ by the unitary extension principle [46]. Then a tight frame F is constructed by the method described in [8]. The iteration is

$$\mathbf{f}^{k+1} = FT_\lambda F^T\left(H_0^T\mathbf{g} + \sum_{i=1}^{r} H_i^T H_i \mathbf{f}^k\right), \tag{4.6}$$

where H_i is the filtering matrix for the filter $\mathbf{h}_i$; see [8]. It was proven in [12, 16] that the tight frame coefficient $\mathbf{u}^k := T_\lambda F^T(H_0^T\mathbf{g} + \sum_{i=1}^{r} H_i^T H_i \mathbf{f}^k)$ converges to a solution of

$$\min_{\mathbf{u}\in\mathbb{R}^N} \frac{1}{2}\|BF\mathbf{u} - \mathbf{g}\|_2^2 + \|(I - F^T F)\mathbf{u}\|_2^2 + \lambda\|\mathbf{u}\|_1.$$

We show the comparison results in Figure 4.6. Algorithms 1 and 2 take far fewer iterations, and the restored images by Algorithms 1 and 2 are slightly better than those by (4.6) in terms of PSNRs.



(a) Noise $\sigma = 2$ (b) 8 iters, 29.7dB (c) 8 iters, 29.8dB (d) 316 iters, 29.0dB

(e) Noise $\sigma = 5$ (f) 5 iters, 27.7dB (g) 5 iters, 27.8dB (h) 164 iters, 27.4dB

(i) Noise $\sigma = 10$ (j) 4 iters, 26.4dB (k) 4 iters, 26.4dB (l) 103 iters, 22.4dB

Noisy Blurred Image | Algorithm 1 | Algorithm 2 | Iteration (4.6)

FIG. 4.6. *Comparison of Algorithms 1 and 2 with iteration* (4.6).

We also show the intermediate iteration steps of the iteration in Algorithm 1, and iteration (4.6), in Figure 4.9. Again, Algorithm 1 restores and keeps edges at earlier steps, while details and small features are recovered step by step. On the other hand, iteration (4.6) does not get sharp edges at earlier steps. It takes a lot of steps of the iteration to restore sharp edges.

4.4. Comparisons with Nonlocal and Other Variational Methods. The second set of algorithms which we compare ours with is the most recent deblurring method based on nonlocal operators together with other variational methods based on PDE approach. In particular, we compare Algorithms 1 and 2 with the deblurring algorithms [38] using nonlocal operators. As a consequence, it also leads to comparisons with other variational algorithms as they are already compared in [38] that include nonlocal means deblurring [6] and TV (ROF model) [47]. The settings used here are exactly the same as those in [38]. The blurring kernels are Gaussians and a box average, which are considered to be hard. Examples of restored images by Algorithms 1 and 2 and the SNR values are shown in Figure 4.10.

Following [38], instead of using the PSNR as before, we use the signal to noise ratio (SNR) to measure the quality of the restored images. Recall that the SNR is defined by $20\log_{10}\frac{\|\mathbf{f}-\bar{\mathbf{f}}\|_2}{\|\mathbf{f}-\widetilde{\mathbf{f}}\|_2}$ with $\mathbf{f}$, $\bar{\mathbf{f}}$, and $\widetilde{\mathbf{f}}$ being the original image, its mean values,

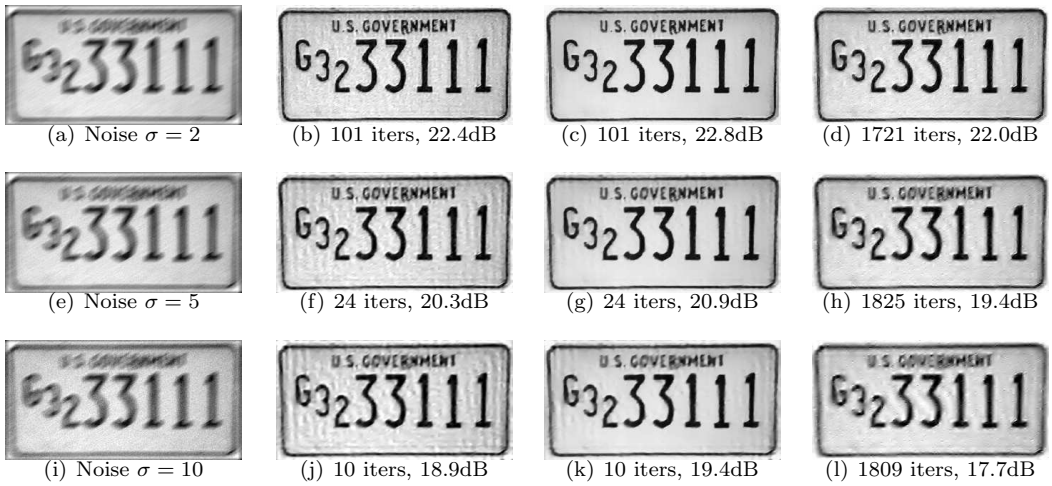


FIG. 4.7. *Comparison of Algorithms 1 and 2 with algorithm* (4.4) *proposed in [27]. The blurring kernel is a motion blur generated by the MATLAB command* `fspecial('motion',15,30)`.

FIG. 4.8. *Comparison of intermediate iteration steps of the iteration in Algorithm 1 with iteration* (4.4) *for the image deblurring in Fig 4.7 with* $\sigma = 2$. *The top row shows the 10th, 20th, and 50th iteration steps of the iteration in Algorithm 1. The bottom row shows the 10th, 100th, and 1000th iteration steps of* (4.4).

and the restored images, respectively. The SNR results are summarized in Table 4.1, where the SNR values are averages of 10 runs. The corresponding deblurring results by other PDE-based algorithms for the same blurring kernel and the same level of noise are shown in Table 4.2, which is provided by Table II in [38]. These tables show that both Algorithms 1 and 2 perform better in terms of SNR than the ROF model in [47] and are comparable to the most advanced nonlocal algorithms.

In summary, while our algorithm is simple and fast, its performance is comparable to the benchmark algorithms known in the literature. The keys to make the algorithms work are: (a) the linearized Bregman iteration, which restores the main features of images first and adds details as the iteration continues, (b) the tight frames, whose redundancy makes the sparse approximation more robust, and (c) the preconditioning techniques, which accelerate the convergence. We further remark that here we choose only the simplest redundant tight framelet system. It can be expected that performance can be even better when more adaptive frame systems are used. Since our aim here is to make the implementation simple, we forgo further tune-ups of the

FIG. 4.9. *Comparison of intermediate iteration steps of the iteration in Algorithm 1 with iteration* (4.6) *for the image deblurring in Fig 4.6 with* $\sigma = 2$. *The top row shows the 2nd, 4th, and 6th iteration steps of the iteration in Algorithm 1. The bottom row shows the 5th, 50th, and 150th iteration steps of* (4.6).

Image	Shape	Barbara	Cameraman
Blur kernel	Gaussian $\sigma_b = 2$	Gaussian $\sigma_b = 1$	9×9 box average
noise	$\sigma = 10$	$\sigma = 5$	$\sigma = 3$
Algorithm 1	16.19	12.10	13.25
Algorithm 2	18.45	12.11	13.58

TABLE 4.1
SNR results in dB of Algorithms 1 and 2. The results are averages of 10 runs.

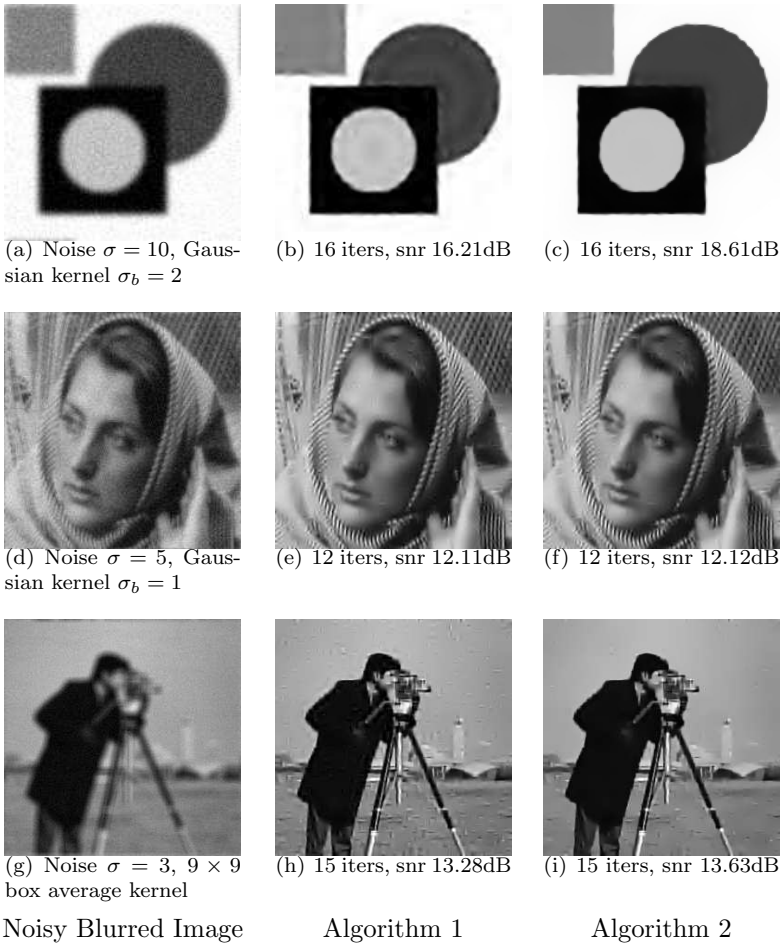
implementations. However, this does not prevent us from foreseeing the potential of the algorithms provided here.

4.5. Split Bregman iteration deblurring. Instead of finding a sparsest frame coefficient solution of the deblurring problem in the frame domain as in (2.15) and (4.4), one may find the sparsest canonical frame coefficient solution. This amounts to solving the following problem:

$$\min_{\mathbf{f}\in\mathbb{R}^M} \frac{1}{2}\|B\mathbf{f} - \mathbf{g}\|_2^2 + \mu\|F^T\mathbf{f}\|_1. \tag{4.7}$$

Since the weighted norm of canonical frame coefficients links the function norm of the underlying image, it gives a smoother solution. The hard part is that the thresholding cannot keep the resulting frame coefficients canonical. It turns out that the split Bregman iteration in [33] is very handy here. The basic idea is to transfer the unconstrained minimization (4.7) to a constrained one as follows:

$$\begin{cases} \min\limits_{\mathbf{f}\in\mathbb{R}^M} \frac{1}{2}\|B\mathbf{f} - \mathbf{g}\|_2^2 + \mu\|\mathbf{u}\|_1 \\ \text{s.t. } \mathbf{u} = F^T\mathbf{f}. \end{cases} \tag{4.8}$$



(a) Noise $\sigma = 10$, Gaussian kernel $\sigma_b = 2$ (b) 16 iters, snr 16.21dB (c) 16 iters, snr 18.61dB

(d) Noise $\sigma = 5$, Gaussian kernel $\sigma_b = 1$ (e) 12 iters, snr 12.11dB (f) 12 iters, snr 12.12dB

(g) Noise $\sigma = 3$, 9×9 box average kernel (h) 15 iters, snr 13.28dB (i) 15 iters, snr 13.63dB

Noisy Blurred Image Algorithm 1 Algorithm 2

FIG. 4.10. *The restored shape, Barbara, and cameraman images (from top to bottom) by Algorithms 1 and 2. The settings are the same as those in [38].*

Then, an inexact Bregman iteration algorithm is applied. This is the splitting Bregman algorithm. It generates a sequence by, starting from $\mathbf{f}^0 = 0$, $\mathbf{u}^0 = 0$, and $\mathbf{b}^0 = 0$,

$$\begin{cases} \mathbf{f}^{k+1} = \arg\min\limits_{\mathbf{f}\in\mathbb{R}^M} \dfrac{\lambda}{2}\|F^T\mathbf{f} - (\mathbf{u}^k - \mathbf{b}^k)\|_2^2 + \dfrac{1}{2}\|B\mathbf{f} - \mathbf{g}\|_2^2, \\ \mathbf{u}^{k+1} = \arg\min\limits_{\mathbf{u}\in\mathbb{R}^N} \dfrac{\lambda}{2}\|\mathbf{u} - (F^T\mathbf{f}^{k+1} + \mathbf{b}^k)\|_2^2 + \mu\|\mathbf{u}\|_1, \\ \mathbf{b}^{k+1} = \mathbf{b}^k + (F^T\mathbf{f}^{k+1} - \mathbf{u}^{k+1}). \end{cases} \tag{4.9}$$

In each iteration, the first two steps can be repeated several times before the third step is performed. Notice that the second step is cheap to compute, since it is reduced to a soft-thresholding. However, it is hard to find the solution of the first step, because it is equivalent to solving the system of linear equations

$$\mathbf{f}^{k+1} = (\lambda I + B^T B)^{-1}[\lambda F(\mathbf{u}^k - \mathbf{b}^k) + B^T\mathbf{g}].$$

Since $B^T B$ is a convolution, we can use circulant preconditioned conjugate gradient (PCG) to solve it efficiently; see [19]. We compare the linearized Bregman deblurring

Image	Shape	Barbara	Cameraman
Blur kernel	Gaussian $\sigma_b = 2$	Gaussian $\sigma_b = 1$	9×9 box average
noise	$\sigma = 10$	$\sigma = 5$	$\sigma = 3$
Wiener filter [1]	13.20	10.57	11.29
ROF [47]	14.22	10.71	12.34
Nonlocal means deblur (NL-means) [6]	16.19	12.09	12.43
Nonlocal total variation (NLTV) [38]	17.53	11.95	12.75
Wiener + NLTV [38]	20.19	12.45	13.50

TABLE 4.2
SNR results in dB of other algorithms. The results are copied from Table II in [38].

(a) Noise $\sigma = 3$, 9×9 box average kernel (b) 15 iters, 25.85dB (c) 34 iters, 25.58dB

Noisy Blurred Image Algorithm 2 Split Bregman (4.9)

FIG. 4.11. *Comparison of Algorithm 2 and the split Bregman iteration* (4.9).

(Algorithm 2) with the split Bregman iteration (4.9). For simplicity, we use the circular boundary condition in B such that the first step in (4.9) can be solved by two FFTs. As pointed out in [33], the algorithm can still give a good result even when the constraint in (4.8) is satisfied with a low accuracy. Therefore, we stop the split Bregman iteration early when $\|F^T\mathbf{f}^k - \mathbf{u}^k\|_2/\|\mathbf{g}\|_2 < 10^{-3}$. The comparison results are shown in Figure 4.11. Algorithm 2 and iteration (4.9) produce comparable results, although Algorithm 2 takes fewer steps to converge. Since the splitting Bregman iteration and its corresponding model (4.7) are not the focus of this paper, we give only a brief discussion here. A comprehensive study of model (4.7) and the analysis of the splitting Bregman iteration (4.9) will be given in a forthcoming paper.

REFERENCES

[1] H. C. ANDREWS AND B. R. HUNT, *Digital image restoration*, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[2] G. AUBERT AND P. KORNPROBST, *Mathematical problems in image processing*, 2nd ed., Appl. Math. Sci. 147, Springer-Verlag, New York, second ed., 2006. Partial differential equations and the calculus of variations.

[3] L. BORUP, R. GRIBONVAL, AND M. NIELSEN, *Bi-framelet systems with few vanishing moments characterize Besov spaces*, Appl. Comput. Harmon. Anal., 17 (2004), pp. 3–28.

[4] K. BREDIES AND D. A. LORENZ, *Iterated hard shrinkage for minimization problems with sparsity constraints*, SIAM J. Sci. Comput., 30 (2008), pp. 657–683.

[5] A. BUADES, B. COLL, AND J. M. MOREL, *A review of image denoising algorithms, with a new one*, Multiscale Model. Simul., 4 (2005), pp. 490–530 (electronic).

[6] A. BUADES, B. COLL, AND J.M. MOREL, *Image enhancement by non-local reverse heat equation*, 2006. CMLA Tech. Rep. 22.

[7] J.-F. CAI, R. CHAN, L. SHEN, AND Z. SHEN, *Restoration of chopped and nodded images by*

framelets, SIAM J. Sci. Comput., 30 (2008), pp. 1205–1227.
[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] J.-F. Cai, H. Ji, and C. Liu, *Motion deblur of multiple images*, 2008. preprint.
[10] J.-F. Cai, S. Osher, and Z. Shen, *Convergence of the Linearized Bregman Iteration for ℓ_1-norm Minimization*, 2008. Math. Comp., to appear; see also UCLA CAM Report (08-52).
[11] J.-F. Cai, S. Osher, and Z. Shen, *Linearized Bregman Iterations for Compressed Sensing*, 2008. Math. Comp., to appear; see also UCLA CAM Report (08-06).
[12] J.-F. Cai and Z. Shen, *Deconvolution: A wavelet frame approach, II*, 2008. preprint.
[13] E. J. Candès and D. L. Donoho, *New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities*, Comm. Pure Appl. Math., 57 (2004), pp. 219–266.
[14] E. Cetin, *Reconstruction of signals from Fourier transform samples*, Signal Process., 16 (1989), pp. 129–148.
[15] E. Cetin, *An iterative algorithm for signal reconstruction from bispectrum*, IEEE Trans. Signal Process., 39 (1991), pp. 2621–2628.
[16] A. Chai and Z. Shen, *Deconvolution: A wavelet frame approach*, Numer. Math., 106 (2007), pp. 529–587.
[17] R. H. Chan, T. F. Chan, L. Shen, and Z. Shen, *Wavelet algorithms for high-resolution image reconstruction*, SIAM J. Sci. Comput., 24 (2003), pp. 1408–1432 (electronic).
[18] R. H. Chan, S. D. Riemenschneider, L. Shen, and Z. Shen, *Tight frame: an efficient way for high-resolution image reconstruction*, Appl. Comput. Harmon. Anal., 17 (2004), pp. 91–115.
[19] R. H.-F. Chan and X.-Q. Jin, *An Introduction to Iterative Toeplitz Solvers*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007.
[20] T. F. Chan and J. Shen, *Image processing and analysis*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005. Variational, PDE, wavelet, and stochastic methods.
[21] J. Chen, *http://people.csail.mit.edu/jiawen/#code*.
[22] R. R. Cofiman and D. L. Donoho, *Translation-invariant de-noising*, in Wavelets and Statist., A. Antoniadis and G. Oppenheim, eds., vol. 103 of Lecture Notes in Statistics, New York, 1995, Springer-Verlag, pp. 125–150.
[23] J. Darbon and S. Osher, *Fast discrete optimization for sparse approximations and deconvolutions*, 2007. preprint.
[24] I. Daubechies, *Ten lectures on wavelets*, vol. 61 of CBMS-NSF Regional Conf. Ser. Appl. Math., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.
[25] I. Daubechies, M. Defrise, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. Pure Appl. Math., 57 (2004), pp. 1413–1457.
[26] I. Daubechies, B. Han, A. Ron, and Z. Shen, *Framelets: MRA-based constructions of wavelet frames*, Appl. Comput. Harmon. Anal., 14 (2003), pp. 1–46.
[27] I. Daubechies, G. Teschke, and L. Vese, *Iteratively solving linear inverse problems under general convex constraints*, Inverse Probl. Imaging, 1 (2007), pp. 29–46.
[28] D. C. Dobson and F. Santosa, *Recovery of blocky images from noisy and blurred data*, SIAM J. Appl. Math., 56 (1996), pp. 1181–1198.
[29] D. L. Donoho, *De-noising by soft-thresholding*, IEEE Trans. Inform. Theory, 41 (1995), pp. 613–627.
[30] S. Durand and M. Nikolova *Denoising of frame coefficients using ℓ^1 data-fidelity term and edge-preserving regularization*, Multiscale Model. Simul., 6 (2007), pp. 547–576.
[31] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, *Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)*, Appl. Comput. Harmon. Anal., 19 (2005), pp. 340–358.
[32] G. Gilboa and S. Osher, *Nonlocal linear image regularization and supervised segmentation*, Multiscale Model. Simul., 6 (2007), pp. 595–630 (electronic).
[33] T. Goldstein and S. Osher, *The Split Bregman Algorithm for L1 Regularized Problems*, SIAM J. Imaging Sci., to appear.
[34] B. Han and Z. Shen, *Dual wavelet frames and Riesz bases in Sobolev spaces*, Constr. Approx., to appear.
[35] L. He, A. Marquina, and S. J. Osher, *Blind deconvolution using TV regularization and Bregman iteration*, Int. J. Imaging Syst. Technol., 15 (2005), pp. 74–83.
[36] X.-Q. Jin, *Developments and applications of block Toeplitz iterative solvers*, vol. 2 of Combinatorics and Computer Science, Kluwer Academic Publishers Group, Dordrecht, The Netherlands, 2002.

[37] M. Li, B. Hao, and X. Feng, *Iterative regularization and nonlinear inverse scale space based on translation invariant wavelet shrinkage*, Int. J. Wavelets Multiresolut. Inf. Process., 6 (2008), pp. 83–95.
[38] Y. Lou, X. Zhang, S. Osher, and A. Bertozzi, *Image recovery via nonlocal operators*, 2008. UCLA CAM Report (08-35).
[39] S. Mallat, *A wavelet tour of signal processing*, Academic Press Inc., San Diego, 1998.
[40] A. Marquina, *Inverse scale space methods for blind deconvolution*, 2006. UCLA CAM Report (06-36).
[41] A. Marquina and S. J. Osher, *Image super-resolution by TV-regularization and Bregman Iteration*, J. Sci. Comput., 37 (2008), pp. 367–382.
[42] M. Nikolova, *Local strong homogeneity of a regularized estimator*, SIAM J. Appl. Math., 61 (2000), pp. 633–658 (electronic).
[43] 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 (electronic).
[44] S. Osher, Y. Mao, B. Dong, and W. Yin, *Fast Linearized Bregman Iteration for Compressed Sensing and Sparse Denoising*, 2008. UCLA CAM Reprots (08-37).
[45] S. Paris, and F. Durand, *A fast approximation of the bilateral filter using a signal processing approach*, Proceedings of ECCV, 2006, pp. 568–580.
[46] A. Ron and Z. Shen, *Affine systems in $L_2(\mathbf{R}^d)$: the analysis of the analysis operator*, J. Funct. Anal., 148 (1997), pp. 408–447.
[47] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[48] C. Tomasi, and R. Manduchi, *Bilateral Filtering for Gray and Color Images*, Proceedings of the 1998 IEEE International Conference on Computer Vision, Bombay, India.
[49] A. N. Tikhonov and V. Y. Arsenin, *Solutions of ill-posed problems*, V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York, 1977. Scripta Series in Mathematics.
[50] J. Xu and S. J. Osher, *Iterative regularization and nonlinear inverse scale space applied to wavelet-based denoising*, IEEE Trans. Image Process., 16 (2007), pp. 534–544.
[51] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168.