

## RESTORATION OF CHOPPED AND NODDED IMAGES BY FRAMELETS

JIAN-FENG CAI\*, RAYMOND CHAN†, LIXIN SHEN‡, AND ZUOWEI SHEN§

**Abstract.** In infrared astronomy, an observed image from a chop and nod process can be considered as the result of passing the original image through a highpass filter. Here we propose a restoration algorithm which builds up a tight framelet system that has the highpass filter as one of the framelet filters. Our approach reduces the solution of restoration problem to that of recovering the missing coefficients of the original image in the tight framelet decomposition. The framelet approach provides a natural setting to apply various sophisticated framelet denoising schemes to remove the noise without reducing the intensity of major stars in the image. A proof of the convergence of the algorithm based on convex analysis is also provided. Simulated and real images are tested to illustrate the efficiency of our method over the projected Landweber method.

**Key words.** Tight frame, chopped and noded image, projected Landweber method, convex analysis

**AMS subject classifications.** 42C40, 65T60, 68U10, 94A08

**1. Introduction.** We start with a very brief introduction to the formation of chopped and noded images and refer the readers to the papers by Bertero *et al.* in [2]–[5] for details. In ground-based astronomy at mid-infrared wavelengths ( $\lambda \approx 5\text{--}20\mu\text{m}$ ), the weak astronomical signal  $f$  is corrupted by the overwhelming thermal background produced by the atmosphere and the telescope. The observed signal  $s$  from the direction  $(x, y)$  at time  $t$  on the detector plane is the superposition of the weak signal  $f(x, y)$  together with a large time-varying background  $\eta(x, y, t)$  coming from the atmosphere and telescope optics, i.e.,

$$s = f(x, y) + \eta(x, y, t).$$

To extract the celestial source  $f$ , we need to eliminate the effect of the background  $\eta(x, y, t)$ . A common approach called *chop-and-nod* is employed. Chopping refers to the rapid modulation of the telescope beam between the target and an empty sky area. Nodding refers to a second chopping sequence done with the telescope pointing to an offset position. In essence, if the target position is  $(x, y)$  and the two sky areas are  $(x, y + \Delta)$  and  $(x, y - \Delta)$ , the chopping and nodding techniques produce a second difference of the observed signal  $s$

$$-f(x, y - \Delta) + 2f(x, y) - f(x, y + \Delta) + e, \quad (1.1)$$

---

\*Department of Mathematics, the Chinese University of Hong Kong, Shatin, Hong Kong, China. Email: [tslcai@nus.edu.sg](mailto:tslcai@nus.edu.sg). Current address: Temasek Laboratories and Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543.

†Department of Mathematics, the Chinese University of Hong Kong, Shatin, Hong Kong, China. Email: [rchan@math.cuhk.edu.hk](mailto:rchan@math.cuhk.edu.hk). Phone: (852) 2609-7970. Fax: (852) 2603-5154. Research supported in part by HKRGC Grant CUHK 400503 and CUHK DAG 2060257. This work was partially done while this author was visiting the Institute for Mathematical Sciences, National University of Singapore in 2003. The visit was supported by the Institute.

‡(corresponding author) Department of Mathematics, Syracuse University, Syracuse, NY 13244. Email address: [lshen03@syr.edu](mailto:lshen03@syr.edu). Phone: (315) 443-1487. Fax: (315) 443-1475.

§Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543. Email: [matzuows@nus.edu.sg](mailto:matzuows@nus.edu.sg). Phone: (65) 6874-6913. Fax: (65) 6874-5452. Research supported in part by Grant R-146-000-060-112 at the National University of Singapore.

where the quantity  $\Delta$  is called the *chopping throw* or *chopping amplitude* and  $e := -\eta(x, y - \Delta, t') + 2\eta(x, y, t) - \eta(x, y + \Delta, t'')$  with the time  $t'$  and  $t''$  being close to  $t$ . Under suitable assumptions [2], the term  $e$  can be modeled by a white Gaussian process. One gets the so-called *chopped and noded image*:

$$g(x, y) := -f(x, y - \Delta) + 2f(x, y) - f(x, y + \Delta) + e. \quad (1.2)$$

The effect of noise  $e$  to the accuracy of the restoration is discussed in Section 5. To restore  $f$ , we need to solve the inversion problem (1.2) from  $g$ .

In this paper, we only consider the case when the chopping amplitude  $\Delta$  is an integral multiple of the sampling distance in the detector plane, i.e.  $\Delta = K$  is an integer. We remark that this assumption can be restrictive in some situations, see [3, 5, 21]. We can write (1.2) in the following discrete form:

$$g_{j,m} = -f_{j,m-K} + 2f_{j,m} - f_{j,m+K} + e_{j,m} \quad (1.3)$$

where  $g_{j,m}$  and  $f_{j,m}$  are the samples of  $g(x, y)$  and  $f(x, y)$  respectively, and  $e_{j,m}$  is the samples of  $e$  at  $(j, m)$ . For each  $j$ , let  $\mathbf{g}_j$  be the vector whose  $m$ th entry is given by  $g_{j,m}$ ,  $m = 1, 2, \dots, N$  respectively; and  $\mathbf{f}_j$  be the vector with the  $n$ th entry given by  $f_{j,n-K}$ ,  $n = 1, 2, \dots, N + 2K$ . Then (1.3) can be written in a matrix form:

$$\mathbf{g}_j = A\mathbf{f}_j + \mathbf{e}_j \quad (1.4)$$

where the  $(m, n)$ th entry of  $A$  is given by  $A(m, n) = -\delta_{m,n} + 2\delta_{m+K,n} - \delta_{m+2K,n}$  with  $m = 1, 2, \dots, N$ ;  $n = 1, 2, \dots, N + 2K$  and  $\delta_{m,n} = 0$  if  $m \neq n$  and  $\delta_{n,n} = 1$ . The matrix  $A$  is called the *imaging matrix*.

Formulation (1.4) is similar to deconvolution problems except here that  $A$  is a highpass filter instead of being a lowpass filter. One standard approach for deconvolution problems is to find  $\mathbf{f}_j$  such that  $A\mathbf{f}_j \approx \mathbf{g}_j$  (data-fitting) while requiring  $\mathbf{f}_j$  to be smooth in certain sense (regularization). In [4], the *projected Landweber method* is used to find the solution of (1.4). It is defined as follows:

$$\mathbf{f}_j^{(n+1)} = \mathbf{P}_+ \left[ \mathbf{f}_j^{(n)} + \varrho A^T (\mathbf{g}_j - A\mathbf{f}_j^{(n)}) \right], \quad n = 0, 1, \dots \quad (1.5)$$

where  $\mathbf{P}_+$  is the projection operator onto the set of non-negative vectors (since brightness distribution of a celestial source must be non-negative) and  $\varrho$  is a relaxation parameter which satisfies  $0 < \varrho < 2/\lambda_1$  with  $\lambda_1$  being the largest eigenvalue of  $A^T A$ . For a detail discussion of the method, see [1, Chapter 6].

It was pointed out in [1, 4] that the projected Landweber method has a regularization property, known as *semiconvergence*: the iterates  $\mathbf{f}_j^{(n)}$  first approach the true image, but the noise will be amplified when  $n$  is larger than a certain threshold. Thus a stopping criterion, called the *discrepancy principle*, is introduced in order to obtain the best approximation of the required solution. However, due to the special structure of the matrix  $A$ , the restored images always have some kinds of artifacts (see Figure 5.4 (b)).

In this paper, we introduce a tight frame method for solving (1.4). There are many papers on using wavelet methods to solve inverse problems, and in particular, deconvolution problems. One of the main ideas is to construct a wavelet or “wavelet inspired” basis that can almost diagonalize the given operator. The underlying solution has a sparse expansion with respect to the chosen basis. The Wavelet-Vaguelette decomposition proposed in [20, 22] and the deconvolution in mirror wavelet bases in

[28, 29] can both be viewed as examples of this strategy. Another approach is to apply Galerkin-type methods to inverse problems using an appropriate, but fixed wavelet basis (see e.g. [6, 13]). Again, the idea there is that if the given operator has a sparse representation and the solution has a sparse expansion with respect to the wavelet basis, then the inversion is reduced approximately to the inversion of a truncated operator.

Recently, two new iterative thresholding ideas have been proposed in [10, 11, 12] and [17, 24]. Instead of requiring the system to have an almost diagonal representation of the operator, they only require that the underlying solution has a sparse expansion with respect to a given orthonormal but not necessary wavelet basis or to a tight frame system. The main idea of [17] is to expand the current iterate with respect to the chosen orthonormal basis for a given algorithm such as the Landweber method. Then a thresholding algorithm is applied to the coefficients of this expansion. The results are then combined to form the next iterate. The algorithm of [17] is shown to converge to the minimizer of certain cost functional. An essentially same algorithm for inverting convolution operator acting on objects that are sparse in the wavelet domain is given in [24].

The tight frame algorithm that we are going to propose for (1.4) is closer to the approach in [10, 11, 12], where high-resolution image reconstruction or, more generally, deconvolution problem with the convolution kernel being a lowpass filter are considered. An analysis of the convergence and optimal property of these algorithms are given in [10]. In fact, our algorithm in this paper is motivated by the ideas in [10, 11, 12] to convert the deconvolution problem  $\mathbf{g} = \mathbf{A}\mathbf{f}$  into an inpainting problem in the transformed domain. To make use of the given data  $\mathbf{g}$  for inpainting, one needs to construct a tight frame system where the given convolution operator  $A$  corresponds to one of the framelet masks of the system, say  $h$ . Then the convolution equation  $\mathbf{g} = \mathbf{A}\mathbf{f}$  can be viewed as giving us the framelet coefficients of  $\mathbf{f}$  corresponding to the given mask  $h$ . Hence the problem is to find the framelet coefficients of  $\mathbf{f}$  corresponding to the framelet masks other than  $h$ . In short, by choosing one of the framelet masks of the system corresponding to  $A$ , the deconvolution problem is converted to the problem of inpainting in the framelet domain—finding the missing framelet coefficients. This is to be compared with inpainting in the image domain where we are given part of the image and finding the missing part of the image. Here we iteratively regenerate the missing framelet coefficients. The noise are removed by thresholding at each iteration. We will see that this iterative algorithm converges and its limit satisfies certain optimal properties. To make all these work, we need to build up a tight framelet system from the  $A$  given in (1.4). This can be done by using the unitary extension principle of [34]. We remark here that for our  $A$ , it is impossible to construct an orthonormal (or non-redundant) wavelet system with  $A$  corresponds to one of the masks.

We note that since tight frames are redundant systems, so information lost along one framelet direction can still be contained in, and hence, recovered from other framelet directions. In fact, the redundancy not only helps in recovering the missing framelet coefficients, it also helps in reducing artifacts introduced by the thresholding denoising scheme built in the algorithm as pointed out in [14]. We further remark that unlike the approaches in [6, 13, 20, 22, 28, 29], our approach does not attempt to find a tight frame system under which the convolution operator can be sparsely represented. Instead, similar to the approaches in [10, 11, 12, 17, 24], we only require the underlying solution  $f$  to have a sparse representation under the tight frame system we constructed. It is shown in [7, 8] that piecewise smooth functions with a few

spikes do have sparse representations by compactly supported tight frame systems. Hence, implicitly, we assume that  $f$  is piecewise smooth with possibly some spikes. Furthermore, the fact that the limit minimizes a functional of  $\ell_1$  norm of framelet coefficients is desirable in image restoration as have already been observed by many researchers (see e.g. [10]).

Before getting into the technical details of our approach, we summarize the main ideas in the followings:

1. **Designing a tight framelet system:** In view of (1.3), we consider the chopped and noded image  $g$  in (1.2) as the output obtained by passing the true image  $f$  through the highpass filter:

$$(-1, \underbrace{0, \dots, 0}_{K-1}, 2, \underbrace{0, \dots, 0}_{K-1}, -1). \quad (1.6)$$

We first design a tight framelet system from a multiresolution analysis that has this chop-and-nod highpass filter (1.6) as one of the framelet masks. Then  $g$  can be understood as framelet coefficients of  $f$  corresponding to the framelet with (1.6) as the framelet mask. The restoration of  $f$  becomes the problem of recovering the coefficients of other framelets and the coefficients of the coarse level approximations (low frequency information) in the tight frame representation of  $f$ .

2. **Restoring the missing information in  $f$ :** The missing coefficients of  $f$  are found by an iterative procedure. The previous iterate is first decomposed by a tight framelet decomposition. Then the missing coefficients of  $f$  are approximated by the corresponding coefficients in the previous iterate and combined with  $g$  in the tight framelet reconstruction algorithm to obtain a new iterate. The tight framelet decomposition and reconstruction algorithms are based on those given by [18]. We will see that the projected Landweber method (1.5) with  $\varrho = 1/16$  is a simple version of our tight frame algorithm, where *no* noise removal thresholding is done on any framelet coefficients. This observation not only gives a new interpretation of the Landweber method, but it also gives a platform to understand the method in terms of framelet theory and multiresolution analysis.
3. **Denoising by thresholding:** In our tight frame algorithm, the denoising is done by damping the framelet coefficients using a tight framelet denoising scheme. The scheme can denoise part of the components in each iterate, and leave other components intact. It is because our tight frame approach can identify the precise components that needs to be denoised. In contrast, the denoising scheme in [17, 24] is applied to every component of the iterate. As it is shown in the numerical simulation, this step also helps to remove some of the artifacts.

Since we have to restrict the solution onto the set of non-negative vectors due to the physical meaning of  $f$ , the analysis in [10], which is based on framelet analysis, cannot be applied here. In this paper, we give an entirely different approach from [10] to prove the convergence of our framelet algorithm. The analysis uses the framework of proximal forward-backward splitting proposed in [15] constructed under the theory of convex analysis and optimization. It will be shown in our numerical tests that the timing of our tight frame algorithm is higher than those of Landweber type methods (see, e.g., [3, 5]) but is quite manageable. Our method can be used as a good post-processing method to clean up these kinds of chopped and noded infrared images.

The outline of the paper is as follows. In Section 2, we give a brief review on tight frames and design filters used in this paper. In Section 3, we give our iterative algorithm. In Section 4, we prove the convergence of the algorithm. Results on simulated and real images are presented in Section 5.

**2. Tight Framelet Analysis.** The chopped and noded image  $g$  in (1.2) and (1.3) can be viewed as the result of passing the original image  $f$  through a highpass filter given in (1.6). Since we only have the highpass filter available to us, it only can be a framelet mask. We therefore have to design a framelet system and its associated multiresolution analysis with this given highpass filter. Here, we note that the associated multiresolution analysis is important in order to have a fast algorithm for the framelet transform, and the framelet transform is a necessity for any framelet based algorithm. To this end, we will make use of the unitary extension principle for tight frame constructions given in [34]. We also remark that there are no papers that we are aware of on building a wavelet or framelet system and its associated multiresolution analysis when one of the given wavelet or framelet masks is a highpass filter.

We start with the basics of tight frames in Section 2.1 and their constructions in Section 2.2. As can be seen in (1.4), the recovering of the chopped and noded images can be reduced to the restoration of one-dimensional signals (along every fixed  $j$  in (1.4)). We therefore consider the univariate setting only. However, it is straightforward to extend the analysis given here to the multivariate case.

**2.1. Preliminaries on Tight Framelets.** A system  $X \subset L^2(\mathbb{R})$  is called a *tight frame* of  $L^2(\mathbb{R})$  if

$$f = \sum_{h \in X} \langle f, h \rangle h, \quad \forall f \in L^2(\mathbb{R}). \quad (2.1)$$

This is equivalent to

$$\|f\|_2^2 = \sum_{h \in X} |\langle f, h \rangle|^2, \quad \forall f \in L^2(\mathbb{R}), \quad (2.2)$$

where  $\langle \cdot, \cdot \rangle$  and  $\|\cdot\|_2 = \langle \cdot, \cdot \rangle^{1/2}$  are the inner product and norm of  $L^2(\mathbb{R})$ . It is clear that an orthonormal basis is a tight frame, and a tight frame is a generalization of orthonormal basis. A tight frame preserves the identities (2.1) and (2.2) which hold for an arbitrary orthonormal basis of  $L^2(\mathbb{R})$ . But it sacrifices the orthonormality and the linear independence of the system in order to get more flexibility. Therefore tight frames can be redundant. This redundancy is often useful in image processing applications such as denoising, see [16].

If  $X(\Psi)$  is the collection of the dilations and the shifts of a finite set  $\Psi \subset L^2(\mathbb{R})$ , i.e.,

$$X(\Psi) = \{2^{k/2}\psi(2^k x - j) : \psi \in \Psi, k, j \in \mathbb{Z}\},$$

then  $X(\Psi)$  is called a *wavelet* (or *affine*) *system*. In this case the elements in  $\Psi$  are called the *generators*. When  $X(\Psi)$  is a tight frame for  $L^2(\mathbb{R})$ , then  $\psi \in \Psi$  are called (*tight*) *framelets*.

A normal framelet construction starts with a refinable function. A compactly supported function  $\phi \in L^2(\mathbb{R})$  is *refinable* (a scaling function) with a refinement mask  $\tau_\phi$  if it satisfies

$$\widehat{\phi}(2\cdot) = \tau_\phi \widehat{\phi}.$$

Here  $\widehat{\phi}$  is the Fourier transform of  $\phi$ , and  $\tau_\phi$  is a trigonometric polynomial with  $\tau_\phi(0) = 1$ , i.e., a refinement mask of a refinable function must be a lowpass filter. One can define a multiresolution analysis from a given refinable function, we omit the detailed discussion here and refer the readers to [19, 26].

For a given compactly supported refinable function, the construction of tight framelet systems is to find a finite set  $\Psi$  that can be represented in the Fourier domain as

$$\widehat{\psi}(2\cdot) = \tau_\psi \widehat{\phi}$$

for some  $2\pi$ -periodic  $\tau_\psi$ . The unitary extension principle of [34] says that the wavelet system  $X(\Psi)$  generated by a finite set  $\Psi$  forms a tight frame in  $L^2(\mathbb{R})$  provided that the masks  $\tau_\phi$  and  $\{\tau_\psi\}_{\psi \in \Psi}$  satisfy:

$$\tau_\phi(\omega) \overline{\tau_\phi(\omega + \gamma\pi)} + \sum_{\psi \in \Psi} \tau_\psi(\omega) \overline{\tau_\psi(\omega + \gamma\pi)} = \delta_{\gamma,0}, \quad \gamma = 0, 1 \quad (2.3)$$

for almost all  $\omega$  in  $\mathbb{R}$ . Practically, we require all masks to be trigonometric polynomials. Thus, (2.3) together with the fact that  $\tau_\phi(0) = 1$  imply that  $\tau_\psi(0) = 0$  for all  $\psi \in \Psi$ . Hence,  $\{\tau_\psi\}_{\psi \in \Psi}$  must correspond to highpass filters. The sequences of Fourier coefficients of  $\tau_\psi$ , as well as  $\tau_\psi$  itself, are called *framelet masks*. The construction of framelets  $\Psi$  essentially is to design, for a given refinement mask  $\tau_\phi$ , framelet masks  $\{\tau_\psi\}_{\psi \in \Psi}$  such that (2.3) holds.

The unitary extension principle of [34] gives the flexibility in designing filters and will be used here. A more general principle of construction tight framelets, the oblique extension principle, was obtained recently in [18].

**2.2. Filter Design.** For any positive integer  $K$ , we need to design a set of framelets  $\Psi$  such that the chop-and-nod highpass filter given in (1.6) is one of the framelet masks (up to a constant factor). Note that the trigonometric polynomial corresponding to this chop-and-nod highpass filter is  $\sin^2(K\omega/2)$ . We have the following result:

**PROPOSITION 2.1.** *For an arbitrary odd number  $K$ , let  $\tau_2(\omega) = \sin^2(K\omega/2)$  be the given chop-and-nod highpass filter given in (1.6). Let*

$$\tau_0(\omega) = \cos^2\left(K\frac{\omega}{2}\right) \quad \text{and} \quad \tau_1(\omega) = -\sqrt{-2} \sin\left(K\frac{\omega}{2}\right) \cos\left(K\frac{\omega}{2}\right).$$

Then,  $\tau_0$ ,  $\tau_1$ , and  $\tau_2$  satisfy (2.3). Furthermore,

1. the function

$$\phi(x) = \begin{cases} \frac{1}{K} - \frac{|x|}{K^2}, & \text{if } x \in [-K, K], \\ 0, & \text{otherwise,} \end{cases} \quad (2.4)$$

is the refinable function with the refinement mask  $\tau_0$ , and

2.  $\psi_1$  and  $\psi_2$ , defined by

$$\widehat{\psi}_1(2\omega) = \tau_1(\omega) \widehat{\phi}(\omega), \quad \widehat{\psi}_2(2\omega) = \tau_2(\omega) \widehat{\phi}(\omega), \quad (2.5)$$

are tight framelets, i.e.  $X(\Psi)$ , with  $\Psi = \{\psi_1, \psi_2\}$ , forms a tight frame of  $L^2(\mathbb{R})$ .

*Proof.* The proof is a straightforward extension of spline framelets given in [34]. For completeness, we give an outline of the proof here. First, the spline theory says immediately that the refinable function corresponding to the mask  $\tau_0$  is the piecewise linear spline with  $1/K$  at origin and 0 at  $\pm K$ , i.e.  $\phi(x) = \frac{1}{K} - \frac{|x|}{K^2}$  for  $x \in [-K, K]$ , and 0, otherwise. Hence,  $\phi \in L^2(\mathbb{R})$  has a support  $[-K, K]$ . Secondly, we can check directly that  $\tau_0, \tau_1$ , and  $\tau_2$  satisfy (2.3). Therefore,  $X(\Psi)$  is a tight frame for  $L^2(\mathbb{R})$  by the unitary extension principle of [34].  $\square$

The Fourier coefficients of the masks  $\tau_0, \tau_1$  and  $\tau_2$  are:

$$a = \left\{ \frac{1}{4}, \underbrace{0, \dots, 0}_{K-1}, \frac{1}{2}, \underbrace{0, \dots, 0}_{K-1}, \frac{1}{4} \right\}, \quad (2.6)$$

$$b_1 = \left\{ -\frac{\sqrt{2}}{4}, \underbrace{0, \dots, 0}_{2K-1}, \frac{\sqrt{2}}{4} \right\}, \quad (2.7)$$

$$b_2 = \left\{ -\frac{1}{4}, \underbrace{0, \dots, 0}_{K-1}, \frac{1}{2}, \underbrace{0, \dots, 0}_{K-1}, -\frac{1}{4} \right\}. \quad (2.8)$$

Clearly  $b_2$  matches the chop-and-nod filter in (1.6). This, together with (2.4) and (2.5), leads to

$$\psi_i = 2 \sum_{j \in \mathbb{Z}} b_i(j) \phi(2 \cdot -j), \quad i = 1, 2.$$

Hence, the framelets are precisely the piecewise linear functions supported on  $[-K, K]$ .

When  $K$  is even, the simple construction of the tight frame in Proposition 2.1 does not work. Nevertheless, the filter design for even  $K$  can be done similarly. Since the real astronomical images we have are all obtained using odd  $K$ , we omit the discussion for this case.

**2.3. Matrix Form.** To implement our algorithm, we need to convert the filters to operators in matrix forms. For this, we need to consider boundary conditions, i.e. assumptions of the true image outside the region of interest. For simplicity, we use symmetric (reflective) extension here, see [32]. For other extensions, the derivation is similar, see for example [9]. In fact, for our method, the difference between symmetric and periodic boundary extensions is small.

For a given sequence  $h = \{h(j)\}_{j=-J}^J$ , we let  $T(h)$  to be the  $M$ -by- $M$  Toeplitz matrix

$$T(h) := \begin{bmatrix} h(0) & \cdots & h(-J) & & 0 \\ \vdots & \ddots & \ddots & h(-J) & \\ h(J) & \ddots & \ddots & \ddots & h(-J) \\ & \ddots & \ddots & \ddots & \vdots \\ 0 & & h(J) & \cdots & h(0) \end{bmatrix},$$

where  $M > 2J + 1$ . We also define two matrices:

$$T_\ell(h) := \left[ \begin{array}{cccccc|c} h(1) & h(2) & \cdots & h(J-1) & h(J) & & \\ h(2) & \ddots & \ddots & \ddots & \ddots & & \\ \vdots & \ddots & \ddots & & & & \\ h(J-1) & \ddots & \ddots & & 0 & & \\ h(J) & & & & & & \\ \hline & & & & & & 0_{J \times (M-J-1)} \\ \hline & & & & & & 0_{(M-J) \times (M-J-1)} \end{array} \right],$$

and

$$T_r(h) = \left[ \begin{array}{c|cccccc} 0_{(M-J) \times (M-J-1)} & & & & & & \\ \hline & & & & & & 0_{(M-J) \times J} \\ \hline & & & & & & h(-J) \\ & & & & & & h(-J+1) \\ & & & & & & \vdots \\ & & & & & & h(-2) \\ & & & & & & h(-1) \\ \hline 0_{J \times (M-J-1)} & & & & & & \\ \hline & & & & & & h(-J) & h(-J+1) & \cdots & h(-2) & h(-1) \end{array} \right].$$

They correspond to the symmetric boundary conditions on the left and on the right respectively. Finally let  $S_+(h)$  and  $S_-(h)$  be

$$S_+(h) = T_\ell(h) + T(h) + T_r(h) \quad \text{and} \quad S_-(h) = -T_\ell(h) + T(h) - T_r(h).$$

For the masks  $a$ ,  $b_1$  and  $b_2$  given in (2.6)–(2.8), their corresponding decomposition (and respectively reconstruction) matrices are:

$$H_0 = S_+(a), \quad H_1 = S_+(b_1), \quad H_2 = S_+(b_2) \quad (2.9)$$

(and

$$\tilde{H}_0 = S_+(a), \quad \tilde{H}_1 = S_-(-b_1), \quad \tilde{H}_2 = S_+(b_2).$$

respectively) with  $J = K$  and  $M = N + 2K$ . Clearly  $\tilde{H}_i = H_i^T = H_i$  for  $i = 0$  and  $2$ . Since  $b_1$  is antisymmetric,  $\tilde{H}_1 = H_1^T \neq H_1$ . Using (2.3), it is easy to verify that

$$\tilde{H}_0 H_0 + \tilde{H}_1 H_1 + \tilde{H}_2 H_2 = I, \quad (2.10)$$

where  $I$  is the identity.

**3. Framelet Algorithm.** In this section, we first show that Landweber method with  $\varrho = 1/16$  is a framelet method with no thresholding. Then we introduce a framelet thresholding scheme. Using that, we derive our main algorithm.

**3.1. Landweber Algorithm.** Notice that  $H_2$  defined by (2.9) and the imaging matrix  $A$  in (1.4) are related by:

$$H_2 = \frac{1}{4} \begin{bmatrix} * \\ A \\ * \end{bmatrix}, \quad (3.1)$$

where  $*$  here denotes non-zero matrices of size  $K$ -by- $(N + 2K)$ , cf. (1.6) and (2.8). Therefore, for any given  $\mathbf{f}^{(n)}$  we write

$$\tilde{H}_2 H_2 \mathbf{f}^{(n)} = \tilde{H}_2 \Lambda H_2 \mathbf{f}^{(n)} + \frac{1}{16} A^T A \mathbf{f}^{(n)} \quad (3.2)$$



where

$$\Lambda = \text{diag}(\underbrace{1, \dots, 1}_K, \underbrace{0, \dots, 0}_N, \underbrace{1, \dots, 1}_K).$$

Replacing  $A\mathbf{f}^{(n)}$  in (3.2) by  $\mathbf{g}$  and using (2.10), we obtain an iteration

$$\mathbf{f}^{(n+1)} = \mathbf{P}_+[\tilde{H}_0 H_0 \mathbf{f}^{(n)} + \tilde{H}_1 H_1 \mathbf{f}^{(n)} + (\tilde{H}_2 \Lambda H_2 \mathbf{f}^{(n)} + \frac{1}{16} A^T \mathbf{g})]. \quad (3.3)$$

The idea of the iteration (3.3) is as follows. We view the recovering of  $\mathbf{f}$  as the reconstruction of the finer level approximations of  $f$  from a given framelet coefficient sequence  $H_2 \mathbf{f}$ . Note that the major part of the sequence  $H_2 \mathbf{f}$  is already given as  $\mathbf{g}$ . We need the sequences  $H_0 \mathbf{f}$  and  $H_1 \mathbf{f}$ , which we do not have. At the  $(n+1)$ th iteration of the algorithm, we use the corresponding coefficient sequences of  $\mathbf{f}^{(n)}$  to approximate the missing ones. The first term in the right hand side of (3.3) represents the approximation of the low-frequency components of  $\mathbf{f}$  whereas the second term improves its high frequency approximation. Finally  $\mathbf{P}_+[\cdot]$  ensures that  $\mathbf{f}^{(n+1)}$  is a nonnegative vector.

By (2.10),  $\tilde{H}_0 H_0 \mathbf{f}^{(n)} + \tilde{H}_1 H_1 \mathbf{f}^{(n)} = \mathbf{f}^{(n)} - \tilde{H}_2 H_2 \mathbf{f}^{(n)}$ . Hence by (3.2), (3.3) can be rewritten as

$$\mathbf{f}^{(n+1)} = \mathbf{P}_+[\mathbf{f}^{(n)} + \frac{1}{16}(A^T \mathbf{g} - A^T A \mathbf{f}^{(n)})]. \quad (3.4)$$

Comparing it with (1.5), we see that (3.3), which uses the tight frame approach, is just a reformulation of the projected Landweber method with  $\varrho = 1/16$ .

As a side product of our approach, this new formulation puts the Landweber method within the multiresolution analysis framework and gives analytical representations of the chopped and noded image  $\mathbf{g}$  and the restored image  $\mathbf{f}$  in terms of the tight framelet systems. The analysis tells us that, at each iteration, the Landweber method tries to improve the low resolution approximation and the missing framelet coefficients of the true image, and it does so by combining the corresponding parts in the previous iterate with the given chopped and noded image.

In the next section, we will see that we denoise each  $\mathbf{f}^{(n)}$  by damping the framelet coefficients using a framelet denoising scheme, see (3.10). By comparing (3.3) with (3.10), we see that the framelet coefficients are not denoised at all in (3.3), which is equivalent to (3.4). Our new framelet viewpoint allows us to incorporate more sophisticated nonlinear denoising schemes to identify and remove the noise more accurately than the Landweber method (3.4).

**3.2. Framelet Denoising Scheme.** Here we introduce the framelet denoising scheme. We can use any reasonable framelet systems for our noise removal. For simplicity, we just use the framelet system in Proposition 2.1 by choosing  $K$  to be 1. Its masks are:  $\alpha = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$ ,  $\beta_1 = \{-\frac{\sqrt{2}}{4}, 0, \frac{\sqrt{2}}{4}\}$ , and  $\beta_2 = \{-\frac{1}{4}, \frac{1}{2}, -\frac{1}{4}\}$ . They are short masks, and hence will give a more efficient algorithm in noise suppression. To perform a multi-level framelet decomposition without down sampling in noise removal, we need the mask  $\alpha$  at level  $\ell$

$$\alpha^{(\ell)} = \left\{ \frac{1}{4}, \underbrace{0, \dots, 0}_{2^{(\ell-1)}-1}, \frac{1}{2}, \underbrace{0, \dots, 0}_{2^{(\ell-1)}-1}, \frac{1}{4} \right\}.$$

The masks  $\beta_1^{(\ell)}$  and  $\beta_2^{(\ell)}$  can be given similarly. Let

$$G_0^{(\ell)} = S_+(\alpha^{(\ell)}) \quad \text{and} \quad G_i^{(\ell)} = S_+(\beta_i^{(\ell)}), \quad i = 1, 2; \ell = 1, 2, \dots \quad (3.5)$$

Then the multi-level decomposition matrix to level  $L$  for this tight framelet system is

$$G = \begin{bmatrix} \frac{\prod_{\ell=0}^{L-1} G_0^{(L-\ell)}}{G_1^{(L)} \prod_{\ell=1}^{L-1} G_0^{(L-\ell)}} \\ \frac{\prod_{\ell=0}^{L-1} G_0^{(L-\ell)}}{G_2^{(L)} \prod_{\ell=1}^{L-1} G_0^{(L-\ell)}} \\ \vdots \\ G_1^{(1)} \\ G_2^{(1)} \end{bmatrix} \equiv \begin{bmatrix} G_L \\ G_H \end{bmatrix}, \quad (3.6)$$

and the corresponding reconstruction matrix  $\tilde{G} = G^T$ . By the tight framelet theory in [18], we have

$$\tilde{G}G = \tilde{G}_L G_L + \tilde{G}_H G_H = I. \quad (3.7)$$

For an arbitrary  $\mathbf{f}$ , the thresholding scheme  $\mathbf{D}$  is given by the following formula:

$$\mathbf{D}(\mathbf{f}) = \tilde{G}_L G_L \mathbf{f} + \tilde{G}_H \mathbf{T}_{\mathbf{u}_H}(G_H \mathbf{f}). \quad (3.8)$$

Here,

$$\mathbf{T}_{\mathbf{u}_H}((x_1, \dots, x_l, \dots)^T) = (t_{u_1}(x_1), \dots, t_{u_l}(x_l), \dots)^T,$$

with

$$\mathbf{u}_H = (u_1, \dots, u_l, \dots)^T = \underbrace{(\lambda_L, \dots, \lambda_L)}_{2(N+2K)}, \dots, \underbrace{(\lambda_\ell, \dots, \lambda_\ell)}_{2(N+2K)}, \dots, \underbrace{(\lambda_1, \dots, \lambda_1)}_{2(N+2K)}^T \quad (3.9)$$

and  $t_\lambda(x)$  being  $t_\lambda(x) = \text{sgn}(x) \max(|x| - \lambda, 0)$ , which is referred to as the *soft thresholding*. According to [23], the thresholding parameters  $\lambda_\ell$  are chosen to be

$$\lambda_\ell = 2^{-\ell/2} \kappa \sqrt{2 \log(N + 2K)}$$

where  $\kappa$  is the variance of the noise contained in  $\mathbf{f}^{(n)}$  estimated numerically by the method given in [23]. Our thresholding denoising scheme in the framelet domain is similar to that in the orthonormal wavelet domain [23]. As already pointed out by many authors, see for examples [18, 29, 35], framelets give better denoising results.

**3.3. Main Algorithm.** By applying the thresholding scheme (3.8) on (3.3), we have our main algorithm for 1D signals:

ALGORITHM 1.

(i) Let  $L$  be the number of framelet decomposition levels and  $\mathbf{f}^{(0)}$  be an initial guess.

(ii) Iterate on  $n$  until convergence:

$$\mathbf{f}^{(n+1)} = \mathbf{P}_+ \left[ \tilde{H}_0 \mathbf{D}(H_0 \mathbf{f}^{(n)}) + \tilde{H}_1 \mathbf{D}(H_1 \mathbf{f}^{(n)}) + (\tilde{H}_2 \Lambda H_2 \mathbf{f}^{(n)} + \frac{1}{16} A^T \mathbf{g}) \right], \quad (3.10)$$

where  $\mathbf{D}$  is the soft-thresholding operator given in (3.8) and  $\mathbf{P}_+$  is the projection operator onto the set  $\mathcal{P}_+$  of nonnegative vectors defined by

$$\mathcal{P}_+ = \{\mathbf{f} : \mathbf{f} \geq \mathbf{0} \text{ componentwise}\}.$$

Both framelet masks in (2.6)–(2.8) and masks in previous subsection for denoising have a total of 8 non-zero elements. Therefore the computational cost of the framelet decomposition and reconstruction in (3.10) needs  $16(L+1)(N+2K)$  multiplicative operations, while the cost of applying  $\mathbf{D}$  is  $O((N+2K)\log(N+2K))$ . In contrast, the projected Landweber method requires  $(5N+6K)$  multiplicative operations per iteration [2].

For 2D astronomical images, we first recall that the chop and nod process is a 1D process. In fact, in (1.4), we can restore the image columnwise for each fixed  $j$ . Thus there is no need to change the restoration part of Algorithm 1. More precisely, we can still use the 1D tight framelet system  $\{H_i, \tilde{H}_i\}_{i=0}^2$  for 2D images. However, to better capture the noise and artifacts in between the columns, we use a 2D noise reduction scheme. More precisely, we replace  $G$  and  $\tilde{G}$  by its 2D versions which can be obtained easily by using tensor products. For example, for 2D image  $\mathbf{f}$  expressed as a matrix, the framelet decomposition of  $\mathbf{f}$  gives the data:  $\{G\mathbf{f}G^T\}$  where  $G$  are given in (3.6). The denoising scheme in 2D is

$$\begin{aligned} \mathbf{D}(\mathbf{f}) &= \tilde{G}_L G_L \mathbf{f} G_L^T \tilde{G}_L^T + \tilde{G}_L \mathbf{T}_{\mathbf{u}_{LH}}(G_L \mathbf{f} G_L^T) \tilde{G}_L^T \\ &\quad + \tilde{G}_H \mathbf{T}_{\mathbf{u}_{HL}}(G_H \mathbf{f} G_H^T) \tilde{G}_H^T + \tilde{G}_H \mathbf{T}_{\mathbf{u}_{HH}}(G_H \mathbf{f} G_H^T) \tilde{G}_H^T. \end{aligned}$$

**4. Analysis of Algorithm 1.** In this section, we prove the convergence of Algorithm 1. We also show that its limit satisfies a certain minimization property. For simplicity, we give the 1D proof here. The proof can be extended to 2D images easily. Our proof is based on the framework of proximal forward-backward splitting proposed in [15] constructed under the theory of convex analysis and optimization. We first show that our algorithm can be written as an alternate direction algorithm for a minimization problem. Then we show that it converges.

**4.1. An Equivalent Formulation.** In the following, we partition any vector  $\mathbf{x} \in \mathbb{R}^{(4L+3)(N+2K)}$  into  $\mathbf{x}^T = [\mathbf{x}_{H_0}^T, \mathbf{x}_{H_1}^T, \mathbf{x}_{H_2}^T]$  such that  $\mathbf{x}_{H_0}, \mathbf{x}_{H_1} \in \mathbb{R}^{(2L+1)(N+2K)}$  and  $\mathbf{x}_{H_2} \in \mathbb{R}^{N+2K}$ . Notice that by (3.1) we have

$$\frac{1}{16} A^T \mathbf{g} = \frac{1}{4} [ * \mid A^T \mid * ] \begin{bmatrix} \mathbf{0} \\ \mathbf{g}/4 \\ \mathbf{0} \end{bmatrix} = H_2^T \begin{bmatrix} \mathbf{0} \\ \mathbf{g}/4 \\ \mathbf{0} \end{bmatrix} = \tilde{H}_2 \begin{bmatrix} \mathbf{0} \\ \mathbf{g}/4 \\ \mathbf{0} \end{bmatrix}.$$

Therefore, in matrix form, the iteration (3.10) can be rewritten into

$$\mathbf{f}^{(n+1)} = \mathbf{P}_+ \begin{bmatrix} \tilde{H}_0 \tilde{G} & \tilde{H}_1 \tilde{G} & \tilde{H}_2 \end{bmatrix} \begin{bmatrix} \mathbf{T}_{\mathbf{u}}(GH_0 \mathbf{f}^{(n)}) \\ \mathbf{T}_{\mathbf{u}}(GH_1 \mathbf{f}^{(n)}) \\ \Lambda H_2 \mathbf{f}^{(n)} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}/4 \\ \mathbf{0} \end{bmatrix} \end{bmatrix}, \quad (4.1)$$

where

$$\mathbf{u} \equiv \begin{bmatrix} \mathbf{u}_L \\ \mathbf{u}_H \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{u}_H \end{bmatrix} \quad (4.2)$$

with  $\mathbf{u}_H$  given in (3.9) and by (3.8),  $\mathbf{u}_L$  is a zero vector in  $\mathbb{R}^{N+2K}$ . Denote

$$\begin{bmatrix} \mathbf{T}_{\mathbf{u}}(GH_0 \mathbf{f}^{(n)}) \\ \mathbf{T}_{\mathbf{u}}(GH_1 \mathbf{f}^{(n)}) \\ \Lambda H_2 \mathbf{f}^{(n)} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}/4 \\ \mathbf{0} \end{bmatrix} \end{bmatrix} \equiv \mathbf{x}^{(n)} \equiv \begin{bmatrix} \mathbf{x}_{H_0}^{(n)} \\ \mathbf{x}_{H_1}^{(n)} \\ \mathbf{x}_{H_2}^{(n)} \end{bmatrix} \in \mathbb{R}^{(4L+3)(N+2K)}. \quad (4.3)$$

We are going to show that each component here is a solution to a minimization problem.

First, since the soft-thresholding is equivalent to a minimization procedure (cf. [15, 33]), i.e.,

$$t_\mu(z) = \arg \min_y \left\{ \frac{1}{2}(z - y)^2 + |\mu y| \right\}, \quad \mu \geq 0,$$

we have

$$\mathbf{T}_\mathbf{u}(\mathbf{z}) = \arg \min_{\mathbf{y}} \left\{ \frac{1}{2} \|\mathbf{z} - \mathbf{y}\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}.$$

Therefore,  $\{\mathbf{x}_{H_i}^{(n)}\}_{i=0}^1$  are solutions of the following minimization problems:

$$\mathbf{x}_{H_i}^{(n)} \equiv \mathbf{T}_\mathbf{u}(GH_i\mathbf{f}^{(n)}) = \arg \min_{\mathbf{y}} \left\{ \frac{1}{2} \|GH_i\mathbf{f}^{(n)} - \mathbf{y}\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}, \quad i = 0, 1. \quad (4.4)$$

Secondly, we can show that for an arbitrary vector  $\mathbf{x}$ ,

$$\arg \min_{\mathbf{y} \in \mathcal{C}} \left\{ \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 \right\} = \Lambda \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}/4 \\ \mathbf{0} \end{bmatrix}, \quad (4.5)$$

where

$$\mathcal{C} = \left\{ \mathbf{h} \in \mathbb{R}^{N+2K} : (I - \Lambda)\mathbf{h} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g}/4 \\ \mathbf{0} \end{bmatrix} \right\}.$$

Indeed, for any vector  $\mathbf{z} \in \mathcal{C}$ , we have

$$\begin{aligned} \|\mathbf{x} - \mathbf{z}\|_2^2 &= \sum_{i=1}^K (x_i - z_i)^2 + \sum_{i=K+1}^{N+K} (x_i - z_i)^2 + \sum_{i=N+K+1}^{N+2K} (x_i - z_i)^2 \\ &= \sum_{i=1}^K (x_i - z_i)^2 + \sum_{i=K+1}^{N+K} (x_i - g_{i-K}/4)^2 + \sum_{i=N+K+1}^{N+2K} (x_i - z_i)^2 \\ &\geq \sum_{i=K+1}^{N+K} (x_i - g_{i-K}/4)^2 = \left\| \mathbf{x} - \left( \Lambda \mathbf{x} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}/4 \\ \mathbf{0} \end{bmatrix} \right) \right\|_2^2. \end{aligned}$$

Equation (4.5) shows that  $\mathbf{x}_{H_2}^{(n)}$  is the solution of the minimization problem:

$$\mathbf{x}_{H_2}^{(n)} \equiv \Lambda H_2 \mathbf{f}^{(n)} + \begin{bmatrix} \mathbf{0} \\ \mathbf{g}/4 \\ \mathbf{0} \end{bmatrix} = \arg \min_{\mathbf{y} \in \mathcal{C}} \left\{ \frac{1}{2} \|H_2 \mathbf{f}^{(n)} - \mathbf{y}\|_2^2 \right\}. \quad (4.6)$$

Define the indicator function  $\iota_{\mathcal{C}}$  for the closed convex set  $\mathcal{C}$  by

$$\iota_{\mathcal{C}}(\mathbf{h}) = \begin{cases} 0, & \mathbf{h} \in \mathcal{C}, \\ +\infty, & \mathbf{h} \notin \mathcal{C}. \end{cases} \quad (4.7)$$

Equation (4.6) can be rewritten as

$$\mathbf{x}_{H_2}^{(n)} = \arg \min_{\mathbf{y}} \left\{ \frac{1}{2} \|H_2 \mathbf{f}^{(n)} - \mathbf{y}\|_2^2 + \iota_{\mathcal{C}}(\mathbf{y}) \right\}. \quad (4.8)$$

Denote

$$B = \begin{bmatrix} GH_0 \\ GH_1 \\ H_2 \end{bmatrix}. \quad (4.9)$$

By (4.4) and (4.8),  $\mathbf{x}^{(n)}$  can be written as the solution of the minimization problem:

$$\mathbf{x}^{(n)} = \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|B\mathbf{f}^{(n)} - \mathbf{x}\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}\|_1 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_1}\|_1 + \iota_{\mathcal{C}}(\mathbf{x}_{H_2}) \right\}. \quad (4.10)$$

Substituting (4.3) into (4.1), we obtain

$$\mathbf{f}^{(n+1)} = \mathbf{P}_+ \left\{ \begin{bmatrix} \tilde{H}_0 \tilde{G} & \tilde{H}_1 \tilde{G} & \tilde{H}_2 \end{bmatrix} \mathbf{x}^{(n)} \right\} = \mathbf{P}_+[B^T \mathbf{x}^{(n)}].$$

By the definition of  $\mathbf{P}_+$ ,  $\mathbf{f}^{(n+1)}$  is the solution of

$$\mathbf{f}^{(n+1)} = \arg \min_{\mathbf{f}} \left\{ \frac{1}{2} \|B^T \mathbf{x}^{(n)} - \mathbf{f}\|_2^2 + \iota_{\mathcal{P}_+}(\mathbf{f}) \right\}, \quad (4.11)$$

where  $\iota_{\mathcal{P}_+}$  is the indicator function of  $\mathcal{P}_+$  defined similar to (4.7). Combining (4.10) and (4.11), we can rewrite our iteration (3.10) in Algorithm 1 as

$$\begin{cases} \mathbf{x}^{(n)} = \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|B\mathbf{f}^{(n)} - \mathbf{x}\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}\|_1 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_1}\|_1 + \iota_{\mathcal{C}}(\mathbf{x}_{H_2}) \right\}, \\ \mathbf{f}^{(n+1)} = \arg \min_{\mathbf{f}} \left\{ \frac{1}{2} \|B^T \mathbf{x}^{(n)} - \mathbf{f}\|_2^2 + \iota_{\mathcal{P}_+}(\mathbf{f}) \right\}. \end{cases} \quad (4.12)$$

**4.2. Convergence.** To prove the convergence of the iteration (4.12), we recall the definitions of Moreau's proximal operator and Moreau's envelope originally introduced in [30, 31]. For any convex and lower semi-continuous function  $\xi$ , the proximal operator is defined by

$$\text{prox}_{\xi}(\mathbf{f}) \equiv \arg \min_{\mathbf{h}} \left\{ \frac{1}{2} \|\mathbf{f} - \mathbf{h}\|_2^2 + \xi(\mathbf{h}) \right\}. \quad (4.13)$$

The Moreau's envelope, which is a convex and differentiable function, is defined by

$${}^1\xi(\mathbf{f}) \equiv \min_{\mathbf{h}} \left\{ \frac{1}{2} \|\mathbf{f} - \mathbf{h}\|_2^2 + \xi(\mathbf{h}) \right\}. \quad (4.14)$$

By Lemma 2.5 in [15], the gradient of the envelope  ${}^1\xi$  is given by

$$\nabla({}^1\xi(\mathbf{f})) = \mathbf{f} - \text{prox}_{\xi}(\mathbf{f}). \quad (4.15)$$

Define

$$\varphi(\mathbf{x}) \equiv \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}\|_1 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_1}\|_1 + \iota_{\mathcal{C}}(\mathbf{x}_{H_2}).$$

By (4.12) and (4.13), we obtain

$$\mathbf{f}^{(n+1)} = \text{prox}_{\iota_{\mathcal{P}_+}}(B^T \mathbf{x}^{(n)}) = \text{prox}_{\iota_{\mathcal{P}_+}}(B^T \text{prox}_{\varphi}(B\mathbf{f}^{(n)})).$$

Since by (2.10) and (3.7),  $B^T B = I$ , we have

$$\begin{aligned}\mathbf{f}^{(n+1)} &= \text{prox}_{\iota_{\mathcal{P}_+}}(\mathbf{f}^{(n)} - B^T B \mathbf{f}^{(n)} + B^T \text{prox}_{\varphi}(B \mathbf{f}^{(n)})) \\ &= \text{prox}_{\iota_{\mathcal{P}_+}}(\mathbf{f}^{(n)} - B^T (B \mathbf{f}^{(n)} - \text{prox}_{\varphi}(B \mathbf{f}^{(n)}))).\end{aligned}$$

By (4.15) and the chain rule (as  $\nabla$  is the gradient with respect to  $\mathbf{f}^{(n)}$ ),

$$\nabla({}^1\varphi(B \mathbf{f}^{(n)})) = B^T (B \mathbf{f}^{(n)} - \text{prox}_{\varphi}(B \mathbf{f}^{(n)})). \quad (4.16)$$

Therefore,

$$\mathbf{f}^{(n+1)} = \text{prox}_{\iota_{\mathcal{P}_+}}(\mathbf{f}^{(n)} - \nabla({}^1\varphi(B \mathbf{f}^{(n)}))). \quad (4.17)$$

Thus we see that (3.10) in Algorithm 1, which is equivalent to (4.12), is equivalent to (4.17).

Let  $F_1(\mathbf{f}) \equiv \iota_{\mathcal{P}_+}(\mathbf{f})$ , and  $F_2(\mathbf{f}) \equiv {}^1\varphi(B \mathbf{f})$ . Then (4.17) becomes

$$\mathbf{f}^{(n+1)} = \text{prox}_{F_1}(\mathbf{f}^{(n)} - \nabla F_2(\mathbf{f}^{(n)})).$$

This iteration is the proximal forward-backward splitting in [15] for the minimization problem

$$\min_{\mathbf{f}} \{F_1(\mathbf{f}) + F_2(\mathbf{f})\},$$

which is equivalent to  $\min_{\mathbf{f} \in \mathcal{P}_+} F_2(\mathbf{f})$ . By (4.14), it is further equivalent to

$$\min_{\mathbf{f} \in \mathcal{P}_+} \left\{ \min_{\mathbf{x} \in \{\mathbf{x}: \mathbf{x}_{H_2} \in \mathcal{C}\}} \left\{ \frac{1}{2} \|B \mathbf{f} - \mathbf{x}\|_2^2 + \|\text{diag}(\mathbf{u}) \mathbf{x}_{H_0}\|_1 + \|\text{diag}(\mathbf{u}) \mathbf{x}_{H_1}\|_1 \right\} \right\}. \quad (4.18)$$

We now show that the iteration (4.17), which is equivalent to Algorithm 1, converges to a minimizer of (4.18). For this, we need the following result of [15], which is stated here in the finite dimensional case.

**PROPOSITION 4.1.** *Consider the minimization problem  $\min_{\mathbf{f} \in \mathbb{R}^{(N+2K)}} \{F_1(\mathbf{f}) + F_2(\mathbf{f})\}$ , where  $F_1$  is a convex and lower semi-continuous function, and  $F_2$  is a convex and differentiable function with a  $1/\nu$ -Lipschitz continuous gradient. Let  $0 < \mu < 2\nu$ , then for any initial guess  $\mathbf{f}^{(0)}$ , the iteration*

$$\mathbf{f}^{(n+1)} = \text{prox}_{F_1}(\mathbf{f}^{(n)} - \mu \nabla F_2(\mathbf{f}^{(n)}))$$

*converges to a minimizer of  $F_1(\mathbf{f}) + F_2(\mathbf{f})$  whenever a minimizer exists.*

We first verify that the conditions on  $F_1$  and  $F_2$  are satisfied.

**LEMMA 4.2.**  *$F_1(\mathbf{f}) \equiv \iota_{\mathcal{P}_+}(\mathbf{f})$  is a convex and lower semi-continuous function, and  $F_2(\mathbf{f}) \equiv {}^1\varphi(B \mathbf{f})$  is a convex and differentiable function with a  $1/\nu$ -Lipschitz continuous gradient.*

*Proof.* First, since  $\mathcal{P}_+$  is a closed and convex set,  $\iota_{\mathcal{P}_+}(\mathbf{f})$  is a convex and lower semi-continuous function. Furthermore, since  $\varphi$  is lower semi-continuous and convex, by Lemma 2.5 in [15] the envelope function  ${}^1\varphi(\cdot)$  is always convex and differentiable; hence  ${}^1\varphi(B \mathbf{f})$  is convex and differentiable. Next, we show the  $1/\nu$ -Lipschitz continuity of  $\nabla({}^1\varphi(B \mathbf{f}^{(n)}))$ . For this, we note that by Lemma 2.4 in [15], the inequality

$$\|(\mathbf{f} - \text{prox}_{\xi}(\mathbf{f})) - (\mathbf{h} - \text{prox}_{\xi}(\mathbf{h}))\|_2 \leq \|\mathbf{f} - \mathbf{h}\|_2$$

holds for any convex and lower semi-continuous  $\xi$ . Hence by (4.16),

$$\begin{aligned} \|\nabla^1\varphi(B\mathbf{f}) - \nabla^1\varphi(B\mathbf{h})\|_2 &= \|B^T(B\mathbf{f} - \text{prox}_\varphi(B\mathbf{f})) - B^T(B\mathbf{h} - \text{prox}_\varphi(B\mathbf{h}))\|_2 \\ &\leq \|B^T\|_2\|(B\mathbf{f} - \text{prox}_\varphi(B\mathbf{f})) - (B\mathbf{h} - \text{prox}_\varphi(B\mathbf{h}))\|_2 \\ &\leq \|B^T\|_2\|B(\mathbf{f} - \mathbf{h})\|_2 \leq \|B^T\|_2\|B\|_2\|\mathbf{f} - \mathbf{h}\|_2 \\ &= \|\mathbf{f} - \mathbf{h}\|_2. \end{aligned}$$

The last equality comes from the identity  $B^T B = I$  hence  $\|B^T\|_2 = \|B\|_2 = 1$ . Here the Lipschitz constant  $1/\nu = 1$  and hence  $\nu = 1$ .  $\square$

To show the convergence, it remains to prove the existence of a minimizer.

**THEOREM 4.3.** *If  $K$  and  $N$  are relatively prime, then (4.18) has a minimizer.*

The proof of this theorem is long, so we put it in the appendix. We remark that the requirement on  $K$  and  $N$  can always be achieved by adjusting  $K$  or  $N$ . Now we state our convergence theorem.

**THEOREM 4.4.** *When  $K$  and  $N$  are relatively prime, Algorithm 1 converges to a minimizer of (4.18) for arbitrary initial guess  $\mathbf{f}^{(0)}$ .*

*Proof.* By Lemma 4.2, both  $F_1$  and  $F_2$  satisfy the conditions in Theorem 4.1. Moreover, by Theorem 4.3, when  $K$  and  $N$  are relatively prime, a minimizer of  $\min_{\mathbf{f}}\{F_1(\mathbf{f}) + F_2(\mathbf{f})\}$  exists. Therefore, by Theorem 4.1, (4.17) converges to a minimizer of (4.18) for any initial guess. As we have already shown, Algorithm 1 is equivalent to (4.17), therefore Algorithm 1 converges to a minimizer of (4.18) for any initial guess  $\mathbf{f}^{(0)}$ .  $\square$

In (4.18), since part of the vector  $\mathbf{x}$  is exactly the given data  $\mathbf{g}$ , the term  $\|B\mathbf{f} - \mathbf{x}\|_2^2$  reflects the closeness of the solution  $A\mathbf{f}$  to the given data  $\mathbf{g}$ . Since  $B\mathbf{f}$  is the framelet (packet) coefficients of  $\mathbf{f}$ , the terms also reflects the closeness of the framelet packet coefficients to  $\mathbf{x}$ . The  $\ell_1$  norm of framelet (packet) coefficients are closely related to the Besov norm of  $\mathbf{f}$  (see e.g. [8, 25]). Hence the term

$$\frac{1}{2}\|B\mathbf{f} - \mathbf{x}\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}\|_1 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_1}\|_1$$

in (4.18) balances the closeness of the data and smoothness of the underlying solution.

**5. Numerical Results.** In this section, we test Algorithm 1 against the projected Landweber method for 1D and 2D examples. For comparison, we also include the algorithms developed in [3, 5]. We have used the following two error measures proposed in [2]:

1. the relative restoration error (RRE)

$$\varsigma_n := \frac{\|[\mathbf{f}^{(n)} + \text{mean}(\mathbf{f}^* - \mathbf{f}^{(n)})] - \mathbf{f}^*\|_2}{\|\mathbf{f}^*\|_2},$$

2. the relative discrepancy error (RDE):

$$\varepsilon_n := \frac{\|A\mathbf{f}^{(n)} - \mathbf{g}\|_2}{\|\mathbf{g}\|_2}.$$

Here  $\mathbf{f}^*$  is the true signal and  $\mathbf{f}^{(n)}$  is the  $n$ th iterate. In [2], RRE is used for synthetic data while RDE is for real data. For comparison, for synthetic data, we will also give  $\varsigma_{\text{OR},n}$ , which is the RRE computed by restricting both  $\mathbf{f}^*$  and  $\mathbf{f}^{(n)}$  onto the observation region (OR), i.e. the support of  $\mathbf{g}$ .

TABLE 5.1

Results for Example 1 by adding white noise with a varying standard derivation  $\sigma$ .

$\sigma$	Projected Landweber Method		Algorithm in [5]		Algorithm in [3]		Algorithm 1	
	$\varsigma_n$	$\varsigma_{OR,n}$	$\varsigma_n$	$\varsigma_{OR,n}$	$\varsigma_n$	$\varsigma_{OR,n}$	$\varsigma_n$	$\varsigma_{OR,n}$
0.01	0.1862	0.2409	0.6967	0.0438	0.6968	0.0344	0.0437	0.0235
0.02	0.1921	0.2423	0.6983	0.0557	0.6986	0.0506	0.0496	0.0334
0.04	0.2170	0.2561	0.7020	0.1035	0.7024	0.1148	0.1175	0.1018

TABLE 5.2

Results for Example 2 by adding white noise with a varying standard derivation  $\sigma$ .

$\sigma$	Projected Landweber Method		Algorithm in [5]		Algorithm in [3]		Algorithm 1	
	$\varsigma_n$	$\varsigma_{OR,n}$	$\varsigma_n$	$\varsigma_{OR,n}$	$\varsigma_n$	$\varsigma_{OR,n}$	$\varsigma_n$	$\varsigma_{OR,n}$
0.01	0.2094	0.1558	0.3031	0.1968	0.3632	0.2724	0.0291	0.0224
0.02	0.2223	0.1635	0.3039	0.1975	0.3634	0.2733	0.0368	0.0255
0.04	0.2514	0.1848	0.3080	0.2052	0.3647	0.2767	0.0682	0.0420

For synthetic data we stop the iteration when  $\varsigma_n$  reaches its minimum value. For real data, we stop the iteration when  $|\varepsilon_{n+1} - \varepsilon_n| < 10^{-3}$ . This stopping criterion was used in [3, 5]. The initial guess for all algorithms is set to be zero. For Algorithm 1, the number of decomposition level  $L$  in (3.6) is 5 in all experiments.

We first test three 1D examples, the first two of which are given in [2]. In all three examples,  $N = 128$  and  $K = 37$ . The true object  $\mathbf{f}^*$  has 202 points and the observation region lies between point 38 and point 165.

*Example 1.* The true object consists of two narrow Gaussian functions (simulating two bright stars over a black background), one inside and one outside the observation region (see Figure 5.1(a)).

*Example 2.* The true object consists of one narrow Gaussian function (simulating a bright star) over a smooth background (see Figure 5.2(a)).

*Example 3.* The true object consists of two narrow Gaussian functions (simulating two bright stars with different intensities) over a smooth background (see Figure 5.3(a)).

White Gaussian noise with standard deviations  $\sigma = 0.01, 0.02$ , and  $0.04$  are added to the chopped and noded signals (see Figures 5.1(b)–5.3(b)). The results for these three examples are tabulated in Tables 5.1–5.3. In all three examples, we can see the significant improvement of our method over the projected Landweber method and those in [3, 5]. We remark that the algorithm in [3] (resp. [5]) assumes the signal  $\mathbf{f}^*$  lives on  $[K + 1, N + K]$  and imposes the periodic (resp. Dirichlet) boundary condition at  $K + 1$  to  $N + K$ . In contrast, we assume the Neumann boundary condition for the signal at indices 1 and  $N + 2K$ . We show the visual results in Figures 5.1–5.3. Table 5.4 shows the numbers of iterations and the CPU time for generating the corresponding results listed in Tables 5.1–5.3. The CPU time is the average time of 100 runs of the algorithms on a 2.16 GHz Pentium-IV Dell Laptop. We see that the timing of our algorithm is higher than those of Landweber type methods but is quite manageable. Our method can be used as a good post-processing method to clean up these kinds of infrared images.

Finally we consider a real 2D image obtained from United Kingdom Infra-Red Telescope [2]. The results are given in Figure 5.4. It is clear from the figures that our results have much less noise.



TABLE 5.3

Results for Example 3 by adding white noise with a varying standard derivation  $\sigma$ .

$\sigma$	Projected Landweber Method		Algorithm in [5]		Algorithm in [3]		Algorithm 1	
	$\varsigma_n$	$\varsigma_{\text{OR},n}$	$\varsigma_n$	$\varsigma_{\text{OR},n}$	$\varsigma_n$	$\varsigma_{\text{OR},n}$	$\varsigma_n$	$\varsigma_{\text{OR},n}$
0.01	0.2124	0.1568	0.3072	0.1982	0.3925	0.3074	0.0508	0.0396
0.02	0.2254	0.1644	0.3080	0.1988	0.3929	0.3085	0.0695	0.0507
0.04	0.2547	0.1857	0.3120	0.2065	0.3944	0.3121	0.0894	0.0548

TABLE 5.4

Required numbers of iterations and the CPU time (in second) of four algorithms for three examples.

$\sigma$	Projected Landweber Method		Algorithm in [5]		Algorithm in [3]		Algorithm 1	
	Itr	CPU time	Itr	CPU time	Itr	CPU time	Itr	CPU time
Example 1								
0.01	148	0.0227	6	0.0009	4	0.0003	904	4.0630
0.02	64	0.0095	6	0.0009	4	0.0005	650	2.9234
0.04	39	0.0059	5	0.0008	3	0.0003	546	2.4531
Example 2								
0.01	48	0.0073	7	0.0011	3	0.0003	928	4.1656
0.02	38	0.0056	7	0.0010	3	0.0004	771	3.4693
0.04	25	0.0033	6	0.0008	3	0.0003	582	2.6312
Example 3								
0.01	48	0.0077	7	0.0011	3	0.0005	906	4.1172
0.02	38	0.0058	7	0.0009	3	0.0005	669	3.2010
0.04	25	0.0036	6	0.0010	3	0.0003	445	2.0625

**6. Appendix.** In this appendix, we prove Theorem 4.3, i.e., the existence of a minimizer of (4.18). Notice that (4.18) is equivalent to

$$\min_{\mathbf{f} \in \mathcal{P}_+} F_2(\mathbf{f}), \quad (6.1)$$

where

$$F_2(\mathbf{f}) \equiv {}^1\varphi(B\mathbf{f}) = \min_{\mathbf{x}} \left\{ \frac{1}{2} \|B\mathbf{f} - \mathbf{x}\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}\|_1 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_1}\|_1 + \iota_C(\mathbf{x}_{H_2}) \right\}.$$

Instead of considering the above minimization problem, we consider the following minimization problem first:

$$\min_{\mathbf{f} \in \mathcal{W}} F_2(\mathbf{f}), \quad (6.2)$$

where  $\mathcal{W} = \{\mathbf{w} : \mathbf{w}^T \mathbf{1} = 0\}$  and  $\mathbf{1}$  is the vector of all ones. We will then prove the existence of a minimizer of (6.2). Then we show that the existence of a minimizer of (6.2) implies the existence of a minimizer of (6.1). This concludes that (4.18) has a minimizer.

To start, we need the following lemma on the eigenvalues and eigenvectors of the matrices appearing in the matrix  $B$  in (4.9). In what follows, the discrete cosine transform (DCT) matrix refers to the DCT matrix of the type II in [27].

LEMMA 6.1. *The eigenvalues  $\gamma_i$  of the matrix  $H_0$  in (2.9) is given by*

$$\gamma_i = \cos^2 \frac{iK\pi}{2(N+2K)}, \quad i = 0, 1, \dots, N+2K-1, \quad (6.3)$$

and the corresponding eigenvectors are the columns of the discrete cosine transform (DCT) matrix. The eigenvalues  $\gamma_i^{(\ell)}$  of the matrices  $G_0^{(\ell)}$  in (3.5) are given by

$$\gamma_i^{(\ell)} = \cos^2 \frac{i2^{\ell-1}\pi}{2(N+2K)}, \quad i = 0, 1, \dots, N+2K-1,$$

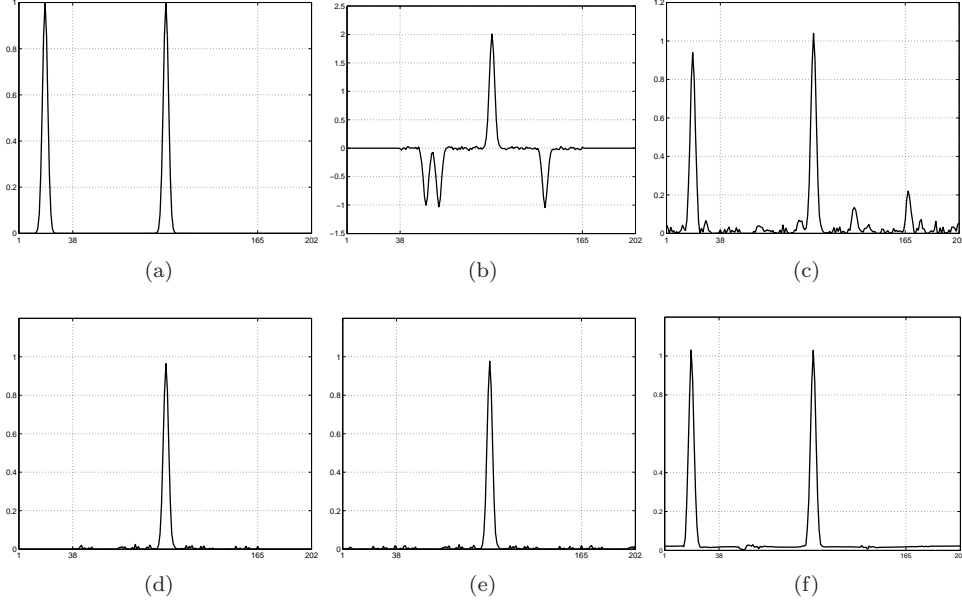


FIG. 5.1. *Example 1. (a) Original signal, (b) chopped and noded signal with Gaussian noise of standard deviation 0.02, (c) result by the projected Landweber method (64 iterations); (d) result by the algorithm in [5] (6 iterations); (e) result by the algorithm in [3] (4 iterations); and (f) result by Algorithm 1 (650 iterations).*

and the corresponding eigenvectors are also columns of the DCT matrix.

*Proof.* By the definition of  $H_0$  and Lemma 3.1 in [32], the matrix  $H_0$  can be diagonalized by the DCT matrix. By the formula (3.3) in [32], after calculation, we can see that the eigenvalues of  $H_0$  are given exactly by (6.3). Since  $G_0^{(\ell)}$  are special cases of  $H_0$  with  $K = 2^{\ell-1}$ , the statements for  $G_0^{(\ell)}$  can be proved similarly.  $\square$

The following lemma shows the existence of a minimizer of (6.2).

LEMMA 6.2. *If  $K$  and  $N$  are relatively prime, then (6.2) has a minimizer.*

*Proof.* Since  $F_2(\mathbf{f})$  is lower semi-continuous and convex, in order to prove the existence of a minimizer of (6.2), by Theorem 2.5.1(ii) in [36], it suffices to show that  $F_2$  is coercive in  $\mathcal{W}$ , i.e., if  $\mathbf{w} \in \mathcal{W}$  is such that  $\|\mathbf{w}\|_2 \rightarrow \infty$ , then  $F_2(\mathbf{w}) \rightarrow \infty$ .

Let  $\mathbf{w} \in \mathcal{W}$ , we define

$$\mathbf{x}^* \equiv \arg \min_{\mathbf{x}} \left\{ \frac{1}{2} \|B\mathbf{w} - \mathbf{x}\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}\|_1 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_1}\|_1 + \iota_{\mathcal{C}}(\mathbf{x}_{H_2}) \right\}.$$

Then by the definition of  $B$  in (4.9) and (4.4),

$$\mathbf{x}_{H_0}^* = \arg \min_{\mathbf{x}_{H_0}} \left\{ \frac{1}{2} \|GH_0\mathbf{w} - \mathbf{x}_{H_0}\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}\|_1 \right\} = \mathbf{T}_{\mathbf{u}}(GH_0\mathbf{w}).$$

By (4.2), we have

$$\begin{aligned} F_2(\mathbf{w}) &\equiv \frac{1}{2} \|B\mathbf{w} - \mathbf{x}^*\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}^*\|_1 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_1}^*\|_1 + \iota_{\mathcal{C}}(\mathbf{x}_{H_2}^*) \\ &\geq \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}^*\|_1 = \|\text{diag}(\mathbf{u}_H)\mathbf{T}_{\mathbf{u}_H}(GH_0\mathbf{w})\|_1. \end{aligned}$$

Let  $\lambda_m$  and  $\lambda_M$  be the smallest and largest entries in  $\mathbf{u}_H$  respectively. Then

$$F_2(\mathbf{w}) \geq \lambda_m \|GH_0\mathbf{w}\|_1 - \lambda_m \lambda_M (N + 2K) \geq \lambda_m \|GH_0\mathbf{w}\|_2 - \lambda_m \lambda_M (N + 2K).$$

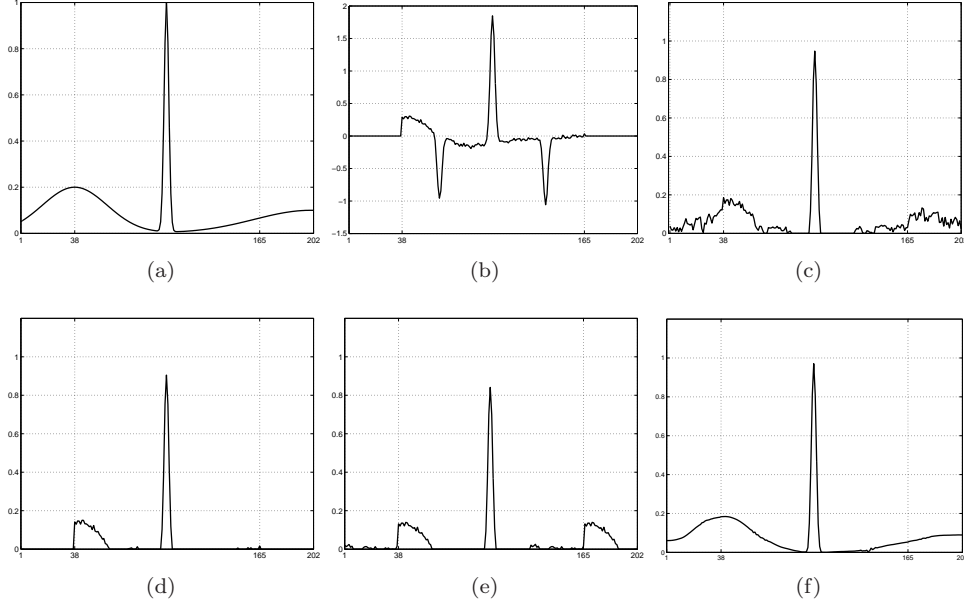


FIG. 5.2. *Example 2. (a) Original signal, (b) chopped and noded signal with Gaussian noise of standard deviation 0.02, (c) result by the projected Landweber method (38 iterations); (d) result by the algorithm in [5] (7 iterations); (e) result by the algorithm in [3] (3 iterations); and (f) result by Algorithm 1 (771 iterations).*

Denote  $\mathbf{v} = H_0 \mathbf{w}$ . By (3.7), we obtain

$$\begin{aligned}
 F_2(\mathbf{w}) &\geq \lambda_m \|G_H \mathbf{v}\|_2 - \lambda_m \lambda_M (N + 2K) \\
 &= \lambda_m \sqrt{\mathbf{v}^T G_H^T G_H \mathbf{v}} - \lambda_m \lambda_M (N + 2K) \\
 &= \lambda_m \sqrt{\mathbf{v}^T (I - G_L^T G_L) \mathbf{v}} - \lambda_m \lambda_M (N + 2K). \tag{6.4}
 \end{aligned}$$

In order to estimate  $\sqrt{\mathbf{v}^T (I - G_L^T G_L) \mathbf{v}}$ , we need to consider the eigenvalues and eigenvectors of

$$I - G_L^T G_L = I - \prod_{\ell=1}^L (G_0^{(\ell)})^T \prod_{\ell=0}^{L-1} G_0^{(L-\ell)}.$$

By Lemma 6.1, all the matrices  $G_0^{(\ell)}$  can be diagonalized by the DCT matrix. Therefore, the matrix  $I - G_L^T G_L$  can be diagonalized by the DCT matrix too, and the eigenvalues are given by

$$1 - \prod_{\ell=0}^L (\gamma_i^{(\ell)})^2,$$

which, by Lemma 6.1, equals to 0 if and only if  $i = 0$ . Therefore, the null space of  $I - G_L^T G_L$  is of dimension 1. It can be verified straightforwardly that

$$(I - G_L^T G_L) \mathbf{1} = \mathbf{0}.$$

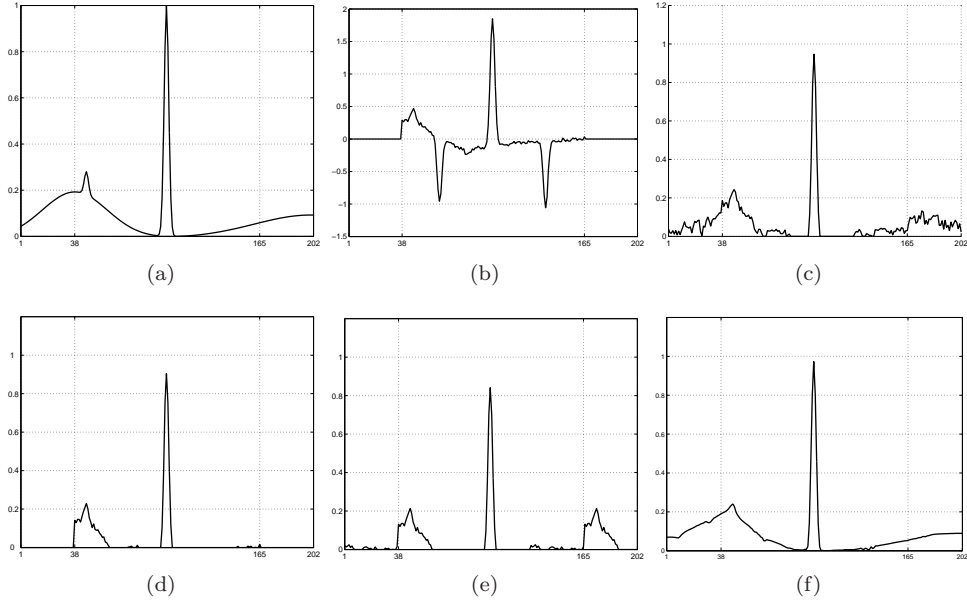


FIG. 5.3. *Example 3.* (a) Original signal, (b) chopped and noded signal with Gaussian noise of standard deviation 0.02, (c) result by the projected Landweber method (38 iterations); (d) result by the algorithm in [5] (7 iterations); (e) result by the algorithm in [3] (3 iterations); and (f) result by Algorithm 1 (669 iterations).

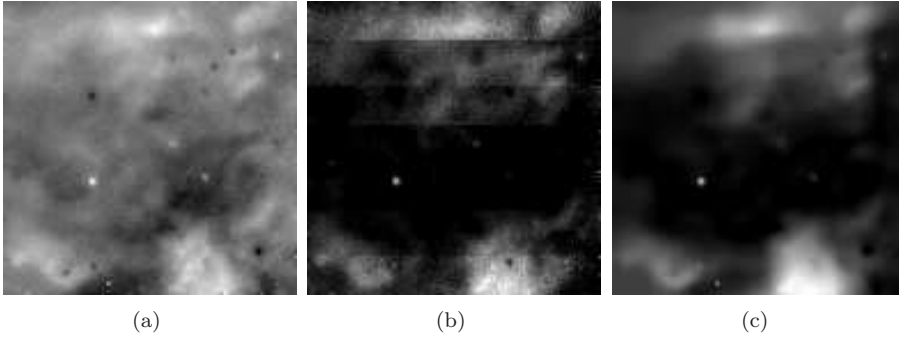


FIG. 5.4.  $K = 37$  and  $N = 128$ . (a) The chopped and noded image  $\mathbf{g}$ ; (b) result from the projected Landweber method (241 iterations and  $\varepsilon_n = 0.000993$ ); and (c) result from Algorithm 1 (202 iterations and  $\varepsilon_n = 0.000994$ ).

Hence, the subspace spanned by  $\mathbf{1}$  is the null space of  $I - G_L^T G_L$ . It implies that

$$\mathbf{w}^T (I - G_L^T G_L) \mathbf{w} \geq \sigma \|\mathbf{w}\|_2^2, \quad \forall \mathbf{w} \in \mathcal{W}, \quad (6.5)$$

where  $\sigma$  is the second smallest eigenvalue of  $I - G_L^T G_L$ .

On the other hand, by (6.3), the eigenvalues of  $H_0$  are  $\gamma_i = \cos^2 \theta_i$ , where

$$\theta_i \equiv \frac{iK\pi}{2(N+2K)} = \frac{\pi}{2} \frac{iK}{N+2K}, \quad i = 0, 1, \dots, N+2K-1.$$

If  $K$  and  $N$  are relatively prime, then  $\theta_i$  cannot be odd integral multiple of  $\frac{\pi}{2}$ . There-

fore,  $H_0$  is nonsingular, and the smallest eigenvalues

$$\gamma = \min_i \gamma_i > 0. \quad (6.6)$$

Furthermore, when  $K$  and  $N$  are relatively prime,  $\theta_i$  cannot be even integral multiple of  $\frac{\pi}{2}$  unless  $i = 0$ . It implies that 1 is a simple eigenvalue of  $H_0$ . It is easy to verify that  $\mathbf{1}$  is the corresponding eigenvector. Therefore,  $\mathcal{W}$  is an invariant subspace of  $H_0$ , i.e.,  $\mathbf{v} = H_0 \mathbf{w}$  is also in  $\mathcal{W}$ . Hence, by (6.5), we obtain

$$\mathbf{v}^T (I - G_L^T G_L) \mathbf{v} \geq \sigma \|\mathbf{v}\|_2^2.$$

This together with (6.4), (6.6) and the definition of  $\mathbf{v}$  imply that

$$\begin{aligned} F_2(\mathbf{w}) &\geq \lambda_m \sqrt{\sigma} \|\mathbf{v}\|_2 - \lambda_m \lambda_M (N + 2K) = \lambda_m \sqrt{\sigma} \|H_0 \mathbf{w}\|_2 - \lambda_m \lambda_M (N + 2K) \\ &\geq \lambda_m \sqrt{\sigma} \gamma \|\mathbf{w}\|_2 - \lambda_m \lambda_M (N + 2K). \end{aligned}$$

Thus,  $F_2(\mathbf{w})$  goes to infinity when  $\|\mathbf{w}\|_2$  goes to infinity. It means that  $F_2$  is coercive in  $\mathcal{W}$ . Hence a minimizer of (6.2) exists by Theorem 2.5.1(ii) in [36].  $\square$

Next we show that the existence of a minimizer of (6.2) implies the existence of a minimizer of (6.1).

LEMMA 6.3. *Assume that a minimizer of (6.2) exists. Then a minimizer of (6.1) also exists.*

*Proof.* First, we observe that if the identity

$$F_2(\mathbf{f} + c\mathbf{1}) = F_2(\mathbf{f}), \quad \forall c \in \mathbb{R} \quad (6.7)$$

holds for every  $\mathbf{f}$ , then the conclusion follows immediately. Indeed, let  $\mathbf{w}^* \in \mathcal{W}$  be a minimizer of (6.2), i.e.  $F_2(\mathbf{w}) \geq F_2(\mathbf{w}^*)$  for all  $\mathbf{w} \in \mathcal{W}$ . Let  $\mathbf{p}_+^* = \mathbf{w}^* + \rho\mathbf{1}$ , where  $\rho$  is the absolute value of the smallest entry of  $\mathbf{w}^*$ . It is obvious that  $\mathbf{p}_+^* \in \mathcal{P}_+$ . For any  $\mathbf{p}_+ \in \mathcal{P}_+$ , we orthogonally decompose it into  $\mathbf{p}_+ = \mathbf{w} + \delta\mathbf{1}$ , where  $\mathbf{w} \in \mathcal{W}$ . Then,

$$F_2(\mathbf{p}_+) = F_2(\mathbf{w} + \delta\mathbf{1}) = F_2(\mathbf{w}) \geq F_2(\mathbf{w}^*) = F_2(\mathbf{w}^* + \rho\mathbf{1}) = F_2(\mathbf{p}_+^*),$$

which implies that  $\mathbf{p}_+^*$  is a minimizer of  $F_2(\mathbf{f})$  in  $\mathcal{P}_+$ , i.e., a minimizer of (6.1).

Next, we verify the equality (6.7). For this, we first define

$$E_0^{(L)}(\mathbf{y}) \equiv \min_{\mathbf{x}_{H_0}^{(L)}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{x}_{H_0}^{(L)}\|_2^2 + \|\text{diag}(\mathbf{u}_L) \mathbf{x}_{H_0}^{(L)}\|_1 \right\},$$

and

$$E_0^{(H)}(\mathbf{y}) \equiv \min_{\mathbf{x}_{H_0}^{(H)}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{x}_{H_0}^{(H)}\|_2^2 + \|\text{diag}(\mathbf{u}_H) \mathbf{x}_{H_0}^{(H)}\|_1 \right\}.$$

By (4.2),  $\mathbf{u}_L$  is a zero vector. Therefore,  $E_0^{(L)}(\mathbf{y}) = 0$  for all  $\mathbf{y}$ . Define

$$E_1(\mathbf{y}) \equiv \min_{\mathbf{y}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{x}_{H_1}\|_2^2 + \|\text{diag}(\mathbf{u}) \mathbf{x}_{H_1}\|_1 \right\}$$

and

$$E_2(\mathbf{y}) \equiv \min_{\mathbf{y}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{x}_{H_2}\|_2^2 + \iota_C(\mathbf{x}_{H_2}) \right\}.$$

Since the minimization of each term in

$$\frac{1}{2}\|B\mathbf{f} - \mathbf{x}\|_2^2 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_0}\|_1 + \|\text{diag}(\mathbf{u})\mathbf{x}_{H_1}\|_1 + \iota_C(\mathbf{x}_{H_2})$$

is independent, we have

$$F_2(\mathbf{f}) = E_0^{(L)}(G_L H_0 \mathbf{f}) + E_0^{(H)}(G_H H_0 \mathbf{f}) + E_1(GH_1 \mathbf{f}) + E_2(H_2 \mathbf{f}).$$

By direct calculation, one has

$$H_0 \mathbf{1} = \mathbf{1}, \quad H_1 \mathbf{1} = \mathbf{0}, \quad H_2 \mathbf{1} = \mathbf{0}, \quad G_H \mathbf{1} = \mathbf{0}.$$

This leads to

$$G_H H_0(\mathbf{f} + c\mathbf{1}) = G_H H_0 \mathbf{f}, \quad GH_1(\mathbf{f} + c\mathbf{1}) = GH_1 \mathbf{f}, \quad H_2(\mathbf{f} + c\mathbf{1}) = H_2 \mathbf{f},$$

for all  $c \in \mathbb{R}$ . Hence

$$\begin{aligned} F_2(\mathbf{f} + c\mathbf{1}) &= E_0^{(L)}(G_L H_0(\mathbf{f} + c\mathbf{1})) + E_0^{(H)}(G_H H_0(\mathbf{f} + c\mathbf{1})) \\ &\quad + E_1(GH_1(\mathbf{f} + c\mathbf{1})) + E_2(GH_2(\mathbf{f} + c\mathbf{1})) \\ &= E_0^{(H)}(G_H H_0 \mathbf{f}) + E_1(GH_1 \mathbf{f}) + E_2(GH_2 \mathbf{f}) \\ &= E_0^{(L)}(G_L H_0 \mathbf{f}) + E_0^{(H)}(G_H H_0 \mathbf{f}) + E_1(GH_1 \mathbf{f}) + E_2(GH_2 \mathbf{f}) = F_2(\mathbf{f}), \end{aligned}$$

which completes the proof of (6.7)  $\square$

Combining Lemmas 6.2 and 6.3, we obtain Theorem 4.3.

**Acknowledgment:** We thank Prof. Bertero for introducing us to the topic and his many advices and helps in our research. We would like to thank the referees for providing us with valuable comments and insightful suggestions which have brought improvements to several aspects of this manuscript.

#### REFERENCES

- [1] M. BERTERO AND P. BOCCACCI, *Introduction to inverse problems in imaging*, Institute of Physics Pub., Bristol, 1998.
- [2] M. BERTERO, P. BOCCACCI, F. D. BENEDETTO, AND M. ROBERTO, *Restoration of chopped and nodded images in infrared astronomy*, *Inverse Problem*, 15 (1999), pp. 345–372.
- [3] M. BERTERO, P. BOCCACCI, A. CUSTO, C. DE MOL, AND M. ROBERTO, *A Fourier-based method for the restoration of chopped and nodded images*, *Astronomy and Astrophysics*, 406 (2003), pp. 765–772.
- [4] M. BERTERO, P. BOCCACCI, AND M. ROBERTO, *An inversion method for the restoration of chopped and nodded images*, in *Infrared Astronomical Instrumentation*, A. M. Fowler, ed., vol. 3354 of Proc. SPIE, 1998, pp. 877–886.
- [5] M. BERTERO, P. BOCCACCI, AND M. ROBERTO, *Inversion of second-difference operators with application to infrared astronomy*, *Inverse Problems*, 19 (2003), pp. 1427–1443.
- [6] G. BEYLKIN, R. COIFMAN, AND V. ROKHLIN, *Fast wavelet transforms and numerical algorithms I*, *Communications on Pure and Applied Mathematics*, 44 (1991), pp. 141–183.
- [7] L. BORUP, R. GRIBONVAL AND M. NIELSEN, *Tight wavelet frames in Lebesgue and Sobolev spaces*. *Journal of Functional Spaces and Applications* 2 (2004), pp. 227–252.
- [8] L. BORUP, R. GRIVONBAL, AND M. NIELSEN, *Bi-framelet systems with few vanishing moments characterize Besov spaces*, *Applied and Computational Harmonic Analysis*, 17 (2004), pp. 3–28.
- [9] S. S. CAPIZZANO, *A note on anti-reflective boundary conditions and fast deblurring models*, *SIAM Journal on Scientific Computing*, 25 (2003), pp. 1307–1325.
- [10] A. CHAI AND Z. SHEN, *Deconvolution: A wavelet frame approach*, *Numerische Mathematik*, 106 (2007), pp. 529–587.

- [11] R. CHAN, T. CHAN, L. SHEN, AND Z. SHEN, *Wavelet algorithms for high-resolution image reconstruction*, SIAM Journal on Scientific Computing, 24 (2003), pp. 1408–1432.
- [12] R. CHAN, S. D. RIEMENSCHNEIDER, L. SHEN, AND Z. SHEN, *Tight frame: The efficient way for high-resolution image reconstruction*, Applied and Computational Harmonic Analysis, 17 (2004), pp. 91–115.
- [13] A. COHEN, M. HOFFMANN, AND M. REISS, *Adaptive wavelet galerkin methods for linear inverse problems*, Tech. Report, May 2002.
- [14] R. Coifman and D. Donoho, *Translation-Invariant De-Noising*, technical report, 1995.
- [15] P. COMBETTES AND V. WAJS, *Signal recovery by proximal forward-backward splitting*, Multi-scale Modeling and Simulation: A SIAM Interdisciplinary Journal, 4 (2005), pp. 1168–1200.
- [16] I. DAUBECHIES, *Ten Lectures on Wavelets*, vol. 61 of CBMS Conference Series in Applied Mathematics, SIAM, Philadelphia, 1992.
- [17] I. DAUBECHIES, M. DEFRISE, AND C. D. MOL, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Communications on Pure and Applied Mathematics, 57 (2004), pp. 1413–1541.
- [18] I. DAUBECHIES, B. HAN, A. RON, AND Z. SHEN, *Framelets: MRA-based constructions of wavelet frames*, Applied and Computation Harmonic Analysis, 14 (2003), pp. 1–46.
- [19] C. DE BOOR, R. DEVORE, AND A. RON, *On the construction of multivariate (pre)-wavelets*, Constructive Approximation, 9 (1993), pp. 123–166.
- [20] C. DE MOL AND M. DEFRISE, *A note on wavelet-based inversion methods, in inverse problem, image analysis and medical imaging*, Contemporary Mathematics, (American Mathematics Society, 2002), pp. 85–96.
- [21] F. DI BENEDETTO, *The  $m$ -th difference operator applied to  $l^2$  functions on a finite interval*, Linear Algebra and its Applications, 366 (2003), pp. 173–198.
- [22] D. DONOHO, *Nonlinear solution of linear inverse problems by wavelet-vagulette decomposition*, Applied and Computational Harmonic Analysis, (1995), pp. 101–126.
- [23] D. DONOHO AND I. JOHNSTONE, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, 81 (1994), pp. 425–455.
- [24] M. FIGUEIREDO AND R. D. NOWAK, *An EM algorithm for wavelet-based image restoration*, IEEE Transactions on Image Processing, 12 (2003), pp. 906–916.
- [25] Y. HUR AND A. RON, *Caplets: wavelet representations without wavelets*, preprint, (2005).
- [26] R. JIA AND Z. SHEN, *Multiresolution and wavelets*, Proceedings of the Edinburgh Mathematical Society, 37 (1994), pp. 271–300.
- [27] T. KAILATH AND V. OLSHEVSKY, *Displacement structure approach to discrete trigonometric transform based preconditioners of G.Strang and T.Chan types*, SIAM Journal on Matrix Analysis and Applications, 26 (2005), pp. 706–734.
- [28] J. KALIFA, S. MALLAT, AND B. ROUGÉ, *Deconvolution by thresholding in mirror wavelet bases*, IEEE Transactions on Image Processing, 12 (2003), pp. 446–457.
- [29] S. MALLAT, *A Wavelet Tour of Signal Processing*, Academic Press, 2nd ed., 1999.
- [30] J.-J. MOREAU, *Fonctions convexes duales et points proximaux dans un espace hilbertien*, C.R. Acad. Sci. Paris Sér. A Math., 255 (1962), pp. 1897–2899.
- [31] ———, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.
- [32] M. NG, R. CHAN, AND W. TANG, *A fast algorithm for deblurring models with Neumann boundary conditions*, SIAM Journal on Scientific Computing, 21 (2000), pp. 851–866.
- [33] M. NIKOLOVA, *Local strong homogeneity of a regularized estimator*, SIAM Journal on Applied Mathematics, 61 (2000), pp. 633–658.
- [34] A. RON AND Z. SHEN, *Affine system in  $L_2(R^d)$ : the analysis of the analysis operator*, Journal of Functional Analysis, 148 (1997), pp. 408–447.
- [35] L. SHEN, I. KAKADIARIS, M. PAPADAKIS, I. KONSTANTINIDIS, D. KOURI, AND D. HOFFMAN, *Image denoising using a tight frame*, IEEE Transactions on Image Processing, 15 (2006), pp. 1254–1263.
- [36] C. ZĂLINESCU, *Convex Analysis in General Vector Spaces*, World Scientific, River Edge, NJ, 2002.