A FRAMELET-BASED IMAGE INPAINTING ALGORITHM

JIAN-FENG CAI*, RAYMOND H. CHAN†, AND ZUOWEI SHEN‡

Abstract. Image inpainting is a fundamental problem in image processing and has many applications. Motivated by the recent tight frame based methods on image restoration in either the image or the transform domain, we propose an iterative tight frame algorithm for image inpainting. We consider the convergence of this framelet-based algorithm by interpreting it as an iteration for minimizing a special functional. The proof of the convergence is under the framework of convex analysis and optimization theory. We also discuss the relationship of our method with other wavelet-based methods. Numerical experiments are given to illustrate the performance of the proposed algorithm.

Key words. Tight frame, inpainting, convex analysis

1. Introduction. The problem of inpainting [2] occurs when part of the pixel data in a picture is missing or over-written by other means. This arises for example in restoring ancient drawings, where a portion of the picture is missing or damaged due to aging or scratch; or when an image is transmitted through a noisy channel. The task of inpainting is to recover the missing region from the incomplete data observed. Ideally, the restored image should possess shapes and patterns consistent with the given data in human vision. Therefore we need to extract information such as edges and texture from the observed data to replace the corrupted part in such a way that it would look natural for human eyes. Many useful techniques have been proposed in recent years to address the problem, see, for example, [1, 2, 3, 9, 11, 12, 13, 14, 15, 26, 27, 29, 38].

The mathematical model for image inpainting can be stated as follows. We will denote images as vectors in $\mathbb{R}^N$ by concatenating their columns. Let the original image $\mathbf{f}$ be defined on the domain $\Omega = \{1, 2, \cdots, N\}$ and the nonempty set $\Lambda \subsetneq \Omega$ be the given observed region. Then the observed (incomplete) image $\mathbf{g}$ is

$$\mathbf{g}(i) = \begin{cases} \mathbf{f}(i) + \epsilon(i), & i \in \Lambda, \\ \text{arbitrary}, & i \in \Omega \setminus \Lambda, \end{cases} \tag{1.1}$$

where $\epsilon(i)$ are the noise terms. The goal is to find $\mathbf{f}$ from $\mathbf{g}$. When $\epsilon(i) = 0$ for all $i \in \Lambda$, we require that $\mathbf{f}(i) = \mathbf{g}(i)$ and $\mathbf{f}$ is just the solution of an interpolation problem. Otherwise, we seek a smooth solution $\mathbf{f}$ that satisfies $|\mathbf{f}(i) - \mathbf{g}(i)| \leq \epsilon(i)$ for all $i \in \Lambda$. In both cases, variational approaches will penalize some cost functionals (which normally are weighted function norms of the underlying solution) to control the roughness of the solution, see for instance [2, 13, 39].

Motivated by the ideas in [7, 8, 10] for high-resolution image reconstruction problems, we proposed in [9] an iterative algorithm based on wavelet tight frames for image inpainting. The algorithm is also similar in spirit to those in [27, 29]. It first transforms the current guess into the framelet domain, then performs thresholding and transforms it back to the image domain; and finally the pixels belonging to Λ are replaced by the known data $\mathbf{g}$ on Λ to get a better approximation of $\mathbf{f}$. The algorithm is efficient and gives 2 to 3 dB improvement in PSNR when compared to variational approaches such as those given in [11, 12]. In this paper, we simplify the algorithm in [9] to obtain

*Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. E-mail: `tslcaij@nus.edu.sg`. Present affiliation: Temasek Laboratories and Department of Mathematics, National University of Singapore.

†Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong. E-mail: `rchan@math.cuhk.edu.hk`. This work was supported by HKRGC Grant CUHK 400405 and CUHK DAG 2060257.

‡Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543. Email: `matzuows@nus.edu.sg`.This work was partially supported by Grant R-146-000-060-112 at the National University of Singapore.

a new but similar algorithm. By interpreting the new algorithm as an iteration for minimizing a special functional, we prove its convergence under Combettes and Wajs's framework of the proximal forward-backward splitting [16]. We also show that the minimization problem that it converges to is related to the Besov norm of the underlying solution. Additionally, in the wavelet coefficient domain, our method minimizes a functional with a weighted ℓ_1 regularization term. Therefore, our method is related to the sparse approximation of images by wavelets. The ability to approximate images sparsely is an important characteristic of wavelets, see [33]. Inpainting algorithms related to ℓ_1 minimization were also proposed in, e.g., [18, 26, 27, 29, 38]. We will compare some of them with our method. We remark that the technique we used can also be adapted to prove the convergence of our algorithm in [9] under additional assumptions on the tight frames.

The outline of the paper is as follows. In Section 2, we give some preliminaries of framelets and examples of framelets used in this paper. In Section 3, we introduce our framelet-based inpainting algorithm and show that it is equivalent to an alternate direction minimization. In Section 4, we prove the convergence of the algorithm and the existence of the minimizers using the theory in [16]. In Section 5, we explain why our algorithm works and its relationship with other wavelet-based methods. In Section 6, we compare our method with other methods numerically. Conclusions are given in Section 7. Finally in Section 8, we provide the convergence proof of the algorithm in [9].

2. Framelets. In this section, we present some preliminaries of tight framelets. For simplicity, we only present the univariate framelets and the framelets for two variables can be constructed by tensor product of univariate framelets. Tight frames in finite dimensional space derived from framelets and their matrix forms are also given. Those who are familiar with the notions of framelets may skip most of this section.

2.1. Framelets in $L_2(\mathbb{R})$. A system $X \subset L_2(\mathbb{R})$ is called a *tight frame* of $L_2(\mathbb{R})$ if

$$f = \sum_{g \in X} \langle f, g \rangle g, \quad \forall f \in L_2(\mathbb{R}). \tag{2.1}$$

This is equivalent to

$$\|f\|_2^2 = \sum_{g \in X} |\langle f, g \rangle|^2, \quad \forall f \in L_2(\mathbb{R}), \tag{2.2}$$

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, since the identities (2.1) and (2.2) hold for arbitrary orthonormal bases in $L_2(\mathbb{R})$. Hence tight frames are generalization of orthonormal bases that bring in the redundancy which is often useful in applications such as denoising, see e.g. [17].

Recall that a wavelet (or affine) system $X(\Psi)$ is defined to be the collection of dilations and shifts of a finite set $\Psi \subset L^2(\mathbb{R})$, i.e.,

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

and the elements in Ψ are called the *generators*. When $X(\Psi)$ is also a tight frame for $L_2(\mathbb{R})$, then $\psi \in \Psi$ are called *(tight) framelets*, following the terminology used in [19]. To construct compactly supported framelet systems, one starts with a compactly supported refinable function $\phi \in L_2(\mathbb{R})$ with a refinement mask (low-pass filter) ζ_ϕ such that ϕ satisfies the refinement equation: $\widehat{\phi}(2\cdot) = \zeta_\phi \widehat{\phi}$. Here $\widehat{\phi}$ is the Fourier transform of ϕ, and ζ_ϕ is a trigonometric polynomial with $\zeta_\phi(0) = 1$. A multiresolution analysis (MRA) from this given refinable function can be formed, see [20, 32] for details.

The compactly supported framelets Ψ are defined in the Fourier domain by $\widehat{\psi}(2\cdot) = \zeta_\psi \widehat{\phi}$ for some trigonometric polynomials ζ_ψ, $\psi \in \Psi$. The unitary extension principle (UEP) of [37] asserts that the

FIG. 2.1. *Piecewise linear framelets.*
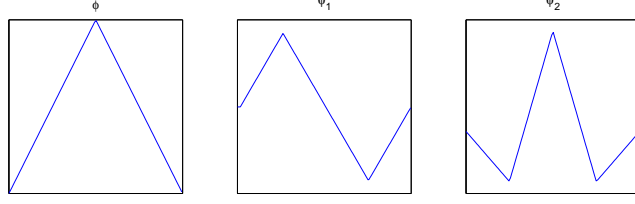
system $X(\Psi)$ generated by a finite set Ψ forms a tight frame in $L_2(\mathbb{R})$ provided that the masks ζ_ϕ and $\{\zeta_\psi\}_{\psi \in \Psi}$ satisfy:

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

for almost all ω in $\mathbb{R}$. The sequences of Fourier coefficients of ζ_ψ, as well as ζ_ψ itself, are called *framelet masks* or *high-pass filters*. The construction of framelets Ψ essentially is to design, for a given refinement mask ζ_ϕ, framelet masks $\{\zeta_\psi\}_{\psi \in \Psi}$ such that (2.3) holds. For a given ϕ with refinement mask ζ_ϕ, as shown in [19, 22], it is easy to construct ζ_ψ, $\psi \in \Psi$, whenever ζ_ϕ satisfies

$$|\zeta_\phi|^2 + |\zeta_\phi(\cdot + \pi)|^2 \leq 1.$$

Furthermore, the framelets can be constructed to be symmetric as long as ϕ is symmetric. In particular, one can construct tight framelet systems from B-splines. Here, we give two examples which will be used in our numerical simulations.

The first example is derived from piecewise linear B-spline whose refinement mask is $h_0 = \frac{1}{4}[1, 2, 1]$. The two corresponding framelet masks are

$$h_1 = \frac{\sqrt{2}}{4}[1, 0, -1], \quad h_2 = \frac{1}{4}[-1, 2, -1].$$

The associated refinable function and framelets are given in Fig. 2.1. The second example is from piecewise cubic B-spline whose refinement mask is $h_0 = \frac{1}{16}[1, 4, 6, 4, 1]$. The four framelet masks are

$$h_1 = \frac{1}{8}[1, 2, 0, -2, -1], \quad h_2 = \frac{\sqrt{6}}{16}[-1, 0, 2, 0, -1], \quad h_3 = \frac{1}{8}[-1, 2, 0, -2, 1], \quad h_4 = \frac{1}{16}[1, -4, 6, -4, 1].$$

The associated refinable function and framelets are given in Fig. 2.2. According to our experience, in actual implementation, we choose piecewise linear framelet for the simplicity of the masks and speed, or piecewise cubic when the smoothness of the framelet is desirable. Construction of tight framelets from B-splines of high orders can be found in [37].

The refinement and framelet masks can be used to derive fast decomposition and reconstruction algorithms similar to the orthonormal wavelet case as detailed in [19], see also [6].

2.2. Frames in $\mathbb{R}^N$. Since images are finite dimensional, we describe briefly here how to convert the framelet decomposition and reconstruction to finite dimension frames. Let $\mathcal{A}$ be a K-by-N ($K \geq N$) matrix whose rows are vectors in $\mathbb{R}^N$. The system, denoted by $\mathcal{A}$ again, consisting of all the rows of $\mathcal{A}$, is a tight frame for $\mathbb{R}^N$ if for any arbitrary vector $\mathbf{x}$ in $\mathbb{R}^N$

$$\|\mathbf{x}\|_2^2 = \sum_{\mathbf{y} \in \mathcal{A}} |\langle \mathbf{x}, \mathbf{y}\rangle|^2.$$

3
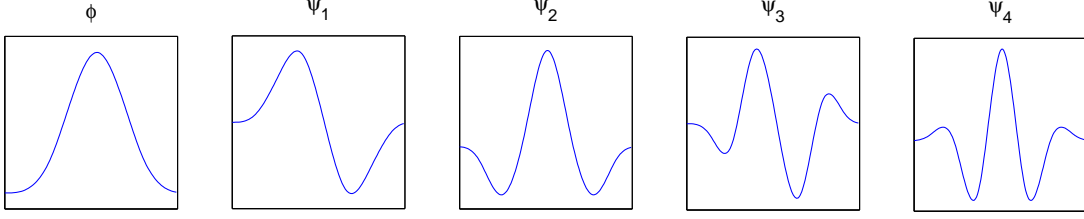
Here and in the following, if not specified, $\langle \cdot, \cdot \rangle$ and $\| \cdot \|_2 = \langle \cdot, \cdot \rangle^{1/2}$ are the inner product and norm of finite dimensional Euclidean spaces. We note that the above equation is equivalent to the perfect reconstruction formula $\mathbf{x} = \sum_{\mathbf{y} \in \mathcal{A}} \langle \mathbf{x}, \mathbf{y} \rangle \mathbf{y}$. The matrix $\mathcal{A}$ is called the *analysis (or decomposition) operator*, and its adjoint $\mathcal{A}^*$ is called the *synthesis (or reconstruction) operator*. The *perfect reconstruction formula* can be rewritten as $\mathbf{x} = \mathcal{A}^*\mathcal{A}\mathbf{x}$. Hence $\mathcal{A}$ is a tight frame if and only if $\mathcal{A}^*\mathcal{A} = \mathcal{I}$. Unlike the orthonormal case, we emphasize that $\mathcal{A}\mathcal{A}^* \neq \mathcal{I}$ in general.

In the following, we derive $\mathcal{A}$ from the masks associated with any given framelet system. For a given filter $h = \{h(j)\}_{j=-J}^{J}$, let the matrix $\mathcal{S}(h)$ be the convolution operator with filter h under the Neumann (symmetric) boundary condition:

$$
\mathcal{S}(h) = \begin{bmatrix} h(0) & \cdots & h(-J) & & 0 \\ \vdots & \ddots & \ddots & \ddots & \\ h(J) & \ddots & \ddots & \ddots & h(-J) \\ & \ddots & \ddots & \ddots & \vdots \\ 0 & & h(J) & \cdots & h(0) \end{bmatrix} + \begin{bmatrix} h(1) & h(2) & \cdots & h(J) & 0 \\ h(2) & \iddots & \iddots & \iddots & h(-J) \\ \vdots & \iddots & \iddots & \iddots & \vdots \\ h(J) & & \iddots & \iddots & h(-2) \\ 0 & h(-J) & \cdots & h(-2) & h(-1) \end{bmatrix}, \quad (2.4)
$$

which is a Toeplitz-plus-Hankel matrix, see [8, 9, 36] for details. In our numerical simulations in Section 6, we use a multi-level tight frame system corresponding to the tight framelet decomposition without down sampling. This is the same as the decomposition under the quasi-affine system introduced in [37] which is called the *undecimated framelet*. To introduce it, we recall that for a given filter $h = \{h(j)\}_{j=-J}^{J}$, the filters $h^{(\ell)}$ at level ℓ corresponding to the decomposition without down sampling is

$$
h^{(\ell)} = \{h(-J), \underbrace{0, \cdots, 0}_{2^{\ell-1}-1}, h(-J+1), 0, \cdots, \cdots, 0, h(-1), \underbrace{0, \cdots, 0}_{2^{\ell-1}-1}, h(0),
$$
$$
\underbrace{0, \cdots, 0}_{2^{\ell-1}-1}, h(1), 0, \cdots, \cdots, 0, h(J-1), \underbrace{0, \cdots, 0}_{2^{\ell-1}-1}, h(J)\}.
$$

Given the low and high-pass filters $\{h_i\}_{i=0}^{r}$, let $\mathcal{H}_i^{(\ell)} \equiv \mathcal{S}(h_i^{(\ell)})$. Then the multi-level decomposition

4

operator up to level L (without down-sampling) is given by

$$
\mathcal{A} = \begin{bmatrix}
\prod_{\ell=0}^{L-1} \mathcal{H}_0^{(L-\ell)} \\
\mathcal{H}_1^{(L)} \prod_{\ell=1}^{L-1} \mathcal{H}_0^{(L-\ell)} \\
\vdots \\
\mathcal{H}_r^{(L)} \prod_{\ell=1}^{L-1} \mathcal{H}_0^{(L-\ell)} \\
\vdots \\
\vdots \\
\mathcal{H}_1^{(1)} \\
\vdots \\
\mathcal{H}_r^{(1)}
\end{bmatrix} := \begin{bmatrix} \mathcal{A}_0 \\ \mathcal{A}_1 \end{bmatrix}, \tag{2.5}
$$

where $\mathcal{A}_0 = \prod_{\ell=0}^{L-1} \mathcal{H}_0^{(L-\ell)}$, and $\mathcal{A}_1$ consists of the remaining blocks of $\mathcal{A}$.

The unitary extension principle asserts that

$$
\mathcal{A}^* \mathcal{A} = \mathcal{A}_0^* \mathcal{A}_0 + \mathcal{A}_1^* \mathcal{A}_1 = \mathcal{I}. \tag{2.6}
$$

Then, up to a dilation, the discrete image $\mathbf{f}$ is the coefficients $\mathbf{f}(i) = \langle f, \phi(\cdot - i) \rangle$, where ϕ is the refinable function associated with the framelet system, and $\langle \cdot, \cdot \rangle$ is the inner product in $L_2(\mathbb{R})$. The sub-matrix $\mathcal{A}_0$ represents the low-pass filter, i.e., $\mathcal{A}_0 \mathbf{f}$ are the coefficients $\langle f, 2^{-L/2} \phi(2^{-L} \cdot -j) \rangle$ at a prescribed coarsest level L. The sub-matrix $\mathcal{A}_1$ represents band-pass and high-pass filters, i.e., $\mathcal{A}_1 \mathbf{f}$ are the framelet coefficients $\langle f, 2^{-l/2} \psi_i(2^{-l} \cdot -j) \rangle$, $1 \leq i \leq r$, for $0 \leq l \leq L$. We denote the numbers of rows in the sub-matrices $\mathcal{A}_0$ and $\mathcal{A}_1$ to be K_0 and K_1 respectively. It is clear that $K_0 + K_1 = K$.

3. Framelet-based Inpainting Algorithm. In this section, we introduce our framelet-based inpainting algorithm and show that it is equivalent to an alternate direction minimization procedure.

3.1. Inpainting Algorithm. Let $\mathcal{P}_\Lambda$ be the diagonal matrix with diagonal entries 1 for the indices in Λ and 0 otherwise. The starting point of our framelet inpainting algorithm is the identities (2.6) and

$$
\mathbf{f} = \mathcal{P}_\Lambda \mathbf{f} + (\mathcal{I} - \mathcal{P}_\Lambda) \mathbf{f}. \tag{3.1}
$$

Substituting the known data $\mathcal{P}_\Lambda \mathbf{g} = \mathcal{P}_\Lambda \mathbf{f}$ and (2.6) into (3.1), we obtain

$$
\mathbf{f} = \mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda) \mathcal{A}^* \mathcal{A} \mathbf{f}.
$$

Thus, the most straightforward way for framelet inpainting is to iterate as

$$
\mathbf{f}_{n+1} = \mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda) \mathcal{A}^* \mathcal{A} \mathbf{f}_n. \tag{3.2}
$$

But this iteration converges in one step for any initial guess, i.e., $\mathbf{f}_{n+1} = \mathbf{f}_n$ for all $n \geq 1$. The information contained in Λ does not propagate into $\Omega \setminus \Lambda$.

Since images that can be modeled as piecewise smooth functions have sparse approximation via a nonlinear approximation scheme under the framelet system, see [4, 31], it motivates us to incorporate into (3.2) a thresholding operator $\mathcal{T}_\lambda$ to obtain the framelet inpainting algorithm:

$$
\mathbf{f}_{n+1} = \mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda) \mathcal{A}^* \mathcal{T}_\lambda(\mathcal{A} \mathbf{f}_n).
$$

Here

$$\mathcal{T}_\lambda([\beta_1, \beta_2, \ldots, \beta_K]^T) \equiv [t_{\lambda_1}(\beta_1), t_{\lambda_2}(\beta_2), \ldots, t_{\lambda_K}(\beta_K)]^T \tag{3.3}$$

with $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_K]^T$, and $t_{\lambda_i}(\cdot)$ is the soft-thresholding function [23]:

$$t_{\lambda_i}(\beta_i) \equiv \begin{cases} \operatorname{sgn}(\beta_i)(|\beta_i| - \lambda_i), & \text{if } |\beta_i| > \lambda_i, \\ 0, & \text{if } |\beta_i| \leq \lambda_i. \end{cases} \tag{3.4}$$

Note that by using the soft-thresholding instead of the hard-thresholding normally used in nonlinear approximation scheme, we can obtain the desire minimization property in each iteration as we will see in Lemma 3.1. Besides, the thresholding operator $\mathcal{T}_\lambda$ also plays two other important roles, namely, removing noises in the image and perturbing the frame coefficients $\mathcal{A}\mathbf{f}_n$ so that information contained in the given region can permeate into the missing region.

Let the thresholding parameters be

$$\lambda = (\nu_1, \nu_2, \cdots, \nu_K)^T, \tag{3.5}$$

where $\nu_i > 0$ for $i = 1, \cdots, K$. The whole algorithm is given as follows:

ALGORITHM 1.

(i) *Set an initial guess* $\mathbf{f}_0$.

(ii) *Iterate on n until convergence:*

$$\mathbf{f}_{n+1} = \mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^* \mathcal{T}_\lambda(\mathcal{A}\mathbf{f}_n). \tag{3.6}$$

(iii) *Let* $\mathbf{f}^\star$ *to the output of Step (ii). If* $\epsilon(i) = 0$ *for all* $i \in \Lambda$ *in (1.1), we set* $\mathbf{f}^\star$ *to be the solution (to the interpolation problem); otherwise, since* $\mathcal{T}_\lambda$ *can remove the noise, we set* $\mathbf{f}^\diamond = \mathcal{A}^* \mathcal{T}_\lambda(\mathcal{A}\mathbf{f}^\star)$ *to be the solution (to the inpainting-plus-denoising problem).*

We remark that Algorithm 1 was first proposed in [9], where we required $\nu_i = 0$, for all $1 \leq i \leq K_0$ in (3.5), i.e., we did not threshold the low-pass framelet coefficients. This is a standard practice in image denoising, since this component contains no noise after passing a very long low-pass filter $\mathcal{A}_0$. However, since the thresholding here also plays a role of perturbing the framelet coefficients to allow information flow, thresholding the low-pass component can help to permeate the information. Furthermore, as the original data will be placed back at each iteration, the loss of the low frequency information in the original data can be placed back. As we will see in the numerical tests in Section 6, there are very small differences in the inpainted results between the two different choices of thresholding parameters. However, the proof of the convergence of Algorithm 1 is much simpler, and will be given in the next section. The convergence proof of the algorithm in [9] requires additional assumptions on $\mathcal{A}$. For completeness, we also provide the convergence proof of the algorithm in [9], see Section 8.

We now re-formulate Algorithm 1 as an iteration for minimizing a special functional. Then we prove its convergence in Section 4.

3.2. Algorithm 1 as Alternate Direction Minimization. In this subsection, we show that (3.6) in Algorithm 1 is equivalent to an alternate direction minimization procedure. To see this, we start with two minimization problems. The first one is the well-known equivalence between the soft-thresholding and a minimization of a functional.

LEMMA 3.1. *The soft-thresholding operator* $\mathcal{T}_\lambda$ *defined by (3.3) satisfies*

$$\mathcal{T}_\lambda(\beta) = \arg \min_{\alpha \in \mathbb{R}^K} \{\frac{1}{2}\|\beta - \alpha\|_2^2 + \|\operatorname{diag}(\lambda)\alpha\|_1\}, \tag{3.7}$$

where $\beta \in \mathbb{R}^K$, $\lambda \in \mathbb{R}_+^K$ and $\operatorname{diag}(\lambda)$ *is a diagonal matrix with the diagonal being the vector* λ.

Proof. It follows from [16] that t_{λ_i} defined in (3.4) satisfies

$$t_u(b) = \arg\min_{a \in \mathbb{R}}\{\frac{1}{2}(b-a)_2^2 + u|a|\}, \quad b, u \in \mathbb{R}, \text{ and } u \geq 0. \tag{3.8}$$

Since the minimization problem in (3.7) can be decoupled into disjoint 1-D minimization problems of the form (3.8), $\mathcal{T}_\lambda(\beta)$ defined in (3.3) is a minimizer of the minimization problem in (3.7). □

The second minimization problem is the projection of a vector onto the set:

$$\mathbf{C} = \{\mathbf{y} : \mathcal{P}_\Lambda \mathbf{y} = \mathcal{P}_\Lambda \mathbf{g}\}.$$

The set $\mathbf{C}$ is convex since it is the inverse image of $\mathcal{P}_\Lambda g$ by $\mathcal{P}_\Lambda$. The projection of a vector $\mathbf{x}$ onto $\mathbf{C}$, denoted by $\mathcal{P}_{\mathbf{C}}(\mathbf{x})$, can be defined as the minimizer of the constrained optimization problem:

$$\mathcal{P}_{\mathbf{C}}(\mathbf{x}) = \arg\min_{\mathbf{y} \in \mathbf{C}} \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2.$$

LEMMA 3.2. *The projection* $\mathcal{P}_{\mathbf{C}}(\mathbf{x})$ *satisfies*
(a) $\mathcal{P}_{\mathbf{C}}(\mathbf{x}) = \mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathbf{x}$, *and*
(b) $\mathcal{P}_{\mathbf{C}}(\mathbf{x}) = \arg\min_{\mathbf{y}}\{\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \iota_{\mathbf{C}}(\mathbf{y})\}$, *where* $\iota_{\mathbf{C}}$ *is the indicator function of* $\mathbf{C}$ *defined by*

$$\iota_{\mathbf{C}}(\mathbf{y}) \equiv \begin{cases} 0, & \mathbf{y} \in \mathbf{C}, \\ +\infty, & \mathbf{y} \notin \mathbf{C}. \end{cases} \tag{3.9}$$

Proof. For part (a), we note that for any vector $\mathbf{z} \in \mathbf{C}$, we have

$$\|\mathbf{z} - \mathbf{x}\|_2^2 = \sum_{i \in \Lambda}(z_i - x_i)^2 + \sum_{i \in \Omega \setminus \Lambda}(z_i - x_i)^2 = \sum_{i \in \Lambda}(g_i - x_i)^2 + \sum_{i \in \Omega \setminus \Lambda}(z_i - x_i)^2$$

$$\geq \sum_{i \in \Lambda}(g_i - x_i)^2 = \|\mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathbf{x} - \mathbf{x}\|_2^2.$$

On the other hand, we note that

$$\mathcal{P}_\Lambda(\mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathbf{x}) = \mathcal{P}_\Lambda \mathbf{g}.$$

It implies that the vector $\mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathbf{x} \in \mathbf{C}$. Therefore, the closest vector to $\mathbf{x}$ in $\mathbf{C}$ is $\mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathbf{x}$. Part (b) is straightforward by the definition of $\iota_{\mathbf{C}}$. □

We now rewrite (3.6) in Algorithm 1 using the above results. By Lemma 3.1, we have

$$\alpha_n \equiv \mathcal{T}_\lambda(\mathcal{A}\mathbf{f}_n) = \arg\min_\alpha\{\frac{1}{2}\|\mathcal{A}\mathbf{f}_n - \alpha\|_2^2 + \|\operatorname{diag}(\lambda)\alpha\|_1\}.$$

Substituting the definition of α_n into (3.6) and using Lemma 3.2, (3.6) can be rewritten as

$$\mathbf{f}_{n+1} = \mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^* \alpha_n = \mathcal{P}_{\mathbf{C}}(\mathcal{A}^* \alpha_n) = \arg\min_{\mathbf{f}}\{\frac{1}{2}\|\mathcal{A}^* \alpha_n - \mathbf{f}\|_2^2 + \iota_{\mathbf{C}}(\mathbf{f})\}.$$

Hence we see that our framelet-based inpainting algorithm (3.6) can be reformulated as a frequency-spatial alternate minimization:

$$\begin{cases} \alpha_n = \arg\min_\alpha\{\frac{1}{2}\|\mathcal{A}\mathbf{f}_n - \alpha\|_2^2 + \|\operatorname{diag}(\lambda)\alpha\|_1\}, \\ \mathbf{f}_{n+1} = \arg\min_{\mathbf{f}}\{\frac{1}{2}\|\mathcal{A}^* \alpha_n - \mathbf{f}\|_2^2 + \iota_{\mathbf{C}}(\mathbf{f})\}. \end{cases} \tag{3.10}$$

This shows that the sequences $\{\mathbf{f}_n\}$ and $\{\alpha_n\}$ are uniquely determined by each other. Furthermore, $\{\mathbf{f}_n\}$ converges if and only if $\{\alpha_n\}$ does, because the operators $\mathcal{T}_\lambda \mathcal{A}$ and $\mathcal{P}_{\mathbf{C}} \mathcal{A}^*$ are all continuous.

4. Convergence Analysis of Algorithm 1. This section is devoted to showing the convergence of Algorithm 1, i.e., both the sequences $\{\mathbf{f}_n\}$ and $\{\alpha_n\}$ in (3.10) converge. We show in Section 4.1 that $\lim_{n\to\infty}\{\mathbf{f}_n\} \equiv \mathbf{f}^\star$ exists and is a minimizer of

$$\min_{\mathbf{f}\in\mathbf{C}}\{\min_{\alpha}\{\frac{1}{2}\|\mathcal{A}\mathbf{f} - \alpha\|_2^2 + \|\mathrm{diag}(\lambda)\alpha\|_1\}\}. \tag{4.1}$$

In Section 4.2, we show that $\alpha^\diamond \equiv \mathcal{T}_\lambda(\mathcal{A}\mathbf{f}^\star)$ is a minimizer of

$$\min_{\alpha}\{\frac{1}{2}\|\mathcal{P}_\Lambda(\mathcal{A}^*\alpha) - \mathcal{P}_\Lambda\mathbf{g}\|_2^2 + \frac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha\|_2^2 + \|\mathrm{diag}(\lambda)\alpha\|_1\}. \tag{4.2}$$

4.1. Convergence of $\mathbf{f}_n$. In this subsection, we show that the iteration (3.6) in Algorithm 1, which is equivalent to (3.10), converges to a minimizer of (4.1). The proof uses proximal forward-backward splitting in convex analysis developed in [16].

Let us recall two basic definitions by Moreau in convex analysis [30, 34, 35]. For any proper, convex, lower semi-continuous function φ which takes its values in $(-\infty, +\infty]$, its *proximal operator* is defined by

$$\mathrm{prox}_\varphi(\mathbf{x}) \equiv \arg\min_{\mathbf{y}}\{\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \varphi(\mathbf{y})\}, \tag{4.3}$$

and its *envelope* is a function defined by

$${}^1\varphi(\mathbf{x}) \equiv \min_{\mathbf{y}}\{\frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2 + \varphi(\mathbf{y})\}. \tag{4.4}$$

By Lemma 2.5 in [16], the function ${}^1\varphi(\mathbf{x})$ is convex and differentiable, and its gradient is

$$\nabla\left({}^1\varphi(\mathbf{x})\right) = \mathbf{x} - \mathrm{prox}_\varphi(\mathbf{x}). \tag{4.5}$$

We now state the main convergence theorem in [16] for the finite dimensional case.

THEOREM 4.1. *Consider the minimization problem*

$$\min_{\mathbf{f}}\{F_1(\mathbf{f}) + F_2(\mathbf{f})\}, \tag{4.6}$$

where F_1 with range $(-\infty, +\infty]$ is a proper, convex, lower semi-continuous function, and F_2 with range in $\mathbb{R}$ is a proper, convex, differentiable function with a $1/b$-Lipschitz continuous gradient. Assume a minimizer to (4.6) exists and $b > 1/2$. Then for any initial guess $\mathbf{f}_0$, the iteration (called the proximal forward-backward splitting):

$$\mathbf{f}_{n+1} = \mathrm{prox}_{F_1}(\mathbf{f}_n - \nabla F_2(\mathbf{f}_n)) \tag{4.7}$$

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

To apply the theorem, we define $\xi(\beta) \equiv \|\mathrm{diag}(\lambda)\beta\|_1$, $F_1(\mathbf{f}) \equiv \iota_{\mathbf{C}}(\mathbf{f})$, and $F_2(\mathbf{f}) \equiv ({}^1\xi \circ \mathcal{A})(\mathbf{f})$. By (3.9) and (4.4),

$$\min_{\mathbf{f}}\{F_1(\mathbf{f}) + F_2(\mathbf{f})\} = \min_{\mathbf{f}\in\mathbf{C}}\{({}^1\xi \circ \mathcal{A})(\mathbf{f})\} = \min_{\mathbf{f}\in\mathbf{C}}\{{}^1\|\mathrm{diag}(\lambda)(\mathcal{A}\mathbf{f})\|_1\}$$

$$= \min_{\mathbf{f}\in\mathbf{C}}\{\min_{\alpha}\{\frac{1}{2}\|\mathcal{A}\mathbf{f} - \alpha\|_2^2 + \|\mathrm{diag}(\lambda)\alpha\|_1\}\}, \tag{4.8}$$

8

which is just the minimization problem (4.1). Note that both F_1 and F_2 are proper, convex and lower semi-continuous. Substituting (4.3) in (3.10) and using $\mathcal{A}^*\mathcal{A} = \mathcal{I}$, Algorithm 1 becomes:

$$\mathbf{f}_{n+1} = \text{prox}_{\iota_{\mathbf{C}}}(\mathcal{A}^*\alpha_n) \tag{4.9}$$
$$= \text{prox}_{\iota_{\mathbf{C}}}(\mathcal{A}^*\text{prox}_\xi(\mathcal{A}\mathbf{f}_n))$$
$$= \text{prox}_{\iota_{\mathbf{C}}}(\mathbf{f}_n - \mathcal{A}^*\mathcal{A}\mathbf{f}_n + \mathcal{A}^*\text{prox}_\xi(\mathcal{A}\mathbf{f}_n))$$
$$= \text{prox}_{\iota_{\mathbf{C}}}(\mathbf{f}_n - \mathcal{A}^*(\mathcal{A}\mathbf{f}_n - \text{prox}_\xi(\mathcal{A}\mathbf{f}_n))). \tag{4.10}$$

Since by (4.5) and the chain rule,

$$\nabla F_2(\mathbf{f}) = \nabla(^1\xi \circ \mathcal{A})(\mathbf{f}) = \mathcal{A}^*(\mathcal{A}\mathbf{f} - \text{prox}_\xi(\mathcal{A}\mathbf{f})), \tag{4.11}$$

clearly (4.10) is just the forward-backward splitting (4.7). Thus to show that (4.10), which is equivalent to Algorithm 1, converges to (4.1), we need to show that the conditions in Theorem 4.1 hold.

LEMMA 4.2. *The functions $\iota_{\mathbf{C}}$ with range in $(-\infty, +\infty]$ and $^1\xi \circ \mathcal{A}$ with range in $\mathbb{R}$ are proper, convex, and lower semi-continuous. Moreover, the latter function is differentiable with a 1-Lipschitz continuous gradient.*

Proof. Since $\mathbf{C}$ is a closed non-empty convex set, it is obvious that $\iota_{\mathbf{C}}(\mathbf{f})$ takes its values in $(-\infty, +\infty]$ and is a proper, convex, and lower semi-continuous function. Since $\xi(\beta)$ has range in $\mathbb{R}$, by its definition so is $^1\xi(\mathcal{A}\mathbf{f})$. Because ξ is proper, convex, and lower semi-continuous, by Lemma 2.5 in [16], $^1\xi \circ \mathcal{A}$ is convex and differentiable, and its gradient is given in (4.11). It remains to prove that the gradient is 1-Lipschitz continuous. To this end, we note that by Lemma 2.4 in [16],

$$\|(\mathbf{x} - \text{prox}_\varphi(\mathbf{x})) - (\mathbf{y} - \text{prox}_\varphi(\mathbf{y}))\|_2 \le \|\mathbf{x} - \mathbf{y}\|_2$$

for any convex and lower semi-continuous φ. Applying this observation and (4.11), we have

$$\|\nabla(^1\xi \circ \mathcal{A})(\mathbf{x}) - \nabla(^1\xi \circ \mathcal{A})(\mathbf{y})\|_2 = \|\mathcal{A}^*(\mathcal{A}\mathbf{x} - \text{prox}_\xi(\mathcal{A}\mathbf{x})) - \mathcal{A}^*(\mathcal{A}\mathbf{y} - \text{prox}_\xi(\mathcal{A}\mathbf{y}))\|_2$$
$$\le \|\mathcal{A}^*\|_2\|(\mathcal{A}\mathbf{x} - \text{prox}_\xi(\mathcal{A}\mathbf{x})) - (\mathcal{A}\mathbf{y} - \text{prox}_\xi(\mathcal{A}\mathbf{y}))\|_2$$
$$\le \|\mathcal{A}^*\|_2\|\mathcal{A}(\mathbf{x} - \mathbf{y})\|_2 \le \|\mathbf{x} - \mathbf{y}\|_2,$$

which says that $^1\xi \circ \mathcal{A}$ has a 1-Lipschitz continuous gradient. □

It remains to show the existence of a minimizer of $\min_{\mathbf{f}}\{F_1(\mathbf{f}) + F_2(\mathbf{f})\}$.

LEMMA 4.3. *Let $\mathcal{A}$ be a tight frame system. Then the minimization problem (4.1) has at least one minimizer.*

Proof. By Proposition 3.1(i) in [16], a minimizer of $\min_{\mathbf{f}}\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\}$ exists if $\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\}$ is coercive, i.e., whenever $\|\mathbf{f}\|_2 \to +\infty$, $\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} \to +\infty$. To prove this, let $\nu_{\min} = \min_{i=1}^K \nu_i$ and $\nu_{\max} = \max_{i=1}^K \nu_i$, where ν_i are defined in (3.5). By the definition of ξ and (4.4),

$$\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} \ge {}^1\xi(\mathcal{A}\mathbf{f}) = {}^1\|\text{diag}(\lambda)(\mathcal{A}\mathbf{f})\|_1 = \min_\alpha\{\frac{1}{2}\|\mathcal{A}\mathbf{f} - \alpha\|_2^2 + \|\text{diag}(\lambda)\alpha\|_1\}. \tag{4.12}$$

By Lemma 3.1, the minimizer is precisely $\mathcal{T}_\lambda(\mathcal{A}\mathbf{f})$. Therefore,

$$\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} \ge \frac{1}{2}\|\mathcal{A}\mathbf{f} - \mathcal{T}_\lambda(\mathcal{A}\mathbf{f})\|_2^2 + \|\text{diag}(\lambda)\mathcal{T}_\lambda(\mathcal{A}\mathbf{f})\|_1$$
$$\ge \|\text{diag}(\lambda)\mathcal{T}_\lambda(\mathcal{A}\mathbf{f})\|_1 = \sum_{i=1}^K \nu_i|t_{\nu_i}(\mathcal{A}\mathbf{f})_i| \ge \nu_{\min}\sum_{i=1}^K |t_{\nu_i}(\mathcal{A}\mathbf{f})_i|$$
$$\ge \nu_{\min}\sum_{i=1}^K (|(\mathcal{A}\mathbf{f})_i| - \nu_i) \ge \nu_{\min}\|\mathcal{A}\mathbf{f}\|_1 - \nu_{\min}\nu_{\max}K$$

9

$$\geq \nu_{\min}\|\mathcal{A}\mathbf{f}\|_2 - \nu_{\min}\nu_{\max}K = \nu_{\min}\|\mathbf{f}\|_2 - \nu_{\min}\nu_{\max}K.$$

Clearly if $\|\mathbf{f}\|_2 \to +\infty$, then $\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} \to +\infty$. □

Combining everything together, we have the convergence of Algorithm 1.

THEOREM 4.4. *Let $\mathcal{A}$ be a tight frame system. Then iteration* (3.6) *in Algorithm 1 converges to a minimizer of the minimization problem* (4.1) *for any initial guess $\mathbf{f}_0$.*

Proof. We have shown that (3.6) is equivalent to (4.7). By Theorem 4.1 and Lemmas 4.2 and 4.3, (4.7) converges to a minimizer of (4.6), which by (4.8) is just (4.1). □

This shows that Algorithm 1 converges for any tight frames. They include the tight frames derived from framelets or other wavelet masks discussed in Section 2. Furthermore, if one chooses to use the framelet or wavelet decomposition with down sampling, Algorithm 1 still converges. However, when an orthonormal wavelet is used, since it is not symmetric (except for the Haar wavelet), symmetric boundary conditions like that in (2.4) implies $\mathcal{A}^*\mathcal{A} \neq \mathcal{I}$. In that case, one can use the periodic boundary conditions instead to obtain the corresponding tight frame system $\mathcal{A}$ (see [6] for details), so that Algorithm 1 can again be applied and converges. But we note that periodic boundary conditions are usually inferior to symmetric ones, see [36].

When the soft-thresholding operator $\mathcal{T}_\lambda$ is replaced by another shrinkage operator which is a proximity operator, Algorithm 1 can still be rewritten as an alternative iteration similar to (3.10). Hence, the convergence can be proved similarly. Examples of shrinkage operators minimizing the ℓ_p-norm, $1 \leq p < 2$, of the framelet or wavelet coefficients are given in [6, 18].

4.2. Convergence of α_n. In this subsection, we show that the limit of the sequence $\{\alpha_n\} \equiv \{\mathcal{T}_\lambda(\mathcal{A}\mathbf{f}_n)\}$ is a minimizer of (4.2). We note that by (4.3) and (4.9), the iteration for α_n in (3.10) can be written as

$$\alpha_{n+1} = \text{prox}_\xi(\mathcal{A}\mathbf{f}_{n+1}) = \text{prox}_\xi[\mathcal{A}\text{prox}_{\iota_{\mathbf{C}}}(\mathcal{A}^*\alpha_n)].$$

Since by (4.9) and (3.6), $\text{prox}_{\iota_{\mathbf{C}}}(\mathcal{A}^*\alpha_n) = \mathbf{f}_{n+1} = \mathcal{P}_\Lambda\mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\alpha_n$; and $\mathcal{P}_\Lambda^2 = \mathcal{P}_\Lambda$, we obtain

$$\begin{aligned}
\alpha_{n+1} &= \text{prox}_\xi[\mathcal{A}\mathcal{P}_\Lambda\mathbf{g} + \mathcal{A}(\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\alpha_n] \\
&= \text{prox}_\xi[\alpha_n - \alpha_n + \mathcal{A}\mathcal{P}_\Lambda\mathbf{g} + \mathcal{A}\mathcal{A}^*\alpha_n - \mathcal{A}\mathcal{P}_\Lambda\mathcal{A}^*\alpha_n] \\
&= \text{prox}_\xi[\alpha_n - ((\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha_n + \mathcal{A}\mathcal{P}_\Lambda(\mathcal{P}_\Lambda\mathcal{A}^*\alpha_n - \mathcal{P}_\Lambda\mathbf{g}))].
\end{aligned}$$

Since $(\mathcal{I} - \mathcal{A}\mathcal{A}^*)$ is the projection operator onto the kernel of $\mathcal{A}^*$, $(\mathcal{I} - \mathcal{A}\mathcal{A}^*)^2 = (\mathcal{I} - \mathcal{A}\mathcal{A}^*)$. Hence

$$\begin{aligned}
\alpha_{n+1} &= \text{prox}_\xi[\alpha_n - ((\mathcal{I} - \mathcal{A}\mathcal{A}^*)^2\alpha_n + \mathcal{A}\mathcal{P}_\Lambda(\mathcal{P}_\Lambda\mathcal{A}^*\alpha_n - \mathcal{P}_\Lambda\mathbf{g}))] \\
&= \text{prox}_\xi[\alpha_n - \nabla(\tfrac{1}{2}\|\mathcal{P}_\Lambda\mathcal{A}^*\alpha_n - \mathcal{P}_\Lambda\mathbf{g}\|_2^2 + \tfrac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha_n\|_2^2)].
\end{aligned}$$

Let $F_3(\alpha) \equiv \xi(\alpha)$ and $F_4(\alpha) \equiv \frac{1}{2}\|\mathcal{P}_\Lambda\mathcal{A}^*\alpha - \mathcal{P}_\Lambda\mathbf{g}\|_2^2 + \frac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha\|_2^2$. The above iteration becomes another proximal forward-backward splitting (c.f. (4.7)):

$$\alpha_{n+1} = \text{prox}_{F_3}(\alpha_n - \nabla F_4(\alpha_n)). \tag{4.13}$$

Next we show that F_3 and F_4 also satisfy the conditions on F_1 and F_2 in Theorem 4.1 too.

LEMMA 4.5. *The function $\xi(\alpha)$ has range in $(-\infty, +\infty]$ and is proper, convex, and lower semicontinuous. The function $\frac{1}{2}\|\mathcal{P}_\Lambda\mathcal{A}^*\alpha - \mathcal{P}_\Lambda\mathbf{g}\|_2^2 + \frac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha\|_2^2$ has range in $\mathbb{R}$, and is proper, convex and differentiable. Moreover, its gradient is 1-Lipschitz continuous.*

Proof. Since both functions are norms for some affine transforms of α, they are proper, convex and continuous functions taking values in $\mathbb{R}$. Moreover, the latter one is differentiable since 2-norms

10

are differentiable. Therefore, the only thing remaining to prove is that the gradient of the latter function is 1-Lipschitz continuous. We show it by direct calculation:

$$\|\nabla[\frac{1}{2}\|\mathcal{P}_\Lambda\mathcal{A}^*\alpha - \mathcal{P}_\Lambda\mathbf{g}\|_2^2 + \frac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha\|_2^2] - \nabla[\frac{1}{2}\|\mathcal{P}_\Lambda\mathcal{A}^*\alpha' - \mathcal{P}_\Lambda\mathbf{g}\|_2^2 + \frac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha'\|_2^2]\|_2$$
$$= \|[(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha + \mathcal{A}\mathcal{P}_\Lambda(\mathcal{P}_\Lambda\mathcal{A}^*\alpha - \mathcal{P}_\Lambda\mathbf{g})] - [(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha' + \mathcal{A}\mathcal{P}_\Lambda(\mathcal{P}_\Lambda\mathcal{A}^*\alpha' - \mathcal{P}_\Lambda\mathbf{g})]\|_2$$
$$= \|(\mathcal{I} - \mathcal{A}(\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*)(\alpha - \alpha')\|_2 \le \|\mathcal{I} - \mathcal{A}(\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\|_2\|\alpha - \alpha'\|_2 \le \|\alpha - \alpha'\|_2.$$

The last inequality follows from the fact that $\|\mathcal{I} - \mathcal{A}(\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\|_2 \le 1$ since $(\mathcal{I} - \mathcal{P}_\Lambda)^2 = (\mathcal{I} - \mathcal{P}_\Lambda)$ and $\|\mathcal{A}(\mathcal{I} - \mathcal{P}_\Lambda)\| < 1$. $\square$

Now we can prove that α_n converges to a minimizer of (4.2).

THEOREM 4.6. *Let $\alpha_n = \mathcal{T}_\lambda(\mathcal{A}\mathbf{f}_n)$ in (3.6). Then α_n converges to a minimizer of the minimization problem (4.2) for any initial guess α_0.*

Proof. Since by Theorem 4.4, the sequence $\{\mathbf{f}_n\}$ converges, the sequence $\{\alpha_n\} \equiv \{\mathcal{T}_\lambda(\mathcal{A}\mathbf{f}_n)\}$ converges too. By (4.13), the limit $\alpha^\diamond$ of $\{\alpha_n\}$ satisfies

$$\alpha^\diamond = \text{prox}_{F_3}(\alpha^\diamond - \nabla F_4(\alpha^\diamond)).$$

This, together with Lemma 4.5 and Proposition 3.1(iii)(b) in [16], implies that the limit $\alpha^\diamond$ is a minimizer of $\min_\alpha\{F_3(\alpha) + F_4(\alpha)\}$, which is equivalent to saying that it is a minimizer of (4.2). $\square$

5. Remarks on Algorithm 1. Here we give some insights and remarks on Algorithm 1.

5.1. Why Algorithm 1 Is Good. Algorithm 1 can be understood as finding $\alpha^\diamond$ first. Then, if there are no noise, i.e. $\epsilon(i) = 0$ for all $i \in \Lambda$ in (1.1), the solution to the interpolation problem is given by $\mathbf{f}^\star = \mathcal{P}_\Lambda\mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\alpha^\diamond$, otherwise, the inpainting-plus-denoising solution is $\mathcal{A}^*\alpha^\diamond$. We note that $\mathcal{P}_\Lambda\mathbf{f}^\star = \mathcal{P}_\Lambda\mathbf{g}$ whereas $\|\mathcal{P}_\Lambda(\mathcal{A}^*\alpha) - \mathcal{P}_\Lambda\mathbf{g}\|_2^2$ is one of the terms being minimized in (4.2). We now explain the purpose of each term in (4.1) and (4.2).

Assume that the true solution is a piecewise smooth function, then it can be sparsely approximated in the framelet domain, see e.g. [4, 31]. Therefore, solving the problem in the framelet domain is usually more efficient. To increase the sparsity, one likes to minimize the term $\|\text{diag}(\lambda)\alpha\|_1$, the weighted ℓ_1-norm of frame coefficient of the approximation solution, among all possible solutions, see [5, 18, 24]. Therefore we have this term in (4.1) and (4.2).

We also like to choose a proper $\alpha^\diamond$ so that the roughness of the solution is under control. In other words, the penalty function should somehow also link to the true solution via some function norms. It is shown from framelet theory (see e.g. [4, 31]) that the (weighted) ℓ_1-norm of the canonical framelet coefficient sequence of a function is equivalent to its Besov norm in the space $B_{1,1}^\sigma$ under some mild conditions on the framelets. Hence, we also require $\alpha^\diamond$ to be close to some canonical coefficient sequence, i.e. to the range of $\mathcal{A}$, so that we can be sure that the (weighted) ℓ_1-norm of $\alpha^\diamond$ is approximately linked to the Besov norm of the true function. Thus we also need to penalize the distance between $\alpha^\diamond$ and the range of $\mathcal{A}$. Note that $(\mathcal{I} - \mathcal{A}\mathcal{A}^*)$ is the projection operator onto the kernel of $\mathcal{A}^*$, i.e. the orthogonal complement of the range of $\mathcal{A}$. In other words, the term $\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha\|_2^2$ in (4.2) exactly penalizes the distance of $\alpha^\diamond$ to the range of $\mathcal{A}$.

Lastly we also require $\mathcal{A}^*\alpha^\diamond$ and $\mathbf{g}$ to be close within the noise level on Λ, and hence we also have the term $\|\mathcal{P}_\Lambda(\mathcal{A}^*\alpha^\diamond) - \mathcal{P}_\Lambda\mathbf{g}\|_2^2$ in (4.2). Notice that artifacts are not only occurring because of the difference between $\mathcal{P}_\Lambda(\mathcal{A}^*\alpha^\diamond)$ and $\mathcal{P}_\Lambda\mathbf{g}$, it can also be generated from the difference between $\mathcal{A}\mathbf{f}^\star$ and $\alpha^\diamond$ in the framelet domain, and hence there is also the term $\|\mathcal{A}\mathbf{f}^\star - \alpha^\diamond\|_2^2$ in (4.1).

Putting everything together, we see that the limits of Algorithm 1, i.e. $\mathbf{f}^\star$ and $\alpha^\diamond$, balance the sparsity and closeness to the given data together with the smoothness of the solution.

11

5.2. Relation to Other Wavelet-Based Methods. From the above discussions, our method is related to the sparse representation of wavelets for images. To have sparse representation, a minimization of the 0-norm, the number of non-zero coefficients, should be sought. However, there are several reasons why $\|\mathcal{D}\alpha\|_1$, where α is the wavelet coefficient and $\mathcal{D}$ is a diagonal matrix, is a good approximation, see [5, 18, 24].

When there is no noise, the sparse representation of wavelet requires us to solve the problem

$$\begin{cases} \min_\alpha \|\mathcal{D}\alpha\|_1, \\ \text{s.t. } \mathcal{P}_\Lambda(\mathcal{A}^*\alpha) = \mathcal{P}_\Lambda \mathbf{g}. \end{cases} \tag{5.1}$$

It is the method proposed in [38], where a linear programming is applied to find the solution. When noise is present in the observed data $\mathcal{P}_\Lambda \mathbf{g}$, we replace the equality constraint in (5.1) by an inequality constraint and solve

$$\begin{cases} \min_\alpha \|\mathcal{D}\alpha\|_1, \\ \text{s.t. } \|\mathcal{P}_\Lambda(\mathcal{A}^*\alpha) - \mathcal{P}_\Lambda \mathbf{g}\|_2 \leq \sigma^2, \end{cases}$$

where σ^2 is the noise level. If we use the Lagrange multiplier method to solve it, we have

$$\min_\alpha \left\{ \|\mathcal{P}_\Lambda(\mathcal{A}^*\alpha) - \mathcal{P}_\Lambda \mathbf{g}\|_2^2 + \rho\|\mathcal{D}\alpha\|_1 \right\}, \tag{5.2}$$

where $1/\rho$ is the Lagrange multiplier. The minimization problem is solved in [27] by the iteration

$$\begin{cases} \alpha_n = \mathcal{T}_\lambda(\alpha_{n-1} + \mathcal{A}(\mathcal{P}_\Lambda \mathbf{g} - \mathcal{P}_\Lambda \mathbf{f}_n)), \\ \mathbf{f}_{n+1} = \mathcal{A}^*\alpha_n. \end{cases} \tag{5.3}$$

The above iteration is similar to the parallel shrinkage algorithm in [25]. We note that when $\mathcal{A}$ is an orthogonal basis, Algorithm 1 and (5.3) are the same and coincide with the method in [18] proposed for general linear inverse problem. It was already pointed out in [18] that the iteration for general linear inverse problem is the same as the algorithm proposed in [21], which was derived in the framework of an expectation-maximization (EM) approach to a maximum penalized likelihood solution in [28]. However, Algorithm 1 is motivated by the deblurring method in [7, 8, 10] and (5.3) is derived from EM method and sparse representation. Here we have unified the two methods from the viewpoint of minimization of functionals when $\mathcal{A}$ is a tight frame. Recall that our method (3.6) converges to a minimizer of (4.2) in the framelet domain. Comparing (4.2) with (5.2), we see that there is an extra term $\frac{1}{2}\|(\mathcal{I} - \mathcal{A}\mathcal{A}^*)\alpha\|_2^2$ in (4.2), which forces the limit of our method to be close to the range of $\mathcal{A}$. The role of the term has been explained in Section 5.1. In Section 6, we will compare our method with (5.3).

5.3. Redundancy. Tight frames are different from orthonormal wavelets because tight frames are redundant. What does the redundancy bring us here? We start with a sort of philosophical point of view on the algorithm and then give some quantitative analysis on the error being reduced. Assume that some blocks of pixels are missing in a given image and we like to solve the inpainting problem in the framelet domain as mentioned before. Since the framelets used are compactly supported, the coefficients of those framelets whose supports fall in the missing blocks are missing and the coefficients of those framelets whose supports overlap with the missing blocks are inaccurate. The main step of Algorithm 1 perturbs the frame coefficients $\mathcal{A}\mathbf{f}_n$ by thresholding so that information contained in the available coefficients will permeate into the missing framelet coefficients. Here, the redundancy

is very important, since the available coefficients contain information of the missing coefficients only if the system is redundant.

We note that applying the thresholding operator on the framelet coefficients is a very important step in Algorithm 1 in order to remove the noise and perturb the coefficients. However, it also brings in new errors and artifacts. To explain how the numerical errors and artifacts introduced by the thresholding can be reduced by the redundancy of the system $\mathcal{A}$, we take the computed solution $\mathbf{f}^\star$ as an example. Our computed solution $\mathbf{f}^\star$ that interpolates the given data satisfies

$$\mathbf{f}^\star = \mathcal{P}_\Lambda \mathbf{g} + (\mathcal{I} - \mathcal{P}_\Lambda)\mathcal{A}^*\mathcal{T}_\lambda \mathcal{A}\mathbf{f}^\star.$$

That is, on Λ, $\mathcal{A}^*\mathcal{T}_\lambda \mathcal{A}\mathbf{f}^\star$ is replaced by $\mathbf{g}$. But since $\mathcal{P}_\Lambda \mathbf{g} = \mathcal{P}_\Lambda \mathbf{f}^\star = \mathcal{P}_\Lambda \mathcal{A}^*\mathcal{A}\mathbf{f}^\star$, we are actually replacing $\mathcal{P}_\Lambda \mathcal{A}^*\mathcal{T}_\lambda \mathcal{A}\mathbf{f}^\star$ by $\mathcal{P}_\Lambda \mathcal{A}^*\mathcal{A}\mathbf{f}^\star$, which generates artifacts. Hence to reduce the artifacts, we require that the norm of

$$\mathcal{P}_\Lambda \mathcal{A}^*\mathcal{A}\mathbf{f}^\star - \mathcal{P}_\Lambda \mathcal{A}^*\mathcal{T}_\lambda \mathcal{A}\mathbf{f}^\star = \mathcal{P}_\Lambda \mathcal{A}^*(\mathcal{A}\mathbf{f}^\star - \mathcal{T}_\lambda \mathcal{A}\mathbf{f}^\star) = \mathcal{P}_\Lambda \mathcal{A}^*(\mathcal{A}\mathbf{f}^\star - \alpha^\diamond)$$

should be small.

Clearly the smaller the norm of $\mathcal{A}^*\mathbf{e} := \mathcal{A}^*(\mathcal{A}\mathbf{f}^\star - \alpha^\diamond)$ is, the smaller the artifact is. Note that the reconstruction operator $\mathcal{A}^*$ can eliminate the error components sitting in the kernel of $\mathcal{A}^*$. In fact, since $\mathcal{A}^*$ projects all sequences down to the orthogonal complement of the kernel of $\mathcal{A}^*$, which is the range of $\mathcal{A}$, the component of $\mathbf{e}$ in the kernel of $\mathcal{A}^*$ does not contribute. The redundant system reduces the errors as long as the component of $\mathbf{e}$ in the kernel of $\mathcal{A}^*$ is not zero. Since larger the kernel of $\mathcal{A}^*$ is, the more redundant the frame is. The higher the redundancy is, the more the errors reduces in general. To increase the redundancy, we use undecimated tight framelet system (i.e. no down sampling in the decomposition). In contrast, if $\mathcal{A}$ is not a redundant system but an orthonormal system, then the kernel of $\mathcal{A}^*$ is just $\{0\}$. In this case, $\|\mathcal{A}^*\mathbf{e}\| = \|\mathbf{e}\|$.

6. Numerical Comparisons. In this section, we compare Algorithm 1, which minimizes both (4.1) and (4.2), with the wavelet-based method (5.3), which minimizes (5.2). For comparisons with the variational methods, we refers the readers to [9]. Throughout the test, the framelet system we used is the 2D version of the multilevel piecewise linear or cubic B-spline tight framelet system $\mathcal{A}$ in (2.5), with different parameters L. The thresholding parameters defined in (3.5) are chosen as

$$\lambda = c \cdot (\underbrace{2^{-L/2}, \cdots, 2^{-L/2}}_{N}, \underbrace{2^{-L/2}, \cdots, 2^{-L/2}}_{2mN}, \cdots, \underbrace{2^{-\ell/2}, \cdots, 2^{-\ell/2}}_{2mN}, \cdots, \underbrace{2^{-1/2}, \cdots, 2^{-1/2}}_{2mN})^T, \quad (6.1)$$

where c is a parameter to be determined, and $m = 1$ or 2 for the piecewise linear or cubic B-spline framelet respectively. The powers of 2 inside the parentheses in (6.1) are the normalization factors from the usage of the quasi (undecimated) frame system (see e.g. [37]). When c is larger, i.e., the thresholding level is larger, the information will permeate faster. However, bigger c means there are less details in the recovered image and bigger jumps between Λ and $\Omega \setminus \Lambda$. Therefore, one has to choose proper c to compromise between the information permeating speed and the details of the recovered image. We tune c in (6.1) manually such that the solutions are the best in the sense of peak signal-to-noise ratio (PSNR). The iteration is stopped when $\|\mathbf{f}_{n+1} - \mathbf{f}_n\|_2/\|\mathbf{g}\|_2 \leq 10^{-4}$. We choose the initial guess to be the cubic interpolation of the observed image $\mathbf{g}$.

Our first test image is a 256×256 synthetic piecewise smooth image with squares, circles, straight lines, sharp edges, and smooth regions. The framelet system is the piecewise linear B-spline framelet and the coarsest decomposition level is chosen $L = 4$. Figure 6.1 shows the results when there is no noise, i.e. $\epsilon(i) = 0$ for all $i \in \Lambda$ in (1.1). Since there is no noise in the known data, the interpolation solution $\mathbf{f}^\star$ is better in the sense of PSNR than the inpainting-plus-denoising solution

FIG. 6.1. *From left to right: (a) Image covered by texts; (b) $\mathbf{f}^\star$ (PSNR = 32.54dB, iter=112); (c) $\mathbf{f}^\diamond$ (PSNR = 32.45dB, iter=112); and (d) the restoration by (5.3) (PSNR=30.56dB, iter=98).*
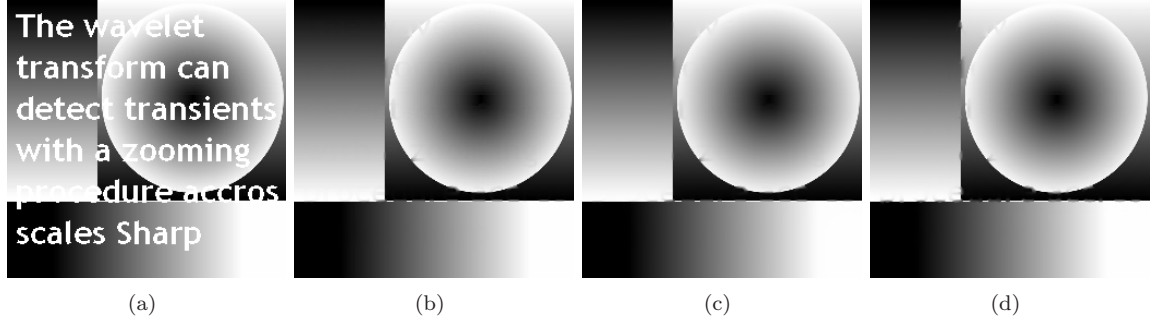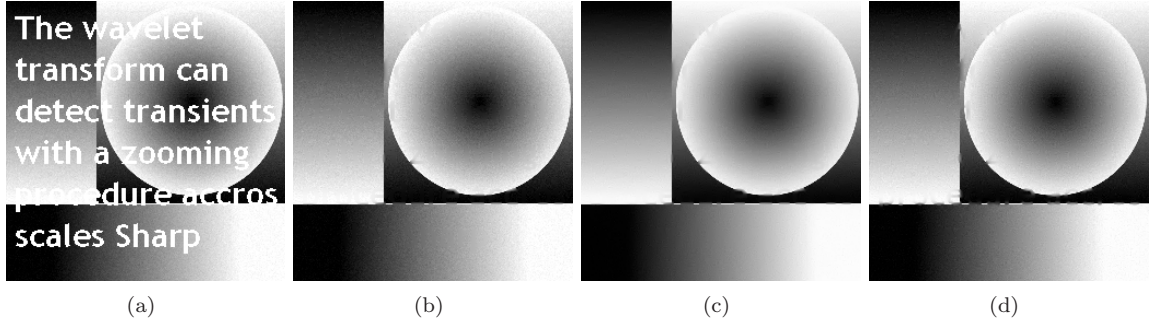

(a) (b) (c) (d)

FIG. 6.2. *From left to right: (a) Image contaminated by white Gaussian noise with standard deviation 5 (SNR=30.63dB), and covered by texts; (b) $\mathbf{f}^\star$ (PSNR = 30.46dB, iter=78); (c) $\mathbf{f}^\diamond$ (PSNR = 31.65dB, iter=78) and (d) the restoration by (5.3) (PSNR=29.44dB, iter=98).*

(a) (b) (c) (d)

$\mathbf{f}^\diamond$ (see Algorithm 1 (iii) for their definitions). Figure 6.2 shows the results when white Gaussian with standard deviation 5 is added to the observed image. In this case, since the noise ϵ is not zero and the soft-thresholding operator can remove noise, the inpainting-plus-denosing solution $\mathbf{f}^\diamond$ is better than the interpolation solution $\mathbf{f}^\star$. In fact, one can see the visual difference between them in Figure 6.2. Moreover, $\mathbf{f}^\diamond$ is also much better than the results obtained by (5.3). In both Figures 6.1 and 6.2, the vertical edge and the left part of the horizontal edge are blurred in the restoration by (5.3), but are well preserved by Algorithm 1. The numbers of iterations required to converge are also reported in the captions in the figures. We see that both algorithms converges in moderate small numbers of iterations.

Next, we compare the two methods for the real image *peppers* of size 256×256. The framelet system is the piecewise cubic B-spline framelet and the coarsest decomposition level is $L = 4$. Figure 6.3 shows the results when there is no noise, while Figure 6.4 shows the results when white Gaussian noise with standard deviation 5 is added. From the two figures, we see that for the real image, the two methods give very similar results, and Algorithm 1 is slightly better. Again, we see that both algorithms converges in moderate small numbers of iterations.

Then, we illustrate the results when the two inpainting algorithms are applied to digital zooming which are widely used in digital cameras. Here the odd-odd indexed pixels of the original image are taken to obtain the down-sampled given image. The computed results are shown in Figure 6.5. Again,

14

FIG. 6.3. *From left to right: (a) Image covered by texts; (b) $\mathbf{f}^{\star}$ (PSNR = 33.85dB, iter=80); (c) $\mathbf{f}^{\diamond}$ (PSNR = 33.85dB, iter=80); and (d) the restoration by (5.3) (PSNR=33.84dB, iter=70).*

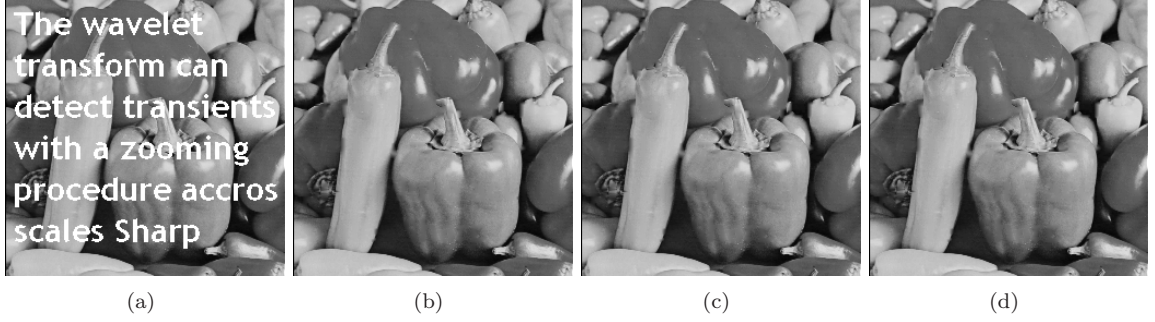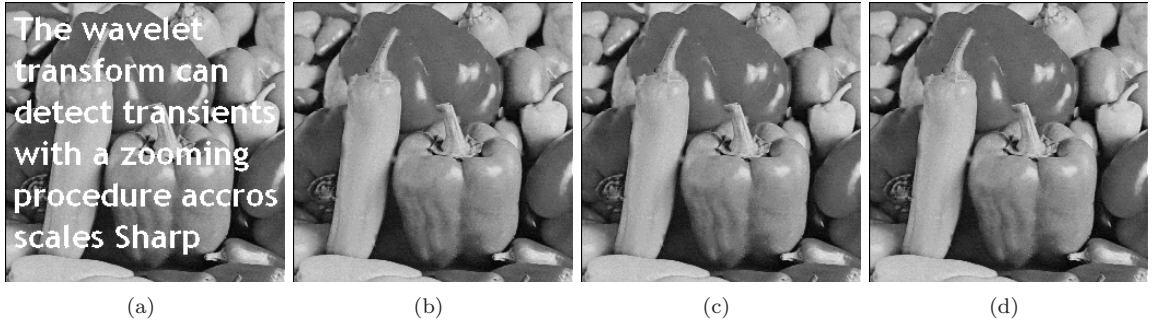(a)　　　　　　(b)　　　　　　(c)　　　　　　(d)

FIG. 6.4. *From left to right: (a) Image contaminated by white Gaussian noise with standard deviation 5 (SNR=28.60dB), and covered by texts; (b) $\mathbf{f}^{\star}$ (PSNR = 31.24dB, iter=115); (c) $\mathbf{f}^{\diamond}$ (PSNR = 31.28dB, iter=115); and (d) the restoration by (5.3) (PSNR=31.27dB, iter=84).*

(a)　　　　　　(b)　　　　　　(c)　　　　　　(d)

the two methods give similar results, but that by Algorithm 1 is slightly better. For the iteration numbers, though they are several times of those for the above examples, they are still moderate small compared to the image size.

Finally, Algorithm 1 is compared with the inpainting algorithm in [9], where the low-pass coefficients are not thresholded, i.e., the thresholding parameters are chosen as

$$\lambda = c \cdot (\underbrace{0, \cdots, 0}_{N}, \underbrace{2^{-L/2}, \cdots, 2^{-L/2}}_{2mN}, \cdots, \underbrace{2^{-\ell/2}, \cdots, 2^{-\ell/2}}_{2mN}, \cdots, \underbrace{2^{-1/2}, \cdots, 2^{-1/2}}_{2mN})^{T}. \qquad (6.2)$$

The results of the algorithm in [9] are shown in Figure 6.6. We see that the PSNR differences between Algorithm 1 and the algorithm in [9] are small, especially for the real image. The corresponding results of (5.3) with thresholding parameters being (6.2) are also shown in Figure 6.6. We see that for both choices of thresholding parameters Algorithm 1 always gives better PSNR results than (5.3).

In summary, for all the examples, the minimizer of (4.2), where there is a term to penalize the distance of $\alpha^{\diamond}$ to the range of $\mathcal{A}$, is better than the minimizer of (5.2). This is due to the relationship between the function smoothness and the ℓ^1 norm of the coefficient in the range of $\mathcal{A}$ for piecewise smooth functions as explained in Section 5.1. The improvement is more significant for synthetic piecewise smooth images.

15

FIG. 6.5. *From left to right: (a) Image contaminated by white Gaussian noise with standard deviation 5 (SNR=28.59dB), and then down-sampled; (b)* $\mathbf{f}^\star$ *(PSNR = 27.28dB, iter=454); (c)* $\mathbf{f}^\diamond$ *(PSNR = 27.28dB, iter=454); and (d) the restoration by (5.3) (PSNR=27.24dB, iter=354).*

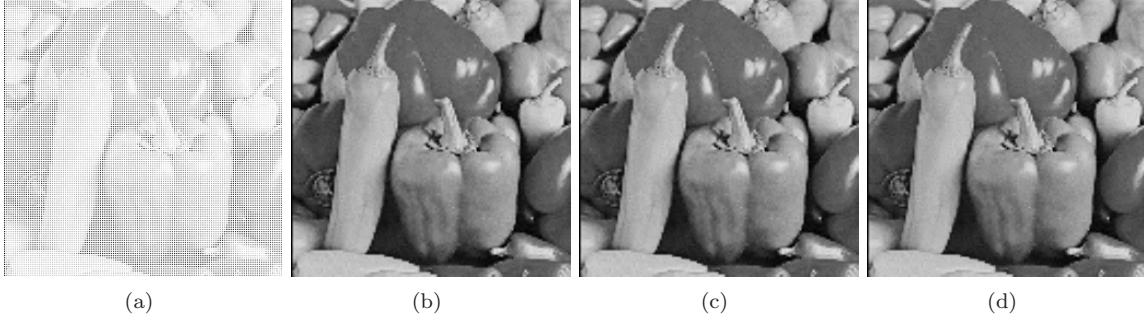(a)   (b)   (c)   (d)

FIG. 6.6. *The results using the thresholding parameter chosen as in (6.2) for both Algorithm 1 and iteration (5.3). From left to right: (a)* $\mathbf{f}^\diamond$ *(PSNR=31.92dB, iter=79) for the degraded image in Fig. 6.2(a); (b) the restoration by (5.3) (PSNR=29.82dB, iter=80) for the degraded image in Fig. 6.2(a); (c)* $\mathbf{f}^\diamond$ *(PSNR = 31.28dB, iter=115) for the degraded image in Fig. 6.4(a); and (d) the restoration by (5.3) (PSNR=31.27dB, iter=84) for the degraded image in Fig. 6.4(a).*

(a)   (b)   (c)   (d)
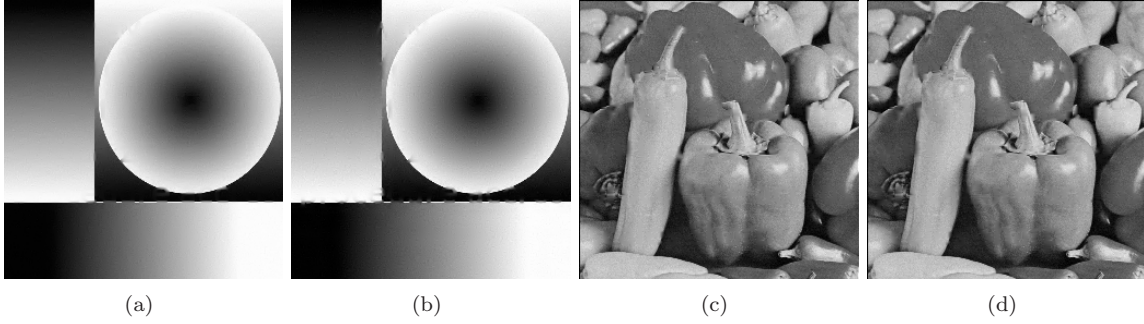
7. Conclusions and Perspectives. In this paper, we give an analysis of a framelet-based algorithm motivated by [6, 8, 9, 10, 29]. By interpreting it as a proximal forward-backward splitting iteration in [16], we prove the convergence of the algorithm. We further prove that the limit minimizes a cost functional that balances the sparsity, regularity and fidelity of the solution. Analytical and numerical comparisons with the framelet-based method presented in [27] are also given to illustrate the effectiveness of our method.

Under the setting of [16], it is difficult to establish the convergence rate of our algorithm. Our next research project will be to find it. It is also interesting to extend the convergence results to infinite dimensional setting.

8. Appendix. In this appendix, we show that Algorithm 1 converges even if we set

$$\lambda = (\underbrace{0, \cdots, 0}_{K_0}, \nu_{K_0+1}, \cdots, \nu_K)^T, \tag{8.1}$$

where $\nu_i > 0$ for $i = K_0 + 1, \cdots, K$. This is the algorithm proposed in [9].

16

8.1. Convergence Proof. Since the proof of Theorem 4.1 and Lemma 4.2 are still valid, it remains to show the existence of a minimizer of $\min_{\mathbf{f}}\{F_1(\mathbf{f}) + F_2(\mathbf{f})\}$.

LEMMA 8.1. *Let $\mathcal{A}$ be a tight frame system that satisfies one of the following conditions:*

(i) 1 is not an eigenvalue of $\mathcal{A}_0^*\mathcal{A}_0$; or

(ii) 1 is a simple eigenvalue of $\mathcal{A}_0^*\mathcal{A}_0$, and the corresponding eigenvector $\mathbf{u}$ satisfies $\mathcal{P}_\Lambda\mathbf{u} \neq \mathbf{0}$, where $\mathcal{A}_0$ is defined in (2.5). Then the minimization problem (4.1) has at least one minimizer.

Proof. Following the proof of Lemma 4.3, we need to show that $\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\}$ is coercive. Let $\nu_{\min} = \min_{i=K_0+1}^{K} \nu_i$ and $\nu_{\max} = \max_{i=K_0+1}^{K} \nu_i$. Then as in the proof of Lemma 4.3 (see (4.12)), we have

$$
\begin{aligned}
\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} &\geq \frac{1}{2}\|\mathcal{A}\mathbf{f} - \mathcal{T}_\lambda(\mathcal{A}\mathbf{f})\|_2^2 + \|\text{diag}(\lambda)\mathcal{T}_\lambda(\mathcal{A}\mathbf{f})\|_1 \\
&\geq \|\text{diag}(\lambda)\mathcal{T}_\lambda(\mathcal{A}\mathbf{f})\|_1 = \sum_{i=1}^{K_1} \nu_{K_0+i}|t_{\nu_{K_0+i}}(\mathcal{A}_1\mathbf{f})_i| \geq \nu_{\min}\sum_{i=1}^{K_1}|t_{\nu_{K_0+i}}(\mathcal{A}_1\mathbf{f})_i| \\
&\geq \nu_{\min}\sum_{i=1}^{K_1}(|(\mathcal{A}_1\mathbf{f})_i| - \nu_{K_0+i}) \geq \nu_{\min}\|\mathcal{A}_1\mathbf{f}\|_1 - \nu_{\min}\nu_{\max}K_1 \\
&\geq \nu_{\min}\|\mathcal{A}_1\mathbf{f}\|_2 - \nu_{\min}\nu_{\max}K_1. \qquad\qquad\qquad\qquad (8.2)
\end{aligned}
$$

First we consider the case when the condition (i) is satisfied. Since 1 is not an eigenvalue of $\mathcal{A}_0^*\mathcal{A}_0$, the matrix $(I - \mathcal{A}_0^*\mathcal{A}_0)$ is nonsingular. Let μ_1 be the smallest eigenvalue of $(I - \mathcal{A}_0^*\mathcal{A}_0)$. Then $\mu_1 > 0$. For any $\mathbf{f} \in \mathbb{R}^N$, using (2.6), we get

$$
\|\mathcal{A}_1\mathbf{f}\|_2^2 = \mathbf{f}^*\mathcal{A}_1^*\mathcal{A}_1\mathbf{f} = \mathbf{f}^*(\mathcal{I} - \mathcal{A}_0^*\mathcal{A}_0)\mathbf{f} \geq \mu_1\|\mathbf{f}\|_2^2. \qquad\qquad (8.3)
$$

By (8.2) and (8.3), we have

$$
\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} \geq \nu_{\min}\|\mathcal{A}_1\mathbf{f}\|_2 - \nu_{\min}\nu_{\max}K_1 \geq \nu_{\min}\sqrt{\mu_1}\|\mathbf{f}\|_2 - \nu_{\min}\nu_{\max}K_1.
$$

Therefore, whenever $\|\mathbf{f}\|_2 \to +\infty$, $\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} \to +\infty$.

Next we consider the case when the condition (ii) is satisfied. Let $\mathbf{V}$ be the subspace orthogonal to the unit eigenvector $\mathbf{u}$ corresponding to the eigenvalue 1 of $\mathcal{A}_0^*\mathcal{A}_0$, i.e., $\mathbf{V} = \{\mathbf{v} : \mathbf{u}^T\mathbf{v} = 0\}$. Since $\mathcal{A}_0^*\mathcal{A}_0$ has a simple eigenvalue 1, $(\mathcal{I} - \mathcal{A}_0^*\mathcal{A}_0)$ has a simple eigenvalue 0. Furthermore, the null space of $(\mathcal{I} - \mathcal{A}_0^*\mathcal{A}_0)$ is one dimensional and spanned by $\mathbf{u}$. Therefore, for an arbitrary $\mathbf{f}$, it can be decomposed into $\mathbf{f} = \mathbf{v} + a\mathbf{u}$ with $\mathbf{v} \in \mathbf{V}$, and

$$
\|\mathcal{A}_1\mathbf{f}\|_2^2 = \mathbf{f}^*\mathcal{A}_1^*\mathcal{A}_1\mathbf{f} = \mathbf{f}^*(\mathcal{I} - \mathcal{A}_0^*\mathcal{A}_0)\mathbf{f} \geq \mu_2\|\mathbf{v}\|_2^2, \qquad\qquad (8.4)
$$

where $\mu_2 > 0$ is the second smallest eigenvalue of $(\mathcal{I} - \mathcal{A}_0^*\mathcal{A}_0)$. Since $\mathcal{P}_\Lambda\mathbf{u} \neq \mathbf{0}$, there exists at least one index $i_0 \in \Lambda$ such that $\mathbf{u}(i_0)$ is nonzero. Let $C = \min\{\frac{\sqrt{3}}{2}, \frac{|\mathbf{u}(i_0)|}{4}\}$. Then $0 < C < 1$. As long as $\|\mathbf{f}\|_2 \geq \frac{4|\mathbf{g}(i_0)|}{|\mathbf{u}(i_0)|}$, we have the following two cases:

- If $\|\mathbf{v}\|_2 \geq C\|\mathbf{f}\|_2$, then by (8.2) and (8.4),

$$
\begin{aligned}
\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} &\geq \nu_{\min}\|\mathcal{A}_1\mathbf{f}\|_2 - \nu_{\min}\nu_{\max}K_1 \geq \nu_{\min}\sqrt{\mu_2}\|\mathbf{v}\|_2 - \nu_{\min}\nu_{\max}K_1 \\
&\geq \nu_{\min}\sqrt{\mu_2}C\|\mathbf{f}\|_2 - \nu_{\min}\nu_{\max}K_1.
\end{aligned}
$$

- If $\|\mathbf{v}\|_2 < C\|\mathbf{f}\|_2$, then $|a| > \sqrt{1 - C^2}\|\mathbf{f}\|_2$. Therefore,

$$
\begin{aligned}
|\mathbf{f}(i_0)| = |\mathbf{v}(i_0) + a\mathbf{u}(i_0)| &\geq |a||\mathbf{u}(i_0)| - |\mathbf{v}(i_0)| \\
&> \sqrt{1 - C^2}\|\mathbf{f}\|_2|\mathbf{u}_{i_0}| - C\|\mathbf{f}\|_2 = (\sqrt{1 - C^2}|\mathbf{u}_{i_0}| - C)\|\mathbf{f}\|_2
\end{aligned}
$$

17

$$\geq \left(\sqrt{1 - (\sqrt{3}/2)^2} |\mathbf{u}_{i_0}| - \frac{|\mathbf{u}(i_0)|}{4} \right) \|\mathbf{f}\|_2$$

$$= \frac{|\mathbf{u}(i_0)|}{4} \|\mathbf{f}\|_2 \geq |\mathbf{g}(i_0)|. \tag{8.5}$$

It implies that $\mathbf{f}$ cannot be in $\mathbf{C}$, hence

$$\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} = +\infty \geq \nu_{\min} \sqrt{\mu_2} C \|\mathbf{f}\|_2 - \nu_{\min} \nu_{\max} K_1.$$

Therefore, in both cases, once $\|\mathbf{f}\|_2 \geq \frac{4|\mathbf{g}(i_0)|}{|\mathbf{u}(i_0)|}$, it holds that

$$\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} \geq \nu_{\min} \sqrt{\mu_2} C \|\mathbf{f}\|_2 - \nu_{\min} \nu_{\max} K_1.$$

Hence, $\|\mathbf{f}\|_2 \to +\infty$ implies $\{\iota_{\mathbf{C}}(\mathbf{f}) + {}^1\xi(\mathcal{A}\mathbf{f})\} \to +\infty$. $\square$

Combining everything together, we can prove the convergence of Algorithm 1 with threshold (8.1). The proof is similar to the one in Theorem 4.4, and hence is omitted.

THEOREM 8.2. *Assume that one of the conditions (i) and (ii) in Lemma 8.1 holds. Then iteration (3.6) in Algorithm 1 converges to a minimizer of the minimization problem (4.1) for any initial guess $\mathbf{f}_0$.*

8.2. Examples of Tight Frames. Next we discuss what kind of tight framelet systems will satisfy the assumptions in Lemma 8.1. First we study the eigenvalues of the matrices defined in (2.4).

LEMMA 8.3. *For any symmetric sequence $h = \{h(j)\}_{j=-J}^{J}$, the eigenvectors of the matrix $\mathcal{S}(h)$ defined in (2.4) are the columns of the discrete cosine transform (DCT) matrix. Moreover, the corresponding eigenvalues are $\widehat{h}(i\pi/N)$ for $0 \leq i \leq N-1$, where $\widehat{h}(\theta) = \sum_{j=-J}^{J} h(j) e^{-j\theta\sqrt{-1}}$.*

Proof. Sine the sequence h is symmetric, by Theorem 3.2 in [36], the matrices $\mathcal{S}(h)$ can be diagonalized by the DCT matrix $\mathcal{C}$ defined by

$$\mathcal{C}(i,j) = \sqrt{\frac{2 - \delta_{i,0}}{N}} \cos\left(\frac{(2j+1)i\pi}{N} \right), \quad 0 \leq i, j \leq N-1,$$

where $\delta_{i,0}$ is the Kronecker delta. Moreover, by (3.3) in [36], the eigenvalues μ_i of $\mathcal{S}(h)$ are given by the formula $\mu_i = [\mathcal{C}\mathcal{S}(h)\mathbf{e}_1](i)/[\mathcal{C}\mathbf{e}_1](i)$ for $0 \leq i \leq N-1$, where $\mathbf{e}_1$ is the first column of the identity matrix. By direct calculations, we obtain

$$\mu_i = \frac{1}{\cos(i\pi/2N)} \sum_{j=1}^{N} [h(j) + h(j-1)] \cos\left(\frac{(2j-1)i\pi}{2N} \right)$$

$$= \frac{1}{\cos\theta_i} \left(\sum_{j=1}^{J+1} h(j-1) \cos[(2j-1)\theta_i] + \sum_{j=1}^{J} h(j) \cos[(2j-1)\theta_i] \right)$$

$$= \frac{1}{\cos\theta_i} \left(h(0) \cos\theta_i + \sum_{j=1}^{J} h(j) \cos[(2j+1)\theta_i] + \sum_{j=1}^{J} h(j) \cos[(2j-1)\theta_i] \right)$$

$$= h(0) + \sum_{j=1}^{J} h(j) \left(\frac{\cos[(2j+1)\theta_i] + \cos[(2j-1)\theta_i]}{\cos\theta_i} \right)$$

$$= h(0) + \sum_{j=1}^{J} h(j) \cos(2j\theta_i) = \sum_{j=0}^{J} h(j) \cos(2j\theta_i) = \sum_{j=-J}^{J} h(j) e^{-2j\theta_i\sqrt{-1}},$$

18

where $\theta_i = i\pi/(2N)$.　□

With this, we can show that Algorithm 1 converges for symmetric tight framelet systems.

PROPOSITION 8.4. *Consider the tight frame system $\mathcal{A}$ given in (2.5) with a symmetric refinement mask. Assume that*

$$\{\theta \in (0, \pi) : |\zeta_\phi(\theta)|^2 = 1\} = \emptyset. \tag{8.6}$$

Then, condition (ii) in Lemma 8.1 holds.

Proof. Since the refinement mask is symmetric, we may assume (up to the multiplication of an exponential) that ζ_ϕ is an even function. This, together with $|\zeta_\phi(0)|^2 = 1$, (8.6), and Lemma 8.3, implies that $(\mathcal{H}_0)^*\mathcal{H}_0$ has a simple eigenvalue 1, and the corresponding eigenvector is $\mathbf{1}$, the vector of all ones. Clearly $\mathcal{P}_\Lambda \mathbf{1} \neq \mathbf{0}$.

Recall $\mathcal{A}_0 \equiv \prod_{\ell=0}^{L-1} \mathcal{H}_0^{(L-\ell)}$. Let $(\mu_i^{(\ell)})^2$ be the eigenvalues of $(\mathcal{H}_0^{(\ell)})^*\mathcal{H}_0^{(\ell)}$. Since the matrices $\mathcal{H}_0^{(\ell)}$ for $\ell = 1, \cdots, L$ can all be diagonalized by the same DCT matrix, the eigenvalues of $\mathcal{A}_0^*\mathcal{A}_0$ are given by

$$\prod_{\ell=1}^{L}(\mu_i^{(\ell)})^2 = \prod_{\ell=1}^{L}\left|\zeta_\phi\left(2^{\ell-1}\frac{i\pi}{N}\right)\right|^2, \quad \text{for } 0 \leq i < N - 1,$$

which equal 1 if and only if $(\mu_i^{(\ell)})^2 = 1$ for all ℓ. This is obviously true when $i = 0$ whose corresponding eigenvector is $\mathbf{1}$. Assume that there is another $i \neq 0$ that satisfies the condition, then

$$\frac{i\pi}{N}, 2\frac{i\pi}{N}, \cdots, 2^{L-1}\frac{i\pi}{N} \in \{\theta : |\zeta_\phi(\theta)|^2 = 1\}$$

which contradicts (8.6). Therefore, 1 is a simple eigenvalue of $\mathcal{A}_0^*\mathcal{A}_0$ whose corresponding eigenvector is $\mathbf{1}$, and $\mathcal{P}_\Lambda \mathbf{1} \neq \mathbf{0}$. □

The lemma shows that Algorithm 1 converges if the symmetric tight framelet systems derived from B-splines, or more general, the pseudo-splines of type II introduced in [22] are used.

REFERENCES

[1] M. Bertalmío, A. Bertozzi, and G. Sapiro, "Navier Stokes, fluid-dynamics and image and video inpainting", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1 (2001), pp. 355–362.

[2] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting", *SIGGRAPH*, 34 (2000), pp. 417–424.

[3] M. Bertalmío, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting", *IEEE Transactions on Image Processing*, 12 (2003), pp. 882–889.

[4] 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.

[5] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information", *IEEE Trans. Inform. Theory*, 52 (2006), pp. 489–509.

[6] A. Chai and Z. Shen, "Deconvolution: A wavelet frame approach", *Numer. Math.*, 106 (2007), pp. 529–587.

[7] 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.

[8] R. Chan, S. 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.

[9] R. Chan, L. Shen, and Z. Shen, "A framelet-based approach for image inpainting", Research Report 2005-04(325), Department of Mathematics, The Chinese University of Hong Kong, 2005.

[10] R. Chan, L. Shen, and T. Xia, "A framelet algorithm for enhancing video stills", *Appl. Comput. Harmon. Anal.*, 23 (2007), pp. 153–170.

[11] T. Chan, and J. Shen, "Non-texture inpainting by curvature-driven diffusions(CDD)", *Journal of Visual Communication and Image Representation*, 12 (2001), pp. 436–449.

[12] T. Chan, and J. Shen, "Mathematical models for local non-texture inpainting", *SIAM Journal on Applied Mathematics*, 62 (2001), pp. 1019–1043.

[13] T. Chan, and J. Shen, "Variational image inpainting", *Communications on Pure and Applied Mathematics*, 58 (2005), pp. 579–619.

[14] T. F. Chan, J. Shen, and H.-M. Zhou, "Total variation wavelet inpainting", *J. Math. Imaging Vision*, 25 (2006), pp. 107–125.

[15] T. Chan, S.H. Kang, and J. Shen, "Euler's elastica and curvature based inpainting", *SIAM Journal on Applied Mathematics*, 63 (2002), pp. 564–592.

[16] P. L. Combettes, and V. R. Wajs, "Signal recovery by proximal forward-backward splitting", *Multiscale Modeling and Simulation*, 4 (2005), pp. 1168–1200.

[17] I. Daubechies, *Ten Lectures on Wavelets*, vol. 61 of CBMS Conference Series in Applied Mathematics, SIAM, Philadelphia, 1992.

[18] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint", *Communications on Pure and Applied Mathematics*, 57 (2004), pp. 1413–1457.

[19] I. Daubechies, B. Han, A. Ron, and Z. Shen, "Framelets: MRA-based constructions of wavelet frames", *Applied and Computational Harmonic Analysis*, 14 (2003), pp. 1–46.

[20] C. de Boor, R. DeVore, and A. Ron, "On the construction of multivariate (pre)-wavelets", *Constructive Approximation*, 9 (1993), pp. 123–166.

[21] C. De Mol and M. Defrise, "A note on wavelet-based inversion algorithms", in *Inverse problems, image analysis, and medical imaging (New Orleans, LA, 2001)*, vol. 313 of Contemp. Math., Amer. Math. Soc., Providence, RI, 2002, pp. 85–96.

[22] B. Dong and Z. Shen, "Pseudo-splines, wavelets and framelets", *Appl. Comput. Harmon. Anal.*, 22 (2007), pp. 78–104.

[23] D. Donoho, "De-noising by soft-thresholding", *IEEE Transactions on Information Theory*, 41 (1995), pp. 613–627.

[24] D. Donoho, "For most large underdetermined systems of equations, the minimal l_1-norm near-solution approximates the sparsest near-solution", *Communications on Pure and Applied Mathematics*, 59 (2006), pp. 907–934.

[25] M. Elad, "Why simple shrinkage is still relevant for redundant representations?", *IEEE Trans. Inform. Theory*, 52 (2006), pp. 5559–5569.

[26] M. Elad, J.-L. Starck, P. Querre, and D. Donoho, "Simultaneous cartoon and texture image inpainting using morphplogical component analysis (MCA)", *Applied and Computational Harmonic Analysis*, 19 (2005), 340–358.

[27] M.J. Fadili, and J.-L. Starck, "Sparse representations and Bayesian image inpainting", *Proc. SPARS'05*, Vol. I, Rennes, France, 2005.

[28] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration", *IEEE Trans. Image Process.*, 12 (2003), pp. 906–916.

[29] O.G. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising – Part II: adaptive algorithms", *IEEE Transactions on Image Processing*, 15 (2006), pp. 555–571.

[30] J.-B. Hiriart-Urruty, and C. Lemaréchal, *Convex Analysis and Minimization Algorithms, vol. I*, Springer-Verlag, Berlin, 1996.

[31] Y. Hur and A. Ron, "CAPlets: wavelet representations without wavelets", preprint, (2005)

[32] R. Jia and Z. Shen, "Multiresolution and wavelets", *Proceedings of the Edinburgh Mathematical Society*, 37 (1994), pp. 271–300.

[33] S. Mallat, *A Wavelet Tour of Signal Processing, 2nd Edition*, Academic Press, 1999.

[34] 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.

[35] J.-J. Moreau, "Proximité et dualité dans un espace Hilbertien", *Bull. Soc. Math. France*, 93 (1965), pp. 273–299.

[36] 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.

[37] A. Ron and Z. Shen, "Affine system in $L_2(\mathbb{R}^d)$: the analysis of the analysis operator", *Journal Func. Anal.*, 148 (1997), pp. 408–447.

[38] I.W. Selesnick, R. Van Slyke, and O.G. Guleryuz, "Pixel recovery via ℓ_1 minimization in the wavelet domain"

Proc. IEEE Intl Conf. on Image Proc.(ICIP2004), Vol. III, pp. 1819–1822, Singapore, 2004.

[39] G. WAHBA, *Spline models for observational data*, vol. 59 of CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.