Tight Frame Based Method for High-Resolution Image Reconstruction

Jian-Feng Cai∗  Raymond Chan†  Lixin Shen‡  Zuowei Shen§
June 25, 2008

1 High-Resolution Image Reconstruction Model

High-resolution image reconstruction is to reconstruct a high-resolution (HR) image from multiple, undersampled, shifted, degraded and noisy frames obtained by using multiple images shifted from each other by sub-window shifts. Problems of high-resolution restoration for images arise in a variety of scientific, medical, and engineering applications. The problem of HR image reconstruction is becoming a very hot field. Two special issues on this topic are launched recently in IEEE Signal Processing Magazine (Volume 20, Issue 3, May 2003) and International Journal of Imaging Systems and Technology (vol. 14, no. 2, 2004).

The earliest formulation of the problem of HR image reconstruction was motivated by the need of improved resolution images from Landsat image data. In [28], Huang and Tsay used the frequency domain approach to demonstrate reconstruction of one improved resolution image from several down-sampled noise-free versions of it. Later on, Kim et al. [30] generalized this idea to noisy and blurred images. Both methods in [28, 30] are computational efficiency, but, they are prone to model errors, and that limit their use [1]. Statistical methods have appeared recently for super-resolution image reconstruction problems. In this direction, tools such as a maximum a posteriori (MAP) estimator with the Huber-Markov random filed prior and a Gibbs image prior are proposed in [25, 43]. In particular, the task of simultaneous image registration and super-resolution image reconstruction are studied in [25, 45]. Iterative spatial domain methods are popular class of methods for solving the problems of resolution enhancement [3, 21, 22, 23, 27, 31, 32, 36, 38, 39, 41]. The problems are formulated as Tikhonov regularization. A great deal of work has been devoted to the efficient calculation of the reconstruction and the estimation of the associated hyperparameters by taking advantage of the inherent structures in the HR system matrix. Bose and Boo [3] use a block semi-circulant matrix decomposition in order to calculate the MAP reconstruction. Ng et al. [36] and Ng and Yip [37] proposed a fast discrete cosine transform based approach for HR image reconstruction with Neumann boundary condition. Nguyen et al. [40, 41] also addressed the problem of efficient calculation. The proper choice of the regularization tuning parameter is crucial to achieving robustness in the presence of noise and avoiding trial-and-error in the selection of an optimal tuning parameter. To this end, Bose et al. [4] used a L-curve based approach. Nguyen et al. [41] used a generalized cross-validation method. Molina et al. [33] used an expectation-maximization algorithm. Lu et al. [32] proposed multiparameter regularization methods which introduce different regularization parameters for different frequency bands of the regularization operator.

Low-resolution images can be viewed as outputs of the original high-resolution images passing through a low-pass filter followed by a decimation process. This viewpoint suggests that a framework of multiresolution analysis can be naturally adopted to produce a HR image from a set of low-resolution images of the same

∗Temasek Laboratories, National University of Singapore. Email: tslcaij@nus.edu.sg. Research supported by the Wavelets and Information Processing Programme under a grant from DSTA, Singapore.
†Department of Mathematics, The Chinese University of Hong Kong. Email: rchan@math.cuhk.edu.hk. The research was supported in part by HKRGC Grant 400505 and CUHK DAG 2000257.
‡Department of Mathematics, Syracuse University. Email: lshen03@syr.edu. This work was supported by the US National Science Foundation under grant DMS-0712827. The work was partially done while this author was visiting the Institute for Mathematical Sciences, National University of Singapore in 2008. The visit was partially supported by the institute.
§Department of Mathematics, National University of Singapore. Email:matzuows@nus.edu.sg. The research was supported in part by Grant R-146-000-060-112 at the National University of Singapore.
scene. In this fashion, a series of work has been done recently, see, e.g., [9, 10, 11, 12, 13]. Extension of these work will be discussed in the paper.

Here we present a mathematical model proposed by Bose and Boo in [3] for high-resolution image reconstruction. Consider $K \times K$ sub-window-shifted low-resolution images in which each image has $N_1 \times N_2$ interrogation windows and the size of each interrogation window is $T_1 \times T_2$. Here, $K \times K$ denotes $K$ shifts in both the vertical and horizontal directions. The goal is to reconstruct a much higher resolution image with $M_1 \times M_2$ sub-windows, where $M_1 = K \times N_1$ and $M_2 = K \times N_2$.

In order to have enough information to resolve the high-resolution image, it is assumed that there are sub-window shifts between the low-resolution images. For a low-resolution image denoted by $(k_1, k_2)$, where $0 \leq k_1, k_2 < K$ with $(k_1, k_2) \neq (0, 0)$, its vertical and horizontal shifts $d_{k_1, k_2}^x$ and $d_{k_1, k_2}^y$ with respect to the $(0, 0)$th reference low-resolution image are given by $d_{k_1, k_2}^x = (k_1 + \epsilon_{k_1, k_2}^x) \frac{T_2}{K}$ and $d_{k_1, k_2}^y = (k_2 + \epsilon_{k_1, k_2}^y) \frac{T_1}{K}$. Here $\epsilon_{k_1, k_2}^x$ and $\epsilon_{k_1, k_2}^y$ are the vertical and horizontal shift errors respectively. We assume that $|\epsilon_{k_1, k_2}^x| < \frac{1}{2}$ and $|\epsilon_{k_1, k_2}^y| < \frac{1}{2}$. Figure 1 shows the example of $2 \times 2$ shifted low-resolution images.

For a low-resolution image $(k_1, k_2)$, the average quantity at its $(n_1, n_2)$th interrogation window is modelled by:

$$g_{k_1, k_2}[n_1, n_2] = \frac{1}{T_1 T_2} \int_{A_{k_1, k_2; n_1, n_2}} f(x, y) \, dx \, dy + \eta_{k_1, k_2}[n_1, n_2],$$

(1)

where the interrogation window in the low-resolution image is

$$A_{k_1, k_2; n_1, n_2} = \left[ T_1(n_1 - \frac{1}{2}) + d_{k_1, k_2}^x, T_1(n_1 + \frac{1}{2}) + d_{k_1, k_2}^x \right] \times \left[ T_2(n_2 - \frac{1}{2}) + d_{k_1, k_2}^y, T_2(n_2 + \frac{1}{2}) + d_{k_1, k_2}^y \right].$$

Here $(n_1, n_2)$ indicates an interrogation window in the low-resolution image $(k_1, k_2)$ (where $0 \leq n_1 < N_1$ and $0 \leq n_2 < N_2$) and $\eta_{k_1, k_2}[n_1, n_2]$ is the noise (refer to [3]). We interface all the sub-window-shifted low-resolution images $g_{k_1, k_2}$ to form an $M_1 \times M_2$ image $g$ by assigning

$$g[Kn_1 + k_1, Kn_2 + k_2] = g_{k_1, k_2}[n_1, n_2].$$

The pseudo high-resolution image $g$ is called the observed high-resolution image.

The integral values on the sub-window of the high-resolution image is approximated by

$$f[i, j] = \frac{K^2}{T_1 T_2} \int_{A_{i,j}} f(x, y) \, dx \, dy, \quad 0 \leq i < M_1, 0 \leq j < M_2,$$

(2)

which is the average quantity inside the $(i, j)$th high-resolution sub-window:

$$A_{i,j} = \left[ \frac{T_1}{K}, (i + 1) \frac{T_1}{K} \right] \times \left[ \frac{T_2}{K}, (j + 1) \frac{T_2}{K} \right], \quad 0 \leq i < M_1, 0 \leq j < M_2.$$

(3)
To obtain the true high-resolution image $f$ from the observed high-resolution image $g$, one will have to solve (1) for $f$. By discretizing (1) and (2) using the rectangular quadrature rule, we have

$$g_{k_1, k_2}[n_1, n_2] = \sum_{p, q=0}^K W[p, q]f[Kn_1 + k_1 + p, Kn_2 + k_2 + q] + \eta_{k_1, k_2}[n_1, n_2], \quad (4)$$

where the weighting matrix $W$ for descretizing the integral equation (1) in the case without shift error is

$$W = \frac{1}{K^2} \left[ \begin{array}{ccc}
\frac{1}{4} & \frac{1}{4} & \cdots & \frac{1}{4} \\
\frac{1}{2} & 1 & \cdots & 1 \\
\frac{1}{2} & 1 & \cdots & 1 \\
\vdots & \vdots & \ddots & \vdots \\
\frac{1}{4} & \frac{1}{4} & \cdots & \frac{1}{4}
\end{array} \right] \quad (5)$$

which is assigned for associated sub-windows of the high-resolution image. We then obtain a system of linear equations relating the unknown values $f[i, j]$ to the given observed high-resolution image values $g[i, j]$.

For simplifying the exposition, $f$ and $g$ will be considered as the column vectors formed by $f[i, j]$ and $g[i, j]$. This linear system corresponding to (4) for high-resolution image reconstruction is reduced to

$$Hf + \eta = g, \quad (6)$$

where the blurring matrix $H$, which is formulated from (4), varies under different boundary conditions and $\eta$ is the noise vector. For the case without shift error, the blurring matrix $H$ is given by

$$H = \frac{1}{K^4} \left[ \begin{array}{ccc}
\cdots & \frac{1}{2} & 1 \\
\frac{1}{2} & 1 & \cdots \\
1 & \cdots & 1
\end{array} \right] \otimes \left[ \begin{array}{ccc}
\cdots & \frac{1}{2} & 1 \\
\frac{1}{2} & 1 & \cdots \\
1 & \cdots & 1
\end{array} \right]$$

where the Kronecker operator $\otimes$ is defined by $A \otimes B = [a_{ij}]B$ with $A = [a_{ij}]$. The key problem is to recover the true high-resolution image $f$ from the observed high-resolution image $g$ by solving (6).

If the low-resolution images are shifted by exactly half of the window, then the problem reduces to solving a spatially invariant linear system. Depending on the boundary conditions we impose on the images, the coefficient matrix $H$ is either Toeplitz or Toeplitz-like. The model was then solved in [3, 36] using preconditioned conjugate gradient method.

We next discuss in details several approaches that will use tight frame for solving the system (4) or (6). The performance of these methods will be examined in numerical simulations. In the next section, we will give a brief review on the frame theory. In particular, we will present the tight frame system with (5) as its low-pass filter.

The outline of this paper is as follows. In Section 2, we give a brief review on tight frames with an emphasis on the unitary extension principle. Section 3 contains four main parts. The first part presents the tight frames arising from the problem of HR image reconstruction. The matrix representations of the tight frame filters associated with the HR image reconstruction are given, by imposing the periodic and symmetric boundary conditions, in the second and third parts, respectively. It follows by showing the multi-level framelet decomposition and reconstruction in the last part. We propose three framelet-based algorithms to tackle the problem of HR image reconstruction in Section 4. In particular, we give a complete analysis for Algorithm I in Section 5. Numerical experiments for all three algorithms are presented in Section 6.

2 Preliminaries on Tight Framelets

The notion of frame was first introduced by Duffin and Schaeffer [20] in 1952. A countable system $X \subset L^2(\mathbb{R})$ is called a frame of $L^2(\mathbb{R})$ if

$$A\|f\|_2^2 \leq \sum_{h \in X} |\langle f, h \rangle|^2 \leq B\|f\|_2^2 \quad (7)$$
where the constants \( A \) and \( B \), \( 0 < A \leq B < \infty \), are lower and upper bounds of the frame system \( X \). The notation \( \langle \cdot, \cdot \rangle \) and \( \| \cdot \|_2 = \langle \cdot, \cdot \rangle^{1/2} \) are the inner product and norm of \( L^2(\mathbb{R}) \). When \( A = B(=1) \), the frame system \( X \) is called a tight frame. In what follows, our discussion is concentrated on the tight frame.

Two operators, namely analysis operator and synthesis operator, are associated with the tight frame. The analysis operator of the frame is defined as

\[
\mathcal{F} : L^2(\mathbb{R}) \longrightarrow \ell^2
\]

with

\[
\mathcal{F}(f) = \{ \langle f, h \rangle \}_{h \in X}. 
\]

Its adjoint operator \( \mathcal{F}^* \), the synthesis operator, is defined as

\[
\mathcal{F}^* : \ell^2 \longrightarrow L^2(\mathbb{R})
\]

with

\[
\mathcal{F}^*(c) = \sum_{h \in X} c_h h, \quad c = \{ c_h \}_{h \in X}. 
\]

Hence, \( X \) is a tight frame if and only if \( \mathcal{F}^* \mathcal{F} = I \). In other words, if

\[
f = \sum_{h \in X} \langle f, h \rangle h, \quad \forall f \in L^2(\mathbb{R}), \tag{8}
\]

this is equivalent to

\[
\| f \|^2 = \sum_{h \in X} |\langle f, h \rangle|^2, \quad \forall f \in L^2(\mathbb{R}). \tag{9}
\]

Equation (8) is the perfect reconstruction formula of the tight frame. Identities (8) and (9) hold for an arbitrary orthonormal basis of \( L^2(\mathbb{R}) \). In this sense, an orthonormal basis is a tight frame, and a tight frame is a generalization of orthonormal basis. But tight frames sacrifice the orthonormality and the linear independence of the system in order to get more flexibility. Therefore tight frames can be redundant.

For a tight frame system \( X \), we have

\[
\sum_{h \in X} |\langle f, h \rangle|^2 \leq \sum_{h \in X} |c_h|^2
\]

for all possible representation of \( f = \sum_{h \in X} c_h h \), \( \{ c_h \} \in \ell^2 \). In other words, the sequence \( \mathcal{F}(f) \) obtained by the analysis operator \( \mathcal{F} \) has the smallest \( \ell^2 \) norm among all sequences \( \{ c_h \} \in \ell^2 \) satisfying \( f = \sum_{h \in X} c_h h \).

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

\[
X(\Psi) = \{ K^{i/2} \psi(K^j x - j) : \psi \in \Psi, i, j \in \mathbb{Z} \},
\]

then \( X(\Psi) \) is called a wavelet (or affine) system of dilation \( K \). 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

\[
\hat{\phi}(K\cdot) = \tau_\phi \hat{\phi}. 
\]

Here \( \hat{\phi} \) 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, details about that is omitted here, but can be found, for instance, in [19, 29].

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

\[
\hat{\psi}(K\cdot) = \tau_\psi \hat{\phi}
\]
for some $2\pi$-periodic $\tau_\psi$. The unitary extension principle (UEP) of [42] 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:

$$
\tau_\phi(\omega)\tau_\phi\left(\omega + \frac{2\gamma \pi}{K}\right) + \sum_{\psi \in \Psi} \tau_\psi(\omega)\tau_\psi\left(\omega + \frac{2\gamma \pi}{K}\right) = \delta_{\gamma,0}, \quad \gamma = 0, 1, \ldots, K - 1
$$

(10)

for almost all $\omega$ in $\mathbb{R}$. Practically, we require all masks to be trigonometric polynomials. Thus, (10) 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 (10) holds. A more general principle of construction tight framelets, the oblique extension principle, was developed recently in [14, 17].

In the next section, we will use the EUP to construct a framelet system arising from the problem of HR image reconstruction.

3 Tight Frame System Arising from High-resolution Image Reconstruction

3.1 Filter Design

The low-pass filter (5) for high-resolution image reconstruction is separable and can be written as follows

$$
W = h_0^T h_0
$$

where

$$
h_0 = \frac{1}{K} \begin{bmatrix} 1 & \frac{1}{2} & 1 & \cdots & 1 & \frac{1}{2} \end{bmatrix}.
$$

Hence, to design a tight frame system with $W$ as its low-pass filter, we just need to construct a tight frame system with $h_0$ as its low-pass filter.

By virtue of the Fourier series of $h_0$, define

$$
\hat{\phi}(\omega) := \prod_{j=1}^{\infty} \hat{h}_0(K^{-j} \omega)
$$

(11)

where

$$
\hat{h}_0(\omega) = \frac{1}{2K} + \frac{1}{K} \left( \sum_{k=1}^{K-1} \exp(-i k \omega) \right) + \frac{1}{2K} \exp(-iK \omega).
$$

It was shown in [44] that $\phi$ is in $L^2(\mathbb{R})$ and Hölder continuous with Hölder exponent $\ln 2/\ln K$.

For any integer $L \geq 2$, define

$$
m_{L,p} := \frac{\sqrt{2}}{L} \left[ \cos \left( \frac{p \pi}{2L} \right), \cos \left( \frac{3p \pi}{2L} \right), \cdots, \cos \left( \frac{(2L-1)p \pi}{2L} \right) \right],
$$

and their Fourier series

$$
\hat{m}_{L,p}(\omega) = \frac{\sqrt{2}}{L} \sum_{\ell=1}^{L} \cos \left( \frac{(2\ell-1)p \pi}{2L} \right) \exp(-i \ell \omega),
$$

for $p = 1, \ldots, L - 1$.

We further define, for any integer $K \geq 2$,

$$
\hat{h}_{2p+q}(\omega) := \hat{m}_{2,q}(\omega) \hat{m}_{K,p}(\omega)
$$

(12)

where $p = 1, \ldots, K - 1$, $q \in \{0, 1\}$. We can easily check that

$$
\sum_{q=0}^{1} \sum_{p=0}^{K-1} \hat{h}_{2p+q}(\omega) \hat{h}_{2p+q}(\omega + \frac{2\pi \ell}{K}) = \delta_{\ell,0}, \quad \ell = 0, 1, \ldots, K - 1.
$$

(13)
The EUP of [42] yields that the functions

\[ \Psi = \{ \psi_{2p+q} : 0 \leq p \leq K - 1, \quad q = 0, 1, \quad (p, q) \neq (0, 0) \} \]

defined by

\[ \hat{\psi}_{2p+q}(\omega) = \hat{h}_{2p+q}\left( \frac{\omega}{K} \right) \hat{\varphi}\left( \frac{\omega}{K} \right). \]

are tight framelets. Furthermore,

\[ X(\Psi) = \left\{ K^{k/2} \psi_{2p+q}(K^k \cdot j) : 0 \leq p \leq K - 1, q = 0, 1, (p, q) \neq (0, 0); k, j \in \mathbb{Z} \right\} \]

is a tight frame system of \( L^2(\mathbb{R}) \).

In the following discussion, we always assume that the indexes of all filters \( h_\ell \), run from \(-K/2\) to \( K/2\) for even number \( K \) and \(-(K + 1)/2\) to \((K - 1)/2\) for odd number.

We are interested in the matrix representation of the identity

\[ \sum_{p=0}^{K-1} \left| \hat{h}_{2p+q}(\omega) \right|^2 = 0 \quad (14) \]

for filters given by (12).

In image processing, periodic and symmetric boundary conditions are usually imposed to give matrix representation of (14). In the following subsections, we will give the corresponding representations for both boundary conditions.

### 3.2 Matrix Representation of Filters with Periodic Boundary Conditions

For simplicity, we are not going to write matrix forms of filters given by (12) for a general integer \( K \); instead, we show these matrices for the filters with \( K = 2 \) and \( K = 3 \) only. One can easily give the matrix representation for filters associated with any integer \( K \).

**Example 1.** For \( K = 2 \), we have the low-pass filter \( h_0 = \frac{1}{4}[1, 2, 1] \) and three high-pass filters \( h_1 = \frac{1}{4}[1, 0, -1], \ h_2 = \frac{1}{4}[1, 0, -1], \) and \( h_3 = \frac{1}{4}[1, -2, 1] \), respectively. The corresponding matrix representation under the periodic boundary condition for filters \( h_0, h_1, h_2, \) and \( h_3 \) are circulant matrices \( H_0, H_1, H_2, \) and \( H_3 \), respectively, with their first rows being the following

\[
\begin{align*}
H_0 &= \begin{bmatrix}
1 & 1 & 0 & \cdots & 0 & 1 \\
2 & 4 & 0 & \cdots & 0 & 4 \\
0 & -1 & 0 & \cdots & 0 & 1 \\
0 & 1 & 0 & \cdots & 0 & 1 \\
-1 & 2 & 0 & \cdots & 0 & 1 \\
0 & 4 & 0 & \cdots & 0 & 4 
\end{bmatrix}, \\
H_1 &= \begin{bmatrix}
1 & 1 & 0 & \cdots & 0 & 1 \\
2 & 4 & 0 & \cdots & 0 & 4 \\
0 & -1 & 0 & \cdots & 0 & 1 \\
0 & 1 & 0 & \cdots & 0 & 1 \\
-1 & 2 & 0 & \cdots & 0 & 1 \\
0 & 4 & 0 & \cdots & 0 & 4 
\end{bmatrix}, \\
H_2 &= \begin{bmatrix}
1 & 1 & 0 & \cdots & 0 & 1 \\
2 & 4 & 0 & \cdots & 0 & 4 \\
0 & -1 & 0 & \cdots & 0 & 1 \\
0 & 1 & 0 & \cdots & 0 & 1 \\
-1 & 2 & 0 & \cdots & 0 & 1 \\
0 & 4 & 0 & \cdots & 0 & 4 
\end{bmatrix}, \\
H_3 &= \begin{bmatrix}
1 & 1 & 0 & \cdots & 0 & 1 \\
2 & 4 & 0 & \cdots & 0 & 4 \\
0 & -1 & 0 & \cdots & 0 & 1 \\
0 & 1 & 0 & \cdots & 0 & 1 \\
-1 & 2 & 0 & \cdots & 0 & 1 \\
0 & 4 & 0 & \cdots & 0 & 4 
\end{bmatrix}.
\end{align*}
\]

We can check that

\[ H_0^T H_0 + H_1^T H_1 + H_2^T H_2 + H_3^T H_3 = I. \]

**Example 2.** For \( K = 3 \), we have the low-pass filter \( h_0 = \frac{1}{6}[1, 2, 2, 1] \) and five high-pass filters \( h_1 = \frac{1}{6}[1, 0, 0, -1], \ h_2 = \frac{\sqrt{3}}{12}[1, 1, -1, -1], \ h_3 = \frac{\sqrt{3}}{12}[1, -1, -1, 1], \ h_4 = \frac{\sqrt{3}}{12}[1, -1, 1, 1], \) and \( h_5 = \frac{\sqrt{3}}{12}[1, -3, 3, -1] \). The corresponding matrix representation under the periodic boundary condition for filters \( h_0, h_1, h_2, h_3, h_4, \) and \( h_5 \) are circulant matrices \( H_0, H_1, H_2, H_3, H_4, \) and \( H_5 \), respectively, with their first rows being the
3.3 Matrix Representation of Filters with Symmetric Boundary Conditions

A filter \( h \) is said to be symmetric (or antisymmetric) with symmetric center \( \frac{N}{2} \) for some integer \( N \) if

\[
h[N - k] = \pm h[k], \quad k \in \mathbb{Z}.
\]

We denote \( c(h) = \frac{N}{2} \) the symmetric center of the symmetric (or antisymmetric) filter \( h \). With this definition, we immediately have the following result.

**Proposition 1.** Let \( h_\ell \) be filters constructed by (12). Then \( c(h_\ell) = 0 \) for \( K \) being even numbers and \( c(h_\ell) = \frac{1}{2} \) for \( K \) being odd numbers.

For any infinite signal \( u = (\ldots, u(-1), u(0), u(1), \ldots)^t \), where \( u(k) \in \mathbb{R}, k \in \mathbb{Z} \), if there exists an integer \( n \) such that \( u(n - k) = \pm u(k) \), for all \( k \in \mathbb{Z} \), then we say that \( u \) is symmetric (or antisymmetric) with the center \( c(u) = \frac{n}{2} \). Accordingly, if \( c(u) = \frac{n}{2} \) is an integer, it is referred to as whole-sample symmetric (or whole-sample antisymmetric) and denoted by WS (or WA); if \( c(v) = \frac{n}{2} \) is not an integer, it is referred to as half-sample symmetric (or half-sample antisymmetric) and denoted by HS (or HA). If \( u \) has two centers \( c_1 \) and \( c_2 \) with \( c_1 < c_2 \), we denote it by \( c(u) = (c_1, c_2) \).

For a finite-length signal

\[
u = (u(0), \ldots, u(N-1))^t,
\]

the infinite signal \( v = (\ldots, v(-1), v(0), v(1), \ldots)^t \) is an extended signal of \( u \) if \( v(k) = u(k) \), for \( 0 \leq k \leq N - 1 \). We define a restriction operator \( P_N \) as follows:

\[
P_N : \mathbb{R}^\infty \rightarrow \mathbb{R}^N; \quad (\ldots, x(-1), x(0), \ldots, x(N-1), x(N), \ldots)^t \mapsto (x(0), \ldots, x(N-1))^t.
\]

Hence, if \( v \) is an extended signal of a finite length signal \( u \) of length \( N \), then \( u = P_N(v) \).

There are many ways to extend a signal into another signal with infinite length. We are mostly interested in a symmetric extension method since all filters \( h_\ell \) are symmetric or antisymmetric. By doing so, our goal is to construct matrices \( H_\ell \) and \( H_\ell^T \) associated with filters \( h_\ell \) and \( h_\ell^T \) (\( h_\ell[k] = h_\ell[-k] \)) of size \( N \times N \) such that

\[
\sum_\ell H_\ell^T H_\ell = I. \tag{15}
\]

Clearly, the properties of extension methods are reflected in the structures of the matrices \( H_\ell \) and \( H_\ell^T \).

To develop the matrix representation of filters in (12), we therefore restrict ourselves to the following extension methods:

\[
\frac{1}{6} [2, 1, 0, \ldots, 0, 1, 2],
\frac{1}{6} [0, -1, 0, \ldots, 0, 1, 0],
\frac{\sqrt{6}}{12} [-1, -1, 0, \ldots, 0, 1, 1],
\frac{\sqrt{6}}{12} [-1, 1, 0, \ldots, 0, 1, -1],
\frac{\sqrt{2}}{12} [-1, 1, 0, \ldots, 0, 1, -1],
\frac{\sqrt{2}}{12} [3, -1, 0, \ldots, 0, 1, -3].
\]

Again, it can be easily checked that

\[
H_0^T H_0 + H_1^T H_1 + H_2^T H_2 + H_3^T H_3 + H_4^T H_4 + H_5^T H_5 = I.
\]
• The whole-sample symmetric extension (WSWS) $E_s^{(1,1)}$:
  \[ E_s^{(1,1)}(u) = (\cdots, u(2), u(1), u(0), u(1), \cdots, u(N-2), u(N-1), u(N-2), u(N-3), \cdots)^t. \]

• The whole-half-sample extension (WSHS) $E_s^{(1,2)}$:
  \[ E_s^{(1,2)}(u) = (\cdots, u(2), u(1), u(0), u(1), \cdots, u(N-2), u(N-1), u(N-2), u(N-3), \cdots)^t. \]

• The half-whole-sample extension (HSWS) $E_s^{(2,1)}$:
  \[ E_s^{(2,1)}(u) = (\cdots, u(2), u(1), u(0), u(1), \cdots, u(N-2), u(N-1), u(N-2), u(N-3), \cdots)^t. \]

• The whole-sample antisymmetric extension (WAWA) $E_a^{(1,1)}$:
  \[ E_a^{(1,1)}(u) = (\cdots, -u(2), -u(1), u(0), u(1), \cdots, u(N-2), u(N-1), -u(N-2), -u(N-3), \cdots)^t. \]

• The whole-half-sample antisymmetric extension (WAHA) $E_a^{(1,2)}$:
  \[ E_a^{(1,2)}(u) = (\cdots, -u(2), -u(1), u(0), u(1), \cdots, u(N-2), u(N-1), -u(N-1), -u(N-2), \cdots)^t. \]

• The half-whole-sample antisymmetric extension (HAWA) $E_a^{(2,1)}$:
  \[ E_a^{(2,1)}(u) = (\cdots, -u(1), -u(0), u(0), u(1), \cdots, u(N-2), u(N-1), -u(N-2), -u(N-3), \cdots)^t. \]

Clearly, $c(E_s^{(1,1)}(u)) = (0, N - 1)$, $c(E_s^{(1,2)}(u)) = (0, N - \frac{1}{2})$, $c(E_s^{(2,1)}(u)) = (-\frac{1}{2}, N - 1)$, $c(E_a^{(1,1)}(u)) = (0, N - 1)$, $c(E_a^{(1,2)}(u)) = (0, N - \frac{1}{2})$, and $c(E_a^{(2,1)}(u)) = (-\frac{1}{2}, N - 1)$.

**Proposition 2.** Let $u = (u(0), \cdots, u(N-1))^T$ be a signal of length $N$. Then

1. $h * E_s^{(1,1)}(u)$ is symmetric with centers 0 and $N - 1$ if $h$ is a symmetric, odd length filter with $c(h) = 0$;

2. $h * E_s^{(1,1)}(u)$ is antisymmetric with centers 0 and $N - 1$ if $h$ is an antisymmetric, odd length filter with $c(h) = 0$;

3. $h * E_s^{(2,2)}(u)$ is symmetric with centers $-\frac{1}{2}$ and $N - \frac{1}{2}$ if $h$ is a symmetric, odd length filter with $c(h) = 0$;

4. $h * E_s^{(2,2)}(u)$ is antisymmetric with centers $-\frac{1}{2}$ and $N - \frac{1}{2}$ if $h$ is an antisymmetric, odd length filter with $c(h) = 0$;

5. $h * E_a^{(1,1)}(u)$ is symmetric with centers 0 and $N - 1$ if $h$ is an antisymmetric, odd length filter with $c(h) = 0$;

6. $h * E_s^{(1,2)}(u)$ is symmetric with centers $-\frac{1}{2}$ and $N - 1$ if $h$ is a symmetric, even length filter with $c(h) = \frac{1}{2}$;

7. $h * E_s^{(1,2)}(u)$ is antisymmetric with centers $-\frac{1}{2}$ and $N - 1$ if $h$ is an antisymmetric, even length filter with $c(h) = \frac{1}{2}$;

8. $h * E_a^{(2,1)}(u)$ is symmetric with centers 0 and $N - \frac{1}{2}$ if $h$ is a symmetric, even length filter with $c(h) = -\frac{1}{2}$;

9. $h * E_a^{(2,1)}(u)$ is symmetric with centers 0 and $N - \frac{1}{2}$ if $h$ is an antisymmetric, even length filter with $c(h) = -\frac{1}{2}$;

The proof of Proposition 2 is straightforward and will be omitted here.

Now, we will show how to form matrices $H_t$ and $H_s$ in (15). Without loss of generality, we show these matrices for the filters with $K = 2$ and $K = 3$ only. In a similar fashion, we can give the matrix representation for filters associated with any integer $K$.  

8
Example 3. For $K = 2$, we have the low-pass filter $h_0 = \frac{1}{4}[1, 2, 1]$ and three high-pass filters $h_1 = \frac{1}{4}[1, 0, -1]$, $h_2 = \frac{1}{4}[1, 0, -1]$, and $h_3 = \frac{1}{4}[1, -2, 1]$, respectively. If we apply the whole-point symmetric extension for the input signal, then

\[
H_0 = \frac{1}{4} \begin{bmatrix} 2 & 2 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad H_1 = \frac{1}{4} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix}
\]

\[
H_2 = H_1, \quad H_3 = \frac{1}{4} \begin{bmatrix} -2 & 2 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \\ 1 & 0 & -2 \end{bmatrix}
\]

By Items 1 and 2 of Proposition 2, matrices $H_\ell$ are

\[
H_0 = H_0, \quad H_1 = \frac{1}{4} \begin{bmatrix} 0 & 2 & 0 \\ -1 & 0 & 1 \\ \vdots & \vdots & \vdots \\ -1 & 0 & 1 \\ -2 & 0 & 0 \end{bmatrix}, \quad H_2 = H_1, \quad H_3 = H_3.
\]

If we apply the half-point extension for the input signal, then

\[
H_0 = \frac{1}{4} \begin{bmatrix} 3 & 1 & 0 & -2 \\ 1 & 2 & 1 & -1 \\ 1 & 2 & 1 & -1 \\ 1 & 3 & 0 & -1 \end{bmatrix}, \quad H_1 = \frac{1}{4} \begin{bmatrix} 1 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & -1 & 0 \end{bmatrix}
\]

\[
H_2 = H_1, \quad H_3 = \frac{1}{4} \begin{bmatrix} -1 & 1 & 0 & 1 \\ 1 & -2 & -1 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & -2 & 1 & 0 \end{bmatrix}
\]

By Items 3 and 4 of Proposition 2, matrices $H_\ell$ are

\[
H_0 = H_0, \quad H_1 = \frac{1}{4} \begin{bmatrix} 1 & 1 & 0 & 1 \\ -1 & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -1 & 0 & 0 & 1 \\ -1 & -1 & 0 & 0 \end{bmatrix}, \quad H_2 = H_1, \quad H_3 = H_0.
\]

In both the whole-point and half-point extensions, the perfect reconstruction (15) is satisfied. Furthermore, the the half-point extension for the input signal, we have

\[
H_0^T H_0 + H_1^T H_1 + H_2^T H_2 + H_3^T H_3 = I.
\]

Example 4. For $K = 3$, we have the low-pass filter $h_0 = \frac{1}{6}[1, 2, 1, 2]$ and five high-pass filters $h_1 = \frac{1}{6}[1, 0, 0, -1]$, $h_2 = \frac{\sqrt{2}}{12}[1, 1, -1, -1]$, $h_3 = \frac{\sqrt{2}}{12}[1, -1, -1, 1]$, $h_4 = \frac{\sqrt{2}}{12}[1, -1, 1, -1]$, and $h_5 = \frac{\sqrt{2}}{12}[1, -3, 3, -1]$. 
Indexes of these filters are from $-2$ to $1$. If we apply the half-point extension at the left end and the whole-point at the right end for the input signal, we have

$$H_0 = \frac{1}{6} \begin{bmatrix} 4 & 2 & 1 \\ 3 & 2 & 1 \\ 1 & 2 & 2 \\ & & \ddots \\ & & & 1 & 2 & 2 & 1 \\ & & & & 1 & 3 & 2 \end{bmatrix}$$

$$H_1 = \frac{1}{6} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 1 & 0 & 0 & -1 \\ & & & \ddots \\ & & & & 1 & 0 & 0 & -1 \end{bmatrix}$$

$$H_2 = \frac{\sqrt{6}}{12} \begin{bmatrix} 0 & 0 \\ 2 & -1 \\ 1 & 1 & -1 & -1 \\ & & \ddots & & \ddots \\ & & & 1 & 1 & -1 & -1 & 1 \\ & & & & 1 & 0 & -1 \end{bmatrix}$$

$$H_3 = \frac{\sqrt{6}}{12} \begin{bmatrix} -2 & 2 \\ 0 & -1 \\ 1 & -1 & -1 & 1 \\ & & \ddots & & \ddots \\ & & & 1 & -1 & -1 & 1 \\ & & & & 1 & 0 & -1 \end{bmatrix}$$

$$H_4 = \frac{\sqrt{3}}{3} H_3$$

$$H_5 = \frac{\sqrt{2}}{12} \begin{bmatrix} 0 & 0 \\ 3 & -2 \\ 1 & -1 \\ -1 & 3 & -3 & 1 \\ & & \ddots & & \ddots \\ & & & -1 & 3 & -4 & 1 \\ & & & & -1 & 3 & -4 & -2 & 6 \end{bmatrix}$$

### 3.4 Multi-level Framelet Decomposition and Reconstruction

To analyze the given signal via a tight frame, one needs to decompose the signal in different levels in the transform domain. This process can be accomplished through a matrix $A$ which is associated with the underlying framelet system. Since Ron and Shen's piecewise linear tight frame [42] will be used in our algorithms developed in the next section, we just give the matrix $A$ for this particular tight frame.
The piecewise linear tight frame is generated by the following low-pass \( h_0 \), bandpass \( h_1 \), and high-pass \( h_2 \) filters:

\[
\begin{align*}
  h_0 &= \frac{1}{4}[1, 2, 1], \\
  h_1 &= \frac{\sqrt{2}}{4}[1, 0, -1], \quad \text{and} \\
  h_2 &= \frac{1}{4}[1, -2, 1].
\end{align*}
\]

The scaling function \( \phi \) and the wavelets \( \psi_1 \) and \( \psi_2 \), associated with \( h_0 \), \( h_1 \), and \( h_2 \), respectively, are given in the Fourier domain by

\[
\begin{align*}
  \hat{\phi}(\omega) &= \frac{\sin^2(\omega/2)}{(\omega/2)^2}, \\
  \hat{\psi}_1(\omega) &= i\sqrt{2}\cos(\omega/4)\sin^3(\omega/4)/((\omega/4)^2), \quad \text{and} \\
  \hat{\psi}_2(\omega) &= -\sqrt{2}\sin^4(\omega/4)/((\omega/4)^2).
\end{align*}
\]

For any non-negative integer \( \ell \), we define

\[
\begin{align*}
  h_{0,\ell} &:= \frac{1}{4}[1, 0, \ldots, 0, 2, 0, \ldots, 0, 1] \\
  h_{1,\ell} &:= \frac{\sqrt{2}}{4}[1, 0, \ldots, 0, 0, \ldots, 0, -1] \\
  h_{2,\ell} &:= \frac{1}{4}[1, 0, \ldots, 0, -2, 0, \ldots, 0, 1]
\end{align*}
\]

Following Example 3 with a half-point symmetric extension, we define matrices \( H_0^{(\ell)} \), \( H_1^{(\ell)} \), and \( H_2^{(\ell)} \) associated with filters \( h_{0,\ell} \), \( h_{1,\ell} \), and \( h_{2,\ell} \), respectively. Further, we have

\[
(H_0^{(\ell)})^T H_0^{(\ell)} + (H_1^{(\ell)})^T H_1^{(\ell)} + (H_2^{(\ell)})^T H_2^{(\ell)} = I.
\]

Now, the matrix \( A \) corresponding to the \( L \)-level framelet decomposition is

\[
A := \begin{bmatrix}
  H_0^{(L-1)} \prod_{\ell=0}^{L-2} H_0^{(L-\ell)} \\
  H_1^{(L)} \prod_{\ell=0}^{L-2} H_0^{(L-\ell)} \\
  H_2^{(L)} \prod_{\ell=0}^{L-2} H_0^{(L-\ell)} \\
  \vdots \\
  H_1^{(1)} \\
  H_2^{(1)}
\end{bmatrix}.
\]

(16)

Obviously, the \( L \)-level perfect reconstruction formula is

\[
A^T A = I.
\]

We remark that a different matrix \( A \) is used in our Algorithm III proposed in the next section.

### 4 Algorithms

For a \( K \times K \) sensor array, the filters \( \{h_1\}_{i=0}^r \), \( r = 2K - 1 \), associated with the high-resolution image reconstruction are given by (12). With a proper assumption about boundary conditions, we have matrices \( \{H_i\}_{i=0}^r \) corresponding to \( \{h_i\}_{i=0}^r \), respectively. These matrices satisfy the perfect reconstruction formula as follows:

\[
\sum_{i=0}^r H_i^T H_i = I.
\]

(17)

We emphasize it again that (17) is the matrix representation of (10) with \( \gamma = 0 \) for the tight frame system arising from the high-resolution image reconstruction.
For the problem of high-resolution image reconstruction, the model (6) is
\[ g = H_0 f + n \]
where \( n \) is noise. With one step further, we consider a slightly more general restoration problem
\[ g = H_s f + n \] (18)
for a certain number \( s \) between 0 and \( r \).

We propose two different types of algorithms for solving the problem (18). The algorithm of the first type is derived directly from the identity (17). It is basically the Landweber algorithm and will be presented in subsection 4.1. The algorithms of second type are modified versions of the algorithm of the first type by incorporating various denoising techniques. As we will see in subsection 4.2, some of algorithms are already appeared in our previous papers [9, 10, 11, 12, 13], but, theoretical results on the convergence of those algorithms given here are new.

### 4.1 Basic Algorithm

Multiplying a vector \( f \) from both sides of the perfect reconstruction formula (17) yields
\[ f = H_s^T H_s f + \sum_{i \neq s} H_i^T H_i f. \] (19)

By substituting the known data \( H_s f \approx g \) in (18) into (19), we obtain an iteration as
\[ f^{k+1} = H_s^T g + \sum_{i \neq s} H_i^T H_i f^k. \] (20)

This is our basic algorithm. The proposed algorithms in the following subsection are all modification of (20) by incorporating different denoising schemes. Algorithm (20) is, in fact, no other than a Landweber’s iteration. For the completeness, we give the convergence of (20) in the following theorem.

**Theorem 1.** The sequence \( f^k \) generated by (20) with initial guess \( f^0 \) converges to a solution of
\[ \begin{align*}
\min_f & \| f - f^0 \|_2, \\
\text{s.t.} & \ H_s f = g.
\end{align*} \] (21)

**Proof.** First we write out the singular value decomposition of \( H_s \), that is,
\[ H_s = U_s \Sigma_s V_s^T, \]
where \( U_s \) and \( V_s \) are orthogonal matrices and \( \Sigma_s \) is a diagonal matrix. Without loss of generality, we assume that
\[ \Sigma_s = \begin{bmatrix} 
\Sigma_{s,1} & 0 \\
0 & 0 
\end{bmatrix} \]
with \( \Sigma_{s,1} \) being an invertible diagonal matrix. The pseudo-inverse \( H_s^\dagger \) of \( H_s \) reads
\[ H_s^\dagger = V_s \Sigma_{s,1}^{-1} U_s^T, \]
where the pseudo-inverse \( \Sigma_{s,1}^\dagger \) of \( \Sigma_{s,1} \) is given by
\[ \Sigma_{s,1}^\dagger = \begin{bmatrix} 
\Sigma_{s,1}^{-1} & 0 \\
0 & 0 
\end{bmatrix}. \]

Corresponding to the structure of \( \Sigma_s \) or \( \Sigma_{s,1}^\dagger \), we can have a partition of any vector, say \( f \), as follows
\[ f = \begin{bmatrix} f_1 \\
f_2 \end{bmatrix}. \]
where the dimension of \( f_1 \) is the same as the number of columns of \( \Sigma_{s,1} \).

Now, we turn to the minimization problem (21). The constrained condition \( H f = g \) implies

\[
\Sigma_s V_s^T f = U_s^T g.
\]

It, by the structure of \( \Sigma_s \), says that

\[
(V_s^T f)_1 = \Sigma_{s,1}^{-1} (U_s^T g)_1.
\]  

(22)

By using (22), we have

\[
\| f - f^0 \|_2^2 = \| V_s^T f - V_s^T f^0 \|_2^2 = \| (V_s^T f)_1 - (V_s^T f^0)_1 \|_2^2 + \| (V_s^T f)_2 - (V_s^T f^0)_2 \|_2^2.
\]

(23)

Obviously, to minimize \( \| f - f^0 \|_2^2 \) with the constrained condition \( H_s f = g \), the minimizer of (21), denoted by \( f^* \), should satisfy the following conditions

\[
(V_s^T f^*)_1 = \Sigma_{s,1}^{-1} (U_s^T g)_1 \quad \text{and} \quad (V_s^T f^*)_2 = (V_s^T f^0)_2.
\]  

Next, let’s look at the iterative algorithm (20). Since \( \sum_{s \neq s} H_s^T H_s = V_s (I - \Sigma_s^2) V_s^T \), then (20) becomes

\[
f^{k+1} = V_s \Sigma_s U_s^T g + V_s (I - \Sigma_s^2) V_s^T f^k,
\]

which is equivalent to

\[
V_s^T f^{k+1} = \Sigma_s U_s^T g + (I - \Sigma_s^2) V_s^T f^k.
\]  

(24)

We can split (24) as

\[
\begin{cases}
(V_s^T f^{k+1})_1 = \Sigma_{s,1} (U_s^T g)_1 + (I - \Sigma_{s,1}^2) (V_s^T f^k)_1, \\
(V_s^T f^{k+1})_2 = (V_s^T f^k)_2
\end{cases}
\]  

(25)

Notice that the two iterations for \( (V_s^T f^k)_1 \) and \( (V_s^T f^k)_2 \) respectively are independent to each other. In the first equation of (25), the absolute values of all non-zero elements of the \( \Sigma_{s,1} \) are strictly greater than 0 and less than or equal to 1. Therefore, the first iteration in (24) is a contract mapping, hence converges. Its limit \( (V_s^T f^*)_1 \) is the unique fix point of the first iteration in (25). More precisely, \( (V_s^T f^*)_1 \) satisfies

\[
(V_s^T f^*)_1 = \Sigma_{s,1} (U_s^T g)_1 + (I - \Sigma_{s,1}^2) (V_s^T f^*)_1,
\]

which is equivalent to

\[
(V_s^T f^*)_1 = \Sigma_{s,1}^{-1} (U_s^T g)_1.
\]  

(26)

On the other hand, the second iteration in (25) converges obviously to

\[
(V_s^T f^*)_2 = (V_s^T f^0)_2
\]  

(27)

Combining (26) and (27) together, we obtain (23). In other words, the limit of (20) is the solution of the minimization problem (21).

\[ \square \]

Since (20) is a Landweber’s iteration, it has a regularization property known as semiconvergence: the iterates \( f^k \) first approach the true image, but the noise will be amplified when \( k \) is larger than a certain threshold. Thus a stopping criterion, called the discrepancy principle, has to be introduced in order to obtain the best approximation of the required solution. Furthermore, the regularization property of projected Landweber’s iterations is related to Tikhonov regularization, hence the edges in the underlying image are smeared. We further remark that the solution of (21) depends on the initial seed \( f^0 \).
4.2 Algorithms with Tight Frame Denoising Scheme

In this subsection, we will incorporate nonlinear denoising schemes into the iterative algorithm (20) with the aim of improving the quality of the reconstructed images. The Donoho’s soft thresholding operator is adopted in our nonlinear denoising schemes. The soft-thresholding operator is given as follows

\[ T_\lambda(u) = [t_{\lambda_1}(u_1), \ldots, t_{\lambda_n}(u_n)]^T \]

with \( \lambda \) being a pre-given vector having non-negative components and

\[ t_{\lambda_j}(u_j) = \begin{cases} 
0, & \text{if } |u_j| \leq \lambda_j, \\
\text{sgn}(u_j)(|u_j| - \lambda_j), & \text{if } |u_j| > \lambda_j.
\end{cases} \]

In what follows, the matrix \( A \) of the multilevel non-downsampled framelet decomposition is defined by (16) for the first two algorithm although it could be a matrix from any other tight frame system. We have following three modified algorithms.

4.2.1 Algorithm I

For framelet coefficients \( H_i f^k \) of \( f^k \) in (20), we modify \( H_i f^k \) into \( A^T T_{\lambda_i}(AH_i f^k) \) by using the multilevel non-downsampled framelet decomposition matrix \( A \) and the soft-thresholding operator \( T_{\lambda_i} \). The resulting algorithm, called Algorithm I, from (20) is given as

\[ f^{k+1} = H_s^T g + \sum_{i \neq s} H_i^T A^T T_{\lambda_i}(AH_i f^k). \]

This algorithm was proposed and studied in [5, 8, 10, 11, 12, 13]. However, a complete analysis for this algorithm is not available. We will prove the convergence of this algorithm in the next section, and give minimization properties of its limit.

4.2.2 Algorithm II

We incorporate a frame based denoising scheme into each iterate of (20), and obtain Algorithm II as

\[ f^{k+1} = A^T T_{\lambda} A \left( H_s^T g + \sum_{i \neq s} H_i^T H_i f^k \right). \]

It was proposed and studied in [7, 8, 11], and can be seen as an extension of the algorithm in [16] from orthonormal system to tight frame systems; another possible extension can be found in [18]. The following convergence theorem for (31) is proved in [7, 8].

**Theorem 2.** Let \( A \) be a tight frame. Let \( H_i, 0 \leq i \leq r \), be satisfying (19). Define

\[ \alpha^k = T_{\lambda} \left( H_s^T g + \sum_{i \neq s} H_i^T H_i f^k \right), \]

where \( f^k \) is generated by (31). Then \( \alpha^k \) converges to a solution of

\[ \min_{\alpha} \left\{ \frac{1}{2} \| H_s A^T \alpha - g \|^2_2 + \frac{1}{2} \|(I - AA^T)\alpha\|^2_2 + \| \text{diag}(\lambda) \alpha \|_1 \right\}. \]

Following [7, 8], the role of each terms in (33) can be explained as follows. The first term is the data fidelity, and the third term is to ensure the sparsity of the tight frame coefficient. The second term measures the distance from \( \alpha \) to the range of \( A \). By the framelet theory in [2, 24], if the coefficient \( \alpha \) is in the range of \( A \), then the (weighted) \( \ell_1 \) norm \( \alpha \) is equivalent to some Besov norm of the image \( A^T \alpha \). Therefore, the second term links the (weighted) \( \ell_1 \) norm to the regularity of the restored image in some sense. Combining all together, the minimization problem (33) fits the known data, and balances the sparsity of the frame coefficient and the regularity of restored image.
4.2.3 Algorithm III

With the same spirit of constructing matrix $A$ using the piecewise linear tight frame in subsection 3.4, we design a matrix using the filters (12) associated with high-resolution image reconstruction. Such matrix, denoted by $\tilde{A}$, can be written into

$$\tilde{A} = [\tilde{A}_{-1-L}H_0, H_1, \ldots, H_s]^T,$$  \hfill (34)

where $\tilde{A}_{-1-L}$ is the framelet decomposition from level $-1$ to $L$. We consider the case that $s = 0$ and $s \neq 0$ respectively.

- If $s = 0$, then we have

$$\tilde{A}f = [\tilde{A}_{-1-L}H_0f, H_1f, \ldots, H_rf]^T = [\tilde{A}_{-1-L}g, H_1f, \ldots, H_rf]^T.$$  

We obtain the algorithm

$$f^{k+1} = \tilde{A}^T T_\lambda [\tilde{A}_{-1-L}g, H_1f^k, \ldots, H_rf^k]^T.$$  \hfill (35)

- If $s \neq 0$, then we have

$$\tilde{A}f = [\tilde{A}_{-1-L}H_0f, H_1f, \ldots, H_sf]^T = [\tilde{A}_{-1-L}H_0f, H_1f, \ldots, H_{s-1}f, g, H_{s+1}f, \ldots, H_rf]^T.$$  

We obtain the algorithm

$$f^{k+1} = \tilde{A}^T T_\lambda [\tilde{A}_{-1-L}H_0f^k, H_1f^k, \ldots, H_{s-1}f^k, g, H_{s+1}f^k, \ldots, H_rf^k]^T.$$  \hfill (36)

This algorithm can be found in [6, 7]. The following results are proved in [7] for the convergence of (35) and (36).

**Theorem 3.** Let $\tilde{A}$ be a tight frame in the form of (34). Define

$$\alpha^k = T_\lambda [\tilde{A}_{-1-L}g, H_1f^k, \ldots, H_rf^k]^T,$$

where $f^k$ is generated by (35). Then $\alpha^k$ converges to a solution of

$$\min_{\alpha \in C_0} \{ ||(I - \tilde{A}^T)\alpha||^2 + \|\text{diag}(\lambda)\alpha\|_1 \},$$  \hfill (37)

where $C_0 = \{ \alpha : \alpha|_\Gamma = T_\lambda |_{\Gamma} \tilde{A}_{-1-L}g \}$ with $\Gamma$ being the positions corresponding to $\tilde{A}_{-1-L}H_0$ in $\tilde{A}$.

**Theorem 4.** Let $\tilde{A}$ be a tight frame in the form of (34). Define

$$\alpha^k = T_\lambda [\tilde{A}_{-1-L}H_0f^k, H_1f^k, \ldots, H_{s-1}f^k, H_{s+1}f^k, \ldots, H_rf^k]^T,$$

where $f^k$ is generated by (35). Then $\alpha^k$ converges to a solution of

$$\min_{\alpha \in C_s} \{ ||(I - \tilde{A}^T)\alpha||^2 + \|\text{diag}(\lambda)\alpha\|_1 \},$$  \hfill (38)

where $C_s = \{ \alpha : \alpha|_\Gamma = T_\lambda |_{\Gamma} g \}$ with $\Gamma$ being the positions corresponding to $H_s$ in $\tilde{A}$.

Again, the minimization problems (37) and (38) fit the data, and balance the sparsity of the frame coefficient and the regularity of the restored image. The data fidelity comes from two aspects. On the one hand, it is obvious that the constraints $\alpha \in C_0$ in (37) and $\alpha \in C_s$ in (38) fit the data. On the other hand, the first term $||(I - \tilde{A}^T)\alpha||^2$ in (37) and (38) also measures the data fidelity, since small $||(I - \tilde{A}^T)\alpha||^2$ implies small $||(I - \tilde{A}^T)\alpha|_\Gamma||^2 \approx \|g - H_s\tilde{A}^T\alpha\|^2$. The balance of the sparsity and the regularity comes from the first and second terms in (37) and (38) as we have explained for (33).
5 Analysis of Algorithm I

In this section, we focus on the convergence of algorithm (30). The main tool that we will use is an iterative algorithm of forward-backward splitting based on proximity operators.

5.1 Proximal Forward-Backward Splitting

We first give a brief review of an iterative algorithm of forward-backward splitting based on proximity operators. This will be used in our analysis. The forward-backward splitting algorithm can be rooted back to 1950’s for the study of numerical solution for partial differential equations. Here we are interested in a forward-backward splitting algorithm in [15] based on proximity operator. For any proper, lower semicontinuous, convex function \( F(x) \) where \( x \in \mathbb{R}^n \), the proximity operator prox \( F \) [34, 35] is defined by

\[
\forall y \in \mathbb{R}^n, \quad \text{prox}_F(y) = \arg \min_x \left\{ \frac{1}{2} \|x - y\|^2_2 + F(x) \right\},
\]

and Moreau’s envelope, or Moreau-Yosida regularization [26], env \( F \) is the continuously differentiable function

\[
\forall y \in \mathbb{R}^n, \quad \text{env}_F(y) = \min_x \left\{ \frac{1}{2} \|x - y\|^2_2 + F(x) \right\}.
\]

There is an important relation between the proximity operator and the envelope function, i.e.,

\[
\nabla \text{env}_F(y) = y - \text{prox}_F(y).
\]

Furthermore, both \( \nabla \text{env}_F \) and \( \text{prox}_F \) are all Lipshitz continuous with Lipshitz constant 1, i.e.,

\[
\| (x - \text{prox}_F(x)) - (y - \text{prox}_F(y)) \|_2 \leq \| x - y \|_2, \quad \forall x, y \in \mathbb{R}^n,
\]

\[
\| \text{prox}_F(x) - \text{prox}_F(y) \|_2 \leq \| x - y \|_2, \quad \forall x, y \in \mathbb{R}^n.
\]

The proximal forward-backward splitting in [15] is to solve the minimization problem

\[
\min_x \{ F_1(x) + F_2(x) \}.
\]

Here \( F_1 \) and \( F_2 \) are all proper, lower semicontinuous, convex functions. Moreover, the function \( F_2 \) is continuously differentiable, and its gradient is Lipshitz-continuous, i.e.,

\[
\| \nabla F_2(x) - \nabla F_2(y) \|_2 \leq \frac{1}{c} \| x - y \|_2.
\]

With all the above settings, the iteration for solving (43) is

\[
x^{k+1} = \text{prox}_{dF_2}(x^k - d\nabla F_2(x^k)),
\]

where \( 0 < d < 2c \). The convergence theory for (45) given in [15] is summarized as the following theorem.

**Theorem 5.** Suppose that \( F_1 \) and \( F_2 \) satisfy

1. \( F_1(x) + F_2(x) \) is coercive, i.e., whenever \( \|x\|_2 \to +\infty \), \( F_1(x) + F_2(x) \to +\infty \);

2. \( F_1 \) is a proper, convex, lower semi-continuous function; and

3. \( F_2 \) is a proper, convex, differentiable function satisfying (44) and \( 0 < d < 2c \).

Then there exists at least one solution of (43), and for any initial guess \( x^0 \) the iteration (45) converges to a solution of (43).
5.2 Convergence of Algorithm I

Let

\[ B = [AH_0; AH_1; \ldots; AH_r]. \]  

(46)

By UEP and \( A^T A = I \), we have \( B^T B = I \). Therefore, the row vectors of \( B \) forms a tight frame. Since \( A \) is a framelet decomposition operator, \( B \) is a framelet package decomposition operator. Let

\[ \alpha^k = [AH_0 f^k; \ldots; AH_{s-1} f^k; g; AH_{s+1} f^k; \ldots; AH_r f^k]. \]

be the coefficients under the framelet package system \( B \).

Define a new sequence of \((r+1)\)-tuple as

\[ \alpha^k = (T_\lambda AH_0 f^k, T_\lambda AH_1 f^k, \ldots, T_\lambda AH_{s-1} f^k, g, T_\lambda AH_{s+1} f^k, \ldots, T_\lambda AH_r f^k), \]

and its \(i\)-th entry, \(i = 0, 1, \ldots, r\), are subscripted by \(i\), i.e., \( \alpha^k_s = g \) and \( \alpha^k_i = AH_i f^k \) for \( i \neq s \) and \( i = 1, \ldots, r \).

We write a new matrix \( B \) as

\[ B = [AH_0; AH_1; \ldots; AH_{s-1}; H_s; AH_{s+1}; \ldots; AH_r]^T. \]

Note that \( B \) and \( B^T \) correspond to a framelet packet decomposition and reconstruction operator respectively, see [8]. Since \( A^T A = I \) and \( \sum_{i=0}^{\infty} H_i^T H_i = I \), we have \( B^T B = I \). By (30) and (47), we obtain

\[ f^{k+1} = B^T \alpha^k. \]

(48)

Lemma 1. The sequence \( \alpha^k \) converges if and only if the sequence \( f^k \) does.

Proof. Since all the operators involved in (47) and (48) are all continuous, the lemma follows immediately. \( \square \)

Therefore, we show that \( f^k \) converges by showing that \( \alpha^k \) converges. The strategy is the following: we first transform the iteration into a proximal forward-backward splitting iteration for a special functional; then by applying Theorem 5, we obtain the convergence for the sequence \( \alpha^k \).

Let \( \mathcal{C} = \{ \alpha | \alpha_s = g \} \) which is a subset \( \mathcal{C} \) of the \((r+1)\)-tuple. It is obviously a closed nonempty convex set. We define the indicator function of \( \mathcal{C} \) by

\[ D_{\mathcal{C}}(\alpha) = \begin{cases} 0, & \alpha \in \mathcal{C}, \\ \infty, & \alpha \notin \mathcal{C}. \end{cases} \]

Lemma 2. The sequence \( \alpha^k \) defined in (47) generated by algorithm (30) is equivalent to that generated by a proximal forward-backward splitting iteration (45) where \( d = 1 \) for

\[ \min_{\alpha \in \mathcal{C}} \left\{ ||(I - BB^T)\alpha||_2^2 + \sum_{i \neq s} ||\text{diag}(\lambda)\alpha_i||_1 \right\} \]

(49)

with

\[ F_1(\alpha) = D_{\mathcal{C}}(\alpha) + \sum_{i \neq s} ||\text{diag}(\lambda)\alpha_i||_1, \quad \text{and} \quad F_2(\alpha) = ||(I - BB^T)\alpha||_2^2. \]

Proof. By (30) and (47), we have

\[ \alpha^{k+1}_s = g, \quad \text{and} \quad \alpha^{k+1}_i = T_\lambda [BB^T \alpha^k]_i, \quad i \neq s. \]

(50)

Here, \([\cdot]_i\) denotes the \(i\)-th entry of the \((r+1)\)-tuple. It is clear that

\[ \nabla F_2(\alpha) = \alpha - BB^T \alpha. \]

(51)

Comparing (50) and (51) with (45) where \( d = 1 \), we see that if we can prove that

\[ \text{prox}_{F_1}(\alpha) = (T_\lambda \alpha_0, \ldots, T_\lambda \alpha_{s-1}, g, T_\lambda \alpha_{s+1}, \ldots, T_\lambda \alpha_r), \]

(52)
then we have done. We verify (52) by considering the definition of the proximal operator in the following

\[ \text{prox}_{F_1}(\alpha) = \arg \min_{\beta} \left\{ \frac{1}{2} \|\beta - \alpha\|^2 + \frac{1}{2} D_C(\beta) + \frac{1}{2} \sum_{i \neq s} \| \text{diag}(\lambda) |\beta_i|_1 \| \right\}, \quad (53) \]

Note that \(D_C(\alpha) = \sum_{i=1}^n D_{g_i}(\alpha_{i,j})\), where \(g_i(\alpha_{i,s}) = 0\) if \(g_i = \alpha_{i,s}\) and \(D_{g_i}(\alpha_{i,j}) = \infty\) otherwise. Therefore, by (53),

\[ [\text{prox}_{F_1}(\alpha)]_{i,s} = \arg \min_{\beta_{i,s}} \left\{ \frac{1}{2} (\beta_{i,s} - \alpha_{i,s})^2 + \frac{1}{2} D_{g_i}(\beta_{i,s}) \right\}, \quad i = 1, \ldots, n, \]

and

\[ [\text{prox}_{F_1}(\alpha)]_{i,j} = \arg \min_{\beta_{i,j}} \left\{ \frac{1}{2} (\beta_{i,j} - \alpha_{i,j})^2 + \frac{1}{2} \lambda_i |\beta_{i,j}| \right\}, \quad j \neq s. \]

The above two equations are identical to (52).

By applying Theorem 5, we get that the sequence \(\alpha^k\) converges to a minimizer of (49).

**Theorem 6.** The sequence \(\alpha^k\) generated by (47) converges to a solution of (49).

**Proof.** We note that the set \(C\) is a closed non-empty convex set, hence \(D_C\) is a proper lower semi-continuous convex function. Therefore, both the function \(F_1\) and \(F_2\) are proper, semi-continuous and convex, and \(F_2\) is differentiable. The gradient of \(F_2\) is Lipschitz continuous with Lipschitz constant 1. Indeed, for any \((r+1)\)-tuple \(\alpha\) and \(\beta\), we have

\[ \| \nabla F_2(\alpha) - \nabla F_2(\beta) \|_2 = \| (I - BB^T)(\alpha - \beta) \|_2 \leq \| I - BB^T \|_2 \| \alpha - \beta \|_2. \]

Since the operator \((I - BB^T)\) satisfies \((I - BB^T)^2 = (I - BB^T)\) and \((I - BB^T)^T = (I - BB^T)\), it is an orthonormal projector, hence its norm is 1. Therefore, \(\nabla F_2\) is Lipschitz continuous with Lipschitz constant 1, hence \(c = 1\). Therefore, \(d = 1\) satisfies the condition in Theorem 5. Thus, by Theorem 5, if we can prove the existence of the minimizers for (49), we get the lemma.

We show the existence by showing the coercivity of \(F_1\) hence of \(F_1 + F_2\). If \(\alpha \in C\), then we have

\[ F_1(\alpha) = \sum_{i \neq s} \| \text{diag}(\lambda) \alpha_i \|_1 \geq \min_j \lambda_j \left( \sum_{i \neq s} \| \alpha_i \|^2 \right)^{1/2} = \min_j \lambda_j (\| \alpha \|^2 - \| g \|^2). \]

If \(\alpha \notin C\), then it is clear \(F_1(\alpha) = \infty\). Therefore, as \(\| \alpha \|_2 \to \infty\), \(F_1(\alpha) \to \infty\). Hence, \(F_1\) is coercive.

5.3 Minimization Property of Algorithm I

We have proven that the limit \(\alpha^k\) generated by (47) is a solution of (49). Let \(f^*\) and \(\alpha^*\) be the limit of the sequence \(f^k\) and \(\alpha^k\) respectively. Then by (47), we have that \(f^* = B^T \alpha^*\). The following lemma explains the minimization property of (49). It states the optimality property of the pair \(\{f^*, \alpha^*\}\) in the sense that, if \(f^*\) is perturbed by \(e\) and \(\alpha^*\) is perturbed by \(Be\), then the energy \(\|H_s f^* - g\|^2 + \sum_{i \neq s} \| \text{diag}(\lambda) \alpha_{i}^* \|_1\) increases. The first term in the energy is a data fidelity, and the second term is a weighted \(\ell_1\) norm which leads to the sparsity.

**Lemma 3.** For any vector \(e \in \mathbb{R}^n\), we have

\[ \| H_s (f^* + e) - g \|^2 + \sum_{i \neq s} \| \text{diag}(\lambda) (AH_i e + \alpha_{i}^*) \|_1 \geq \| H_s f^* - g \|^2 + \sum_{i \neq s} \| \text{diag}(\lambda) \alpha_{i}^* \|_1. \]

**Proof.** Let

\[ \gamma := (AH_0 e, \ldots, AH_{s-1} e, 0, AH_{s+1} e, \ldots, AH_r e). \]

Since \(\alpha^*\) is a minimizer of (49), we have

\[ \| (I - BB^T)(\gamma + \alpha^*) \|^2 + \sum_{i \neq s} \| \text{diag}(\lambda) (AH_i e + \alpha_{i}^*) \|_1 \geq \| (I - BB^T)\alpha^* \|^2 + \sum_{i \neq s} \| \text{diag}(\lambda) \alpha_{i}^* \|_1 \]

(55)
By the definition of $B$, the $(r + 1)$-tuple
$$\delta = (AH_0e, \ldots, AH_{s-1}e, H_s e, AH_{s+1}e, \ldots, AH_r e)$$
satisfies $\delta = Be$. Therefore, by the identity $B^T Be = e$ we obtain
$$(I - BB^T)\delta = 0,$$  \hspace{1cm} (56)
which implies that
$$\| (I - BB^T)(\gamma + \alpha^*) \|_2^2 = \| (I - BB^T)(\gamma - \delta + \alpha^*) \|_2^2.$$  \hspace{1cm} (57)
Notice that
$$B^T(\gamma - \delta) = H_s^T H_s e.$$  \hspace{1cm} (58)
Substituting (58) and the definition of $f^* = A^T \alpha^*$ into (57), we have that
$$\| (I - BB^T)(\gamma + \alpha^*) \|_2^2 = \| (\alpha^* - Bf^*) + (\gamma - \delta) - BH_s^T H_s e \|_2^2,$$  \hspace{1cm} (59)
which is equivalent to
$$\| (I - BB^T)(\gamma + \alpha^*) \|_2^2 = \| (\alpha^* - Bf^*) + (\gamma - \delta) \|_2^2 + \| BH_s^T H_s e \|_2^2 - 2\langle (\alpha^* - Bf^*) + (\gamma - \delta), BH_s^T H_s e \rangle$$
$$= \| (\alpha^* - Bf^*) + (\gamma - \delta) \|_2^2 + \| BH_s^T H_s e \|_2^2 - 2\langle B^T (\alpha^* - Bf^*) + B^T (\gamma - \delta), H_s^T H_s e \rangle.$$  \hspace{1cm} (60)
Since $B^T B = I$, we have
$$\| BH_s^T H_s e \|_2^2 = \| H_s^T H_s e \|_2^2$$  \hspace{1cm} (61)
and
$$B^T (\alpha^* - Bf^*) = B^T \alpha^* - f^* = 0.$$  \hspace{1cm} (62)
Substituting (61) and (62) into (60), we get
$$\| (I - BB^T)(\gamma + \alpha^*) \|_2^2 = \| (\alpha^* - Bf^*) + (\gamma - \delta) \|_2^2 + \| H_s^T H_s e \|_2^2 - 2\langle B^T (\gamma - \delta), H_s^T H_s e \rangle,$$  \hspace{1cm} (63)
which together with (58) implies
$$\| (I - BB^T)(\gamma + \alpha^*) \|_2^2 = \| (\alpha^* - Bf^*) + (\gamma - \delta) \|_2^2 + \| H_s^T H_s e \|_2^2 - 2\langle H_s^T H_s e, H_s^T H_s e \rangle$$
$$= \| (\alpha^* - Bf^*) + (\gamma - \delta) \|_2^2 - \| H_s^T H_s e \|_2^2.$$  \hspace{1cm} (63)
Since the first term in the last equality can be rewritten into
$$\| (\alpha^* - Bf^*) + (\gamma - \delta) \|_2^2 = \sum_{i \neq s} \| \alpha^*_i - AH_i f^* \|_2^2 + \| g - H_s (f^* + e) \|_2^2,$$
we obtain
$$\| (I - BB^T)(\gamma + \alpha^*) \|_2^2 = \sum_{i \neq s} \| \alpha^*_i - AH_i f^* \|_2^2 + \| g - H_s (f^* + e) \|_2^2 - \| H_s^T H_s e \|_2^2.$$  \hspace{1cm} (64)
On the other hand, we have
$$\| (I - BB^T)\alpha^* \|_2^2 = \sum_{i \neq s} \| \alpha^*_i - AH_i f^* \|_2^2 + \| g - H_s f^* \|_2^2.$$  \hspace{1cm} (65)
Substituting (64) and (65) into (56), we obtain
$$\| g - H_s (f^* + e) \|_2^2 - \| H_s^T H_s e \|_2^2 + \sum_{i \neq s} \| \text{diag}(\lambda)(AH_i e + \alpha^*_i) \|_1 \geq \| g - H_s f^* \|_2^2 + \sum_{i \neq s} \| \text{diag}(\lambda)\alpha^*_i \|_1,$$
which obviously implies (49).
6 Numerical Experiments

We now provide numerical experiments of the presented Algorithm I, Algorithm II, and Algorithm III for high-resolution image reconstruction. The algorithms are evaluated by using peak signal-to-noise ratio (PSNR) which compares the reconstructed image $f_r$ with the original image $f$. It is defined by

$$10 \log_{10} \frac{255^2 M_1 M_2}{\|f - f_r\|^2_2}$$

where the size of the reconstructed images is $M_1 \times M_2$.

We use the “Cameraman” and “Bridge” images of size $256 \times 256$ as the original images, see Figure 2. The maximum number of iteration is set to 100 and the iteration processes of Algorithm I, Algorithm II, and Algorithm III is stopped when the reconstructed images achieve the highest PSNR values.

For a $2 \times 2$ sensor array, the observed images are shown in the top row of Figure 3. The reconstructed “Bridge” images (from the second row to the fourth row, the left column of Figure 3) via Algorithm I, Algorithm II, and Algorithm III have PSNR values of 26.10 dB, 26.63 dB, and 25.97 dB, respectively. Likewise, the reconstructed “Cameraman” images (from the second row to the fourth row, the right column of Figure 3) via Algorithm I, Algorithm II, and Algorithm III have PSNR values of 31.75 dB, 31.16 dB, and 31.68 dB, respectively.

The similar experiments are conducted for “Bridge” and “Cameraman” images in $4 \times 4$ sensor array. The reconstructed “Bridge” images (from the second row to the fourth row, the left column of Figure 4) via Algorithm I, Algorithm II, and Algorithm III have PSNR values of 23.71 dB, 24.00 dB, and 23.45 dB, respectively. Likewise, the reconstructed “Cameraman” images (from the second row to the fourth row, the right column of Figure 3) via Algorithm I, Algorithm II, and Algorithm III have PSNR values of 27.56 dB, 27.38 dB, and 27.56 dB, respectively.

In summary, the proposed algorithms have produced very similar results for high-resolution image reconstruction in terms of the PSNR values and the visual quality of the reconstructed images. However, from the viewpoint of computational efficiency, we recommend to choose Algorithm II for the problem of high-resolution image reconstruction.

References


Figure 2: Original “Bridge” image (left) and original “Cameraman” images (right).
Figure 3: $2 \times 2$ sensor array. Left column (from top to bottom) shows the observed “Bridge” image, reconstructed images by Algorithm I, Algorithm II, and Algorithm III with PSNR values of 26.10dB, 26.63dB, and 25.97 dB, respectively. Likewise, right column (from top to bottom) shows the observed “Cameraman” image, reconstructed images by Algorithm I, Algorithm II, and Algorithm III with PSNR values of 31.75dB, 31.16dB, and 31.68 dB.
Figure 4: $4 \times 4$ sensor array. Left column (from top to bottom) shows the observed “Bridge” image, reconstructed images by Algorithm I, Algorithm II, and Algorithm III with PSNR values of 23.71 dB, 24.00 dB, and 23.45 dB, respectively. Likewise, right column (from top to bottom) shows the observed “Cameraman” image, reconstructed images by Algorithm I, Algorithm II, and Algorithm III with PSNR values of 27.56 dB, 27.38 dB, and 27.56 dB.