Simultaneously Inpainting in Image and Transformed Domains

Jian-Feng Cai[*] Raymond H. Chan[†] Lixin Shen[‡] Zuowei Shen[§]

Abstract

In this paper, we focus on the restoration of images that have incomplete data in either the image domain or the transformed domain or in both. The transform used can be any orthonormal or tight frame transforms such as orthonormal wavelets, tight framelets, the discrete Fourier transform, the Gabor transform, the discrete cosine transform, and the discrete local cosine transform. We propose an iterative algorithm that can restore the incomplete data in both domains simultaneously. We prove the convergence of the algorithm and derive the optimal properties of its limit. The algorithm generalizes, unifies, and simplifies the inpainting algorithm in image domains given in [8] and the inpainting algorithms in the transformed domains given in [7,16,19]. Finally, applications of the new algorithm to super-resolution image reconstruction with different zooms are presented.

1 Introduction

In many problems in image processing, the observed data are often incomplete in the sense that features of interest in the image or some coefficients of the image under certain transforms (such as the wavelet transform) are missing or corrupted by noise. We are required to reconstruct the true image or an approximation of it from the given incomplete data. The main challenge is to find a solution that is close to the given observed data, preserves the edges in the true image, and has some preferred regularities.

We will denote images as vectors in $\mathbb{R}^N$ by concatenating their columns. Let the original image $\mathbf{f}$ be in $\mathbb{R}^N$. Let $\mathbf{A}$ be a transform, normally an $M \times N$ matrix. We suppose that $\mathbf{A}^T\mathbf{A} = \mathbf{I}$. This condition is equivalent to that the rows of the matrix $\mathbf{A}$ form a set of tight frames in $\mathbb{R}^N$. Here we assume that $\mathbf{A}$ is real-valued for simplicity. For the more general case, the discussion here can be modified straightforwardly. The transform $\mathbf{A}$ is chosen such that the true image has a sparse approximation in its transformed domain. There are many transforms that can be chosen depending on the application background. These transforms include orthogonal wavelets, the discrete Fourier transform, the Gabor transforms, the discrete cosine transforms, the discrete local cosine transforms (see, e.g., [26,39]), redundant framelets (see, e.g., [27,42]), and curvelets [13]. The transform normally is chosen so that a few transform coefficients can be used to approximate

[*] Temasek Laboratories and Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543. Email:tslcaij@nus.edu.sg.

[†] Department of Mathematics, The Chinese University of Hong Kong, Shatin, N.T., Hong Kong, P. R. China. Email:rchan@math.cuhk.edu.hk. The research was supported in part by HKRGC Grant 400505 and CUHK DAG 2060257.

[‡] Department of Mathematics, Syracuse University, Syracuse, NY 13244. Email: lshen03@syr.edu. The research was supported by the US National Science Foundation under grant DMS-0712827.

[§] Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543. Email: matzuows@nus.edu.sg. The research was supported in part by Grant R-146-000-060-112 at the National University of Singapore.

or represent the intrinsic features of the underlying image. In many applications, this "sparse approximation" property is the key for designing efficient algorithms in the transformed domain for solving various problems.

Let $\mathbf{g} \in \mathbb{R}^N$ be the observed image. Let Λ be a subset of the index set $\mathcal{N} := \{1, \ldots, N\}$ which indicates where data are available in the image domain, i.e., $j \notin \Lambda$ implies $\mathbf{g}_j$ is unknown. Define the projection $\mathbf{P}_\Lambda$ to be the diagonal matrix:

$$\mathbf{P}_\Lambda[i,j] = \begin{cases} 1 & \text{if } i = j \in \Lambda, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Therefore, the known data in the image domain can be expressed by $\mathbf{P}_\Lambda \mathbf{f} = \mathbf{P}_\Lambda \mathbf{g}$.

Let $\mathbf{x} \in \mathbb{R}^M$ be the transform of the true image $\mathbf{f}$ under the transform $\mathbf{A}$. Here in the transformed domain, we similarly define $\Gamma \subset \mathcal{M} := \{1, \ldots, M\}$ to be the set on which data in $\mathbf{x}$ are available, i.e., $\mathbf{x}_j$ for $j \notin \Gamma$ are missing. Define the projection matrix $\mathbf{P}_\Gamma$ as in (1):

$$\mathbf{P}_\Gamma[i,j] = \begin{cases} 1 & \text{if } i = j \in \Gamma, \\ 0 & \text{otherwise.} \end{cases}$$

Then the known data in the transformed domain can be expressed by $\mathbf{P}_\Gamma \mathbf{A} \mathbf{f} = \mathbf{P}_\Gamma \mathbf{x}$.

If we have incomplete data in both the image domain and the transformed domain, then inpainting simultaneously in both domains is to find an $\mathbf{f}$ that satisfies

$$\begin{cases} \mathbf{P}_\Lambda \mathbf{f} = \mathbf{P}_\Lambda \mathbf{g}, \\ \mathbf{P}_\Gamma \mathbf{A} \mathbf{f} = \mathbf{P}_\Gamma \mathbf{x}. \end{cases} \tag{2}$$

The inpainting problem can have trivial solutions in some cases. For example, when $\Lambda = \mathcal{N}$ and $\Gamma = \emptyset$, then $\mathbf{f} = \mathbf{g}$ if $\mathbf{g}$ contains no noise, or it reduces to a denoising problem otherwise. Thus in the following, we assume that $\Lambda \subsetneqq \mathcal{N}$ and $\Gamma \subsetneqq \mathcal{M}$. The inpainting problem can also have infinitely many solutions in some cases. For example, when $\Lambda \subsetneqq \mathcal{N}$ and $\Gamma = \emptyset$, one can choose any values to fill in the region $\mathcal{N} \backslash \Lambda$. In these cases, we need to impose some regularization conditions on the solution, so that the chosen solution has certain smoothness requirements among all possible solutions. Yet in some other cases, the inpainting problem can have no solution at all. For example, when the data set $\mathbf{P}_\Gamma \mathbf{x}$ falls out of the range of $\mathbf{P}_\Gamma \mathbf{A}$. This is possible, since the range of $\mathbf{A}$ is the orthogonal compliment of the kernel of $\mathbf{A}^T$ which is not empty when $\mathbf{A}$ is a redundant system. Even when $\mathbf{P}_\Gamma \mathbf{x}$ does fall inside of the range of $\mathbf{P}_\Gamma \mathbf{A}$, the data given on Λ may not be compatible with the data given on Γ; and this results in (2) having no solution again. In these cases, we choose our solution $\mathbf{f}^*$ so that $\mathbf{P}_\Gamma \mathbf{A} \mathbf{f}^*$ is close to $\mathbf{P}_\Gamma \mathbf{x}$ in some sense.

In this paper, we are going to develop an algorithm for solving (2). To motivate the algorithm, we discuss the special cases where we do the inpainting either in the image domain or in the transformed domain, but not both. First consider $\Lambda \subsetneqq \mathcal{N}$ and $\Gamma = \emptyset$, i.e., we are given some data only in the image domain. It arises for example in restoring ancient drawings, where a portion of the picture is missing or damaged due to aging or scratches; or when an image is transmitted through a noisy channel. The problem of restoration from incomplete data in the image domain is referred to as *image inpainting*. Many useful techniques have been proposed in recent years to address the problem, see, for examples, [1, 2, 22, 23, 34, 36].

Recently, a frame-based method for solving image inpainting problem is proposed in [8, 20]. It is given by the following iteration:

$$\mathbf{f}^{(n+1)} = (\mathbf{I} - \mathbf{P}_\Lambda) \mathbf{T}_{\mathbf{u}} \mathbf{A} \mathbf{f}^{(n)} + \mathbf{P}_\Lambda \mathbf{g}, \quad n = 0, 1, \ldots \tag{3}$$

where $\mathbf{T_u}$ is the soft thresholding operator

$$\mathbf{T_u}(\mathbf{y}) := (t_{u_1}(y_1), \cdots, t_{u_i}(y_i), \cdots, t_{u_M}(y_M))$$

defined in [30] with

$$t_{u_i}(y_i) := \arg\min_x \{\frac{1}{2}|y_i - x|^2 + u_i|x|\} = \begin{cases} 0 & \text{if } |y_i| \le u_i, \\ y_i - \operatorname{sgn}(y_i)u_i & \text{if } |y_i| \ge u_i. \end{cases} \quad (4)$$

The algorithm is efficient and gives 2 to 3 dB improvement in PSNR over the variational approaches given in [22, 23]. As we will see, this algorithm will become a special case of the general algorithm we develop here. It was proven in [8] that, in the image domain, the iterant $\mathbf{f}^{(n)}$ of (3) converges to a minimizer of

$$\min_{\{\mathbf{f}:\mathbf{P}_\Lambda\mathbf{f}=\mathbf{P}_\Lambda\mathbf{g}\}} \left\{ \min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{Af} - \mathbf{y}\|_2^2 + \|\operatorname{diag}(\mathbf{u})\mathbf{y}\|_1 \right\} \right\},$$

whereas in the transformed domain, the coefficients $\mathbf{y}^{(n)} := \mathbf{T_u}\mathbf{Af}^{(n)}$ converges to a minimizer of

$$\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{P}_\Lambda(\mathbf{g} - \mathbf{A}^T\mathbf{y})\|_2^2 + \frac{1}{2}\|(\mathbf{I} - \mathbf{AA}^T)\mathbf{y}\|_2^2 + \|\operatorname{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}.$$

We have explained in [8] that the cost functionals given above balance the data fidelity, regularity and sparsity of the limits.

Next we consider the case when $\Lambda = \emptyset$ and $\Gamma \subsetneqq \mathcal{M}$. The problem (2) reduces to the *inpainting problem in the transformed domain*. Many problems in image processing can be viewed as such problem. For example, in high-resolution or super-resolution image reconstruction [6, 32, 40], the observed low-resolution images can be understood as images obtained by passing the high-resolution image through a low-pass filter. By constructing an appropriate tight framelet system [27], the image reconstruction problem is equivalent to restoring the high-resolution image from the given low-frequency coefficients in the framelet domain. Various algorithms for solving this inpainting problem in the framelet domain were developed and studied in [15–17, 19, 21]. We omit the details here. Another example of inpainting in the transformed domain is the restoration of chopped and nodded images. In infrared imaging in astronomy, incoming data pass through a chopped and nodded process such that the observed image is basically a second order difference of the original image, see [3, 4]. Hence the observed image can be viewed as the original image passing through a high-pass filter. By constructing an appropriate tight framelet system, the image reconstruction problem is equivalent to restoring the original image from part of its high-frequency coefficients in the framelet domain. An algorithm for solving such problem is given in [7]. The transformed-domain based algorithms developed in these papers [7, 15–17, 19, 21] are similar to that in (3) in spirit. In this paper we are going to combine them with (3) to obtain our algorithm for inpainting in both domains.

We should point out that there are many papers related to inpainting in the transformed domain. To name a few, in [24], the authors studied the problem of filling in missing or damaged wavelet coefficients due to communication or lossy image transmission. They solve the problem by a non-linear PDE. In the classical tomography problem in medical imaging, the 2D image is constructed from samples of its discrete Fourier transform on a star-shaped domain, see [29]. It was observed in [14] that a convex minimization gives an almost exact solution to the original image under certain assumptions.

Now we present our algorithm for solving the inpainting problem in both domains, i.e. (2). Motivated by the inpainting algorithm in the image domain (3), we propose the iteration

$$\mathbf{f}^{(n+1)} = (\mathbf{I} - \mathbf{P}_\Lambda)\mathbf{A}^T\mathbf{T_u}\big(\mathbf{P}_\Gamma\mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{Af}^{(n)}\big) + \mathbf{P}_\Lambda\mathbf{g}, \quad n = 0, 1, \ldots \quad (5)$$

It is clear that when $\Gamma = \emptyset$, (5) becomes (3). Algorithm (5) can be understood as follows. Given an approximation $\mathbf{f}^{(n)}$ of the underlying image $\mathbf{f}$, we first transform $\mathbf{f}^{(n)}$ into the transformed domain to get the transformed coefficients $\mathbf{A}\mathbf{f}^{(n)}$. Then the coefficients on Γ are replaced by the known data $\mathbf{P}_\Gamma\mathbf{x}$. After that, we apply the soft-thresholding operator $\mathbf{T}_\mathbf{u}$ on the coefficients $\mathbf{P}_\Gamma\mathbf{x}+(\mathbf{I}-\mathbf{P}_\Gamma)\mathbf{A}\mathbf{f}^{(n)}$ to perturb the transform coefficients and to remove possible noise. Finally, the modified coefficients $\mathbf{T}_\mathbf{u}\big(\mathbf{P}_\Gamma\mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{A}\mathbf{f}^{(n)}\big)$ are transformed back to the image domain, and the image pixels on Λ are replaced by the known data $\mathbf{P}_\Lambda\mathbf{g}$. This gives the next approximation $\mathbf{f}^{(n+1)}$.

In this paper, we will prove the convergence of $\mathbf{f}^{(n)}$ in (5) by using the proximal forward-backward splitting [25]. Clearly $\mathbf{f}^{(n)}$ converges if and only if $\mathbf{y}^{(n)} := \mathbf{T}_\mathbf{u}\big(\mathbf{P}_\Gamma\mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{A}\mathbf{f}^{(n)}\big)$ converges. Let $\mathbf{f}^*$ be the limit of $\mathbf{f}^{(n)}$ and $\mathbf{y}^*$ be the limit of $\mathbf{y}^{(n)}$. Then

$$\mathbf{f}^* = (\mathbf{I} - \mathbf{P}_\Lambda)\mathbf{A}^T\mathbf{y}^* + \mathbf{P}_\Lambda\mathbf{g}. \tag{6}$$

When the observed data $\mathbf{g}$ contains no noise, we use $\mathbf{f}^*$ defined in (6) as the solution to our inpainting problem (2). Then, we have $\mathbf{P}_\Lambda\mathbf{f}^* = \mathbf{P}_\Lambda\mathbf{g}$ in this case. If $\mathbf{g}$ contains noise, we choose $\mathbf{A}^T\mathbf{y}^*$, the denoised version of $\mathbf{f}^*$, to be the solution to (2).

We will show that in the transformed domain, the limit $\mathbf{y}^*$ is a minimizer of:

$$\min_{\{\mathbf{y}:\mathbf{P}_\Gamma\mathbf{y}=\mathbf{P}_\Gamma\mathbf{T}_\mathbf{u}\mathbf{x}\}} \left\{\frac{1}{2}\|\mathbf{P}_\Lambda(\mathbf{g} - \mathbf{A}^T\mathbf{y})\|_2^2 + \frac{1}{2}\|(\mathbf{I} - \mathbf{A}\mathbf{A}^T)\mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1\right\}. \tag{7}$$

The roles of each term in (7) can be explained as follows. The first term penalizes the distance of $\mathbf{A}^T\mathbf{y}^*$ to the given data $\mathbf{P}_\Lambda\mathbf{g}$ if $\mathbf{A}^T\mathbf{y}^*$ is chosen to be the solution. If $\mathbf{f}^*$ is chosen to be the solution, then this term penalizes the artifacts of the solution as discussed in [8]. The third term is to ensure the sparsity of the transformed coefficients, which in turn ensures the edges are sharp. The second term penalizes the distance between the coefficients $\mathbf{y}$ and the range of $\mathbf{A}$, i.e. the distance between $\mathbf{y}$ and the canonical coefficients of the tight frame transform. Hence the second term is related to the smoothness of $\mathbf{f}^*$, since canonical coefficients of a transform is often linked to the smoothness of the underlying function. For example, some weighted norm of the canonical framelet coefficients is equivalent to some Besov norm of the underlying function, see [5]. Therefore, the second term together with the third term guarantee the regularity of $\mathbf{f}$.

In the image domain, we will show that the limit $\mathbf{f}^*$ is a minimizer of

$$\min_{\{\mathbf{f}:\mathbf{P}_\Lambda\mathbf{f}=\mathbf{P}_\Lambda\mathbf{g}\}} \left\{\min_{\{\mathbf{y}:\mathbf{P}_\Gamma\mathbf{y}=\mathbf{P}_\Gamma\mathbf{T}_\mathbf{u}\mathbf{x}\}} \left\{\frac{1}{2}\|\mathbf{A}\mathbf{f} - \mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1\right\}\right\}. \tag{8}$$

Since $\|\mathbf{y} - \mathbf{A}\mathbf{f}\|_2^2 \geq \|\mathbf{P}_\Gamma(\mathbf{y} - \mathbf{A}\mathbf{f})\|_2^2$, the first term penalizes the distance of $\mathbf{A}\mathbf{f}$ to the given data in the transformed domain. It also penalizes the distance of $\mathbf{y}^*$ to the range of $\mathbf{A}$. The second term measures again the sparsity of the solution in the transformed domain. Altogether, we see that the cost functionals (7) and (8) balance the data fidelity in both the image and the transformed domains, and the regularity and sparsity of the limit $\mathbf{y}^*$ in the transformed domain, which in turn guarantee the regularity of solution in the image domain.

To illustrate the applicability of our algorithm (5), we will apply it to a super-resolution image reconstruction problem where we reconstruct a scene from images of the same scene but with different zooms, see e.g. [38]. By constructing appropriate framelet systems, the most zoomed image can be seen as incomplete data in the image domain, while the less zoomed images are just incomplete data in the transformed domain.

The rest of the paper is organized as follows. In Section 2, we prove the convergence of Algorithm (5), and give the minimization properties which the limits satisfy. In Section 3, some applications of our algorithms are presented and numerical simulations are given. Finally, a short conclusion is given in Section 4.

2 Analysis

In this section, we prove the convergence of the proposed algorithm (5). The proximal forward-backward splitting (PFBS) in [25], which is based on the theory of convex analysis [37], is used as the main tool in the proof. To introduce it, we begin with the definitions of the proximal operator and the envelope function. Let φ be a proper, lower semi-continuous function, the proximal operator of φ is defined by

$$\mathrm{prox}_{\varphi}(\mathbf{m}) = \arg\min_{\mathbf{n}} \left\{ \frac{1}{2}\|\mathbf{m}-\mathbf{n}\|_2^2 + \varphi(\mathbf{n}) \right\}, \tag{9}$$

and the envelope of φ is defined by

$$\mathrm{env}_{\varphi}(\mathbf{m}) = \min_{\mathbf{n}} \left\{ \frac{1}{2}\|\mathbf{m}-\mathbf{n}\|_2^2 + \varphi(\mathbf{n}) \right\}. \tag{10}$$

Recall that a function φ is proper if $\varphi(x) < +\infty$ for at least one x and $\varphi(x) > -\infty$ for all x. It can be found in the reference [30]. The key relation between the proximal operator and the envelope of φ is

$$\nabla \mathrm{env}_{\varphi}(\mathbf{m}) = \mathbf{m} - \mathrm{prox}_{\varphi}(\mathbf{m}). \tag{11}$$

Consider the minimization problem

$$\min_{\mathbf{m}}\{F_1(\mathbf{m}) + F_2(\mathbf{m})\} \tag{12}$$

where

(a) $F_1(\mathbf{m})$ is a proper, convex, lower semi-continuous function; and

(b) $F_2(\mathbf{m})$ is a proper, convex function with a $1/b$-Lipschitz continuous gradient, i.e.,

$$\|\nabla F_2(\mathbf{m}) - \nabla F_2(\mathbf{n})\|_2 \le \frac{1}{b}\|\mathbf{m}-\mathbf{n}\|_2, \quad \forall \mathbf{m}, \mathbf{n}.$$

The PFBS iteration in [25] for minimizing (12) is given by

$$\mathbf{m}^{(n+1)} = \mathrm{prox}_{rF_1}(\mathbf{m}^{(n)} - r\nabla F_2(\mathbf{m}^{(n)})), \quad n = 0, 1, \ldots \tag{13}$$

where r is a non-negative real number. We have the following convergence results for (13) which is an immediate consequence of Theorem 3.4 in [25].

Proposition 1 *Assume that a minimizer of* (12) *exists. If* $F_1(\mathbf{m})$ *and* $F_2(\mathbf{m})$ *satisfy the conditions in (a) and (b) respectively, and* $0 < r < 2b$*, then for any initial guess* $\mathbf{m}^{(0)}$*, the iteration* (13) *converges to a minimizer of* (12).

Proposition 1 asserts that the minimizer of the problem (12) is the fixed point of the operator $\mathrm{prox}_{rF_1} \circ (I - r\nabla F_2)$ and that the fixed point can be obtained via the iterative algorithm (13). Thus to prove the convergence of algorithm (5), the strategy is to identify (5) as algorithm (13) for appropriate functions F_1 and F_2. This is the topic in the next two subsections.

2.1 Convergence in the Image Domain

In this subsection, we prove the convergence of (5) in the image domain, i.e. $\lim_{n\to\infty} \mathbf{f}^{(n)}$ exists. Define

$$\mathbf{y}^{(n)} := \mathbf{T}_{\mathbf{u}}\big(\mathbf{P}_\Gamma \mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{A}\mathbf{f}^{(n)}\big). \tag{14}$$

Then (5) can be written as an alternate iteration process,

$$\begin{cases} \mathbf{y}^{(n)} = \mathbf{T}_{\mathbf{u}}\big(\mathbf{P}_\Gamma \mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{A}\mathbf{f}^{(n)}\big), \\ \mathbf{f}^{(n+1)} = (\mathbf{I} - \mathbf{P}_\Lambda)\mathbf{A}^T\mathbf{y}^{(n)} + \mathbf{P}_\Lambda \mathbf{g}. \end{cases}$$

Our main aim is to write both $\mathbf{y}^{(n)}$ and $\mathbf{f}^{(n+1)}$ as proximal operators defined in (9). We start with $\mathbf{y}^{(n)}$ first. Note that $\mathbf{T}_{\mathbf{u}}$ and $\mathbf{P}_\Gamma$ are commutable, and so are $\mathbf{T}_{\mathbf{u}}$ and $\mathbf{I} - \mathbf{P}_\Gamma$. Therefore, $\mathbf{y}^{(n)}$ can be rewritten as:

$$\mathbf{y}^{(n)} = \mathbf{P}_\Gamma \mathbf{T}_{\mathbf{u}} \mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{T}_{\mathbf{u}}(\mathbf{A}\mathbf{f}^{(n)}). \tag{15}$$

By (4),

$$\mathbf{T}_{\mathbf{u}}(\mathbf{A}\mathbf{f}^{(n)}) = \arg\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{A}\mathbf{f}^{(n)} - \mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}. \tag{16}$$

Define the set $\mathcal{T}$ as

$$\mathcal{T} := \left\{ \mathbf{y} \in \mathbb{R}^M : \mathbf{P}_\Gamma \mathbf{y} = \mathbf{P}_\Gamma \mathbf{T}_{\mathbf{u}} \mathbf{x} \right\}. \tag{17}$$

It is nonempty, closed, and convex. Combining (15) and (16), $\mathbf{y}^{(n)}$ can be viewed as the minimizer of the constrained minimization problem

$$\mathbf{y}^{(n)} = \arg\min_{\mathbf{y}\in\mathcal{T}} \left\{ \frac{1}{2}\|\mathbf{A}\mathbf{f}^{(n)} - \mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}. \tag{18}$$

The constrained minimization problem (18) can be converted into an unconstrained one as follows:

$$\mathbf{y}^{(n)} = \arg\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{A}\mathbf{f}^{(n)} - \mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 + \iota_{\mathcal{T}}(\mathbf{y}) \right\},$$

where $\iota_{\mathcal{T}}$ is the indicator function for the convex set $\mathcal{T}$, i.e.,

$$\iota_{\mathcal{T}}(\mathbf{y}) = \begin{cases} 0, & \text{if } \mathbf{y} \in \mathcal{T}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{19}$$

If we define

$$\xi(\mathbf{y}) := \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 + \iota_{\mathcal{T}}(\mathbf{y}), \tag{20}$$

and using the notations (9) and (19), we get an equivalent formulation for (18):

$$\mathbf{y}^{(n)} \;=\; \mathrm{prox}_\xi(\mathbf{A}\mathbf{f}^{(n)}). \tag{21}$$

Next we try to write $\mathbf{f}^{(n+1)}$ as a proximal operator. Note that by (5) and (14):

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

Then

$$\mathbf{f}^{(n+1)} = \arg\min_{\mathbf{f}\in\mathcal{I}} \frac{1}{2}\|\mathbf{A}^T\mathbf{y}^{(n)} - \mathbf{f}\|_2^2,$$

where $\mathcal{I}$ is a nonempty closed convex set defined by

$$\mathcal{I} := \left\{\mathbf{f} \in \mathbb{R}^N : \mathbf{P}_\Lambda \mathbf{f} = \mathbf{P}_\Lambda \mathbf{g}\right\}. \tag{22}$$

If we define the indicator function $\iota_{\mathcal{I}}$ for $\mathcal{I}$ as in (19), we get

$$\mathbf{f}^{(n+1)} = \arg\min_{\mathbf{f}} \left\{\frac{1}{2}\|\mathbf{A}^T\mathbf{y}^{(n)} - \mathbf{f}\|_2^2 + \iota_{\mathcal{I}}(\mathbf{f})\right\},$$

which by (9) is equivalent to

$$\mathbf{f}^{(n+1)} = \mathrm{prox}_{\iota_{\mathcal{I}}}(\mathbf{A}^T\mathbf{y}^{(n)}). \tag{23}$$

Combining (21) and (23) leads to a new formulation of (5) with the help of the proximal operators with respect to $\iota_{\mathcal{I}}$ and ξ:

$$\mathbf{f}^{(n+1)} = \mathrm{prox}_{\iota_{\mathcal{I}}}\big(\mathbf{A}^T\mathrm{prox}_\xi(\mathbf{A}\mathbf{f}^{(n)})\big). \tag{24}$$

Because of (11) and the chain rule, we get

$$\nabla \mathrm{env}_\xi(\mathbf{A}\mathbf{f}^{(n)}) = \mathbf{A}^T(\mathbf{A}\mathbf{f}^{(n)} - \mathrm{prox}_\xi(\mathbf{A}\mathbf{f}^{(n)})). \tag{25}$$

Noting the fact that $\mathbf{A}^T\mathbf{A} = \mathbf{I}$, we can rewrite (24), which is equivalent to our algorithm (5), as

$$\begin{aligned}
\mathbf{f}^{(n+1)} &= \mathrm{prox}_{\iota_{\mathcal{I}}}\big(\mathbf{A}^T\mathrm{prox}_\xi(\mathbf{A}\mathbf{f}^{(n)})\big) \\
&= \mathrm{prox}_{\iota_{\mathcal{I}}}\big(\mathbf{f}^{(n)} - \mathbf{A}^T\mathbf{A}\mathbf{f}^{(n)} + \mathbf{A}^T\mathrm{prox}_\xi(\mathbf{A}\mathbf{f}^{(n)})\big) \\
&= \mathrm{prox}_{\iota_{\mathcal{I}}}\Big(\mathbf{f}^{(n)} - \mathbf{A}^T\big(\mathbf{A}\mathbf{f}^{(n)} - \mathrm{prox}_\xi(\mathbf{A}\mathbf{f}^{(n)})\big)\Big) \\
&= \mathrm{prox}_{\iota_{\mathcal{I}}}\big(\mathbf{f}^{(n)} - \nabla\mathrm{env}_\xi(\mathbf{A}\mathbf{f}^{(n)})\big).
\end{aligned}$$

This is identical to the PFBS iteration (13) with

$$F_1(\mathbf{f}) = \iota_{\mathcal{I}}(\mathbf{f}), \quad F_2(\mathbf{f}) = \mathrm{env}_\xi(\mathbf{A}\mathbf{f}), \quad \text{and} \quad r = 1.$$

Thus our algorithm (5) is the PFBS iteration for

$$\min_{\mathbf{f}}\{F_1(\mathbf{f}) + F_2(\mathbf{f})\} = \min_{\mathbf{f}\in\mathcal{I}} \mathrm{env}_\xi(\mathbf{A}\mathbf{f}),$$

which is equivalent to

$$\min_{\mathbf{f}\in\mathcal{I}}\left\{\min_{\mathbf{y}\in\mathcal{T}}\left\{\frac{1}{2}\|\mathbf{A}\mathbf{f} - \mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1\right\}\right\}. \tag{26}$$

By applying Proposition 1, we get the following convergence theorem for (5).

Theorem 1 *Assume that the thresholding parameter vector $\mathbf{u} > \mathbf{0}$ entrywise. Then for any initial guess $\mathbf{f}^{(0)}$, the iteration* (5) *converges to a minimizer of* (26)*.*

Proof: Let $F_1(\mathbf{f}) = \iota_{\mathcal{I}}(\mathbf{f})$ and $F_2(\mathbf{f}) = \mathrm{env}_\xi(\mathbf{A}\mathbf{f})$. By Proposition 1, we need to verify the conditions in (a) and (b), check $0 < r = 1 < 2b$, and prove the existence of minimizers of (26).

It is well known that the indicator function $F_1(\mathbf{f}) = \iota_{\mathcal{I}}(\mathbf{f})$ is proper, convex, and lower semi-continuous. By Lemma 2.5 in [25], $F_2(\mathbf{f}) = \mathrm{env}_\xi(\mathbf{A}\mathbf{f})$ is convex and differentiable, and its gradient is given in (25). It remains to prove that its gradient is $1/b$-Lipschitz continuous. To this end, we note that by Lemma 2.4 in [25],

$$\|(\mathbf{g} - \mathrm{prox}_\varphi(\mathbf{g})) - (\mathbf{h} - \mathrm{prox}_\varphi(\mathbf{h}))\|_2 \le \|\mathbf{g} - \mathbf{h}\|_2 \tag{27}$$

for any convex and lower semi-continuous function φ. Using (27) and (25), we have

$$\begin{aligned} \|\nabla(\mathrm{env}_\xi(\mathbf{Ag})) - \nabla(\mathrm{env}_\xi(\mathbf{Ah}))\|_2 &= \|\mathbf{A}^T(\mathbf{Ag} - \mathrm{prox}_\xi(\mathbf{Ag})) - \mathbf{A}^T(\mathbf{Ah} - \mathrm{prox}_\xi(\mathbf{Ah}))\|_2 \\ &\leq \|\mathbf{A}^T\|_2\|(\mathbf{Ag} - \mathrm{prox}_\xi(\mathbf{Ag})) - (\mathbf{Ah} - \mathrm{prox}_\xi(\mathbf{Ah}))\|_2 \\ &\leq \|\mathbf{A}^T\|_2\|\mathbf{A}(\mathbf{g} - \mathbf{h})\|_2 \leq \|\mathbf{g} - \mathbf{h}\|_2. \end{aligned}$$

This says that $F_2(\mathbf{f}) = \mathrm{env}_\xi(\mathbf{Af})$ has a 1-Lipschitz continuous gradient, i.e. the Lipshitz constant is $1/b = 1$. Hence $0 < 1 < 2b$.

It remains to show that there exists at least one minimizer of (26). We note that (26) is equivalent to $\min\{F_1(\mathbf{f})+F_2(\mathbf{f})\}$. Since F_1 and F_2 satisfies the conditions in (a) and (b) respectively, by Proposition 3.1 in [25], it suffices to show that $F_1 + F_2$ is coercive, i.e., as $\|\mathbf{f}\|_2 \to +\infty$, $F_1(\mathbf{f}) + F_2(\mathbf{f}) \to +\infty$. Note that by the definition

$$F_2(\mathbf{f}) = \mathrm{env}_\xi(\mathbf{Af}) = \min_{\mathbf{y}\in\mathcal{T}} \left\{ \frac{1}{2}\|\mathbf{Af} - \mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}. \tag{28}$$

Let

$$\mathbf{y}^* = \arg\min_{\mathbf{y}\in\mathcal{T}} \left\{ \frac{1}{2}\|\mathbf{Af} - \mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}. \tag{29}$$

Comparing (28) and (29) with (15) and (18), we see that $\mathbf{y}^* = \mathbf{T_u}(\mathbf{P}_\Gamma\mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{Af})$. Let $U = \max_i u_i$ and $u = \min_i u_i$. Then,

$$\begin{aligned} F_2(\mathbf{f}) &= \frac{1}{2}\|\mathbf{Af} - \mathbf{T_u}(\mathbf{P}_\Gamma\mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{Af})\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{T_u}(\mathbf{P}_\Gamma\mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{Af})\|_1 \\ &\geq \frac{1}{2}\|\mathbf{P}_\Gamma(\mathbf{Af} - \mathbf{T_u}\mathbf{x})\|_2^2 + u\|(\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{T_u}(\mathbf{Af})\|_1 \\ &\geq \frac{1}{2}\|\mathbf{P}_\Gamma(\mathbf{Af} - \mathbf{T_u}\mathbf{x})\|_2^2 + u\|(\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{Af}\|_1 - uUM \\ &\geq \frac{1}{2}\|\mathbf{P}_\Gamma(\mathbf{Af} - \mathbf{T_u}\mathbf{x})\|_2^2 + u\|(\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{Af}\|_2 - uUM. \end{aligned} \tag{30}$$

Since $\|\mathbf{f}\|_2 = \|\mathbf{Af}\|_2$, we have $\|\mathbf{P}_\Gamma\mathbf{Af}\|_2^2 + \|(\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{Af}\|_2^2 = \|\mathbf{f}\|_2^2$. Therefore, either $\|\mathbf{P}_\Gamma\mathbf{Af}\|_2 \geq \frac{\sqrt{2}}{2}\|\mathbf{f}\|_2$ or $\|(\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{Af}\|_2 \geq \frac{\sqrt{2}}{2}\|\mathbf{f}\|_2$. Hence, we have either

$$T_1 := \frac{1}{2}\|\mathbf{P}_\Gamma(\mathbf{Af} - \mathbf{T_u}\mathbf{x})\|_2^2 \geq \frac{1}{2}\left(\frac{\sqrt{2}}{2}\|\mathbf{f}\|_2 - \|\mathbf{P}_\Gamma\mathbf{T_u}\mathbf{x}\|_2\right)^2$$

or

$$T_2 := \|(\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{Af}\|_2 \geq \frac{\sqrt{2}}{2}\|\mathbf{f}\|_2.$$

Putting these two inequalities back into (30), we see that as $\|\mathbf{f}\|_2 \to +\infty$,

$$\begin{aligned} & F_1(\mathbf{f}) + F_2(\mathbf{f}) \geq F_2(\mathbf{f}) \geq T_1 + uT_2 - uUM \\ \geq\ & \max\big\{\frac{1}{2}(\frac{\sqrt{2}}{2}\|\mathbf{f}\|_2 - \|\mathbf{P}_\Gamma\mathbf{T_u}\mathbf{x}\|_2)^2, u\frac{\sqrt{2}}{2}\|\mathbf{f}\|_2\big\} - uUM \to +\infty. \end{aligned}$$

□

2.2 Convergence in the Transformed Domain

In this subsection, we prove the convergence of algorithm (5) in the transformed domain and also study the properties of the limit. Clearly from the definition of $\mathbf{y}^{(n)}$ in (14), $\lim_{n\to\infty}\mathbf{y}^{(n)} = \mathbf{y}^*$ exists since the $\lim_{n\to\infty}\mathbf{f}^{(n)} = \mathbf{f}^*$ exists. In fact, we have $\mathbf{y}^* = \mathbf{T}_{\mathbf{u}}\big(\mathbf{P}_\Gamma\mathbf{x} + (\mathbf{I} - \mathbf{P}_\Gamma)\mathbf{A}\mathbf{f}^*\big)$ since $\mathbf{T}_{\mathbf{u}}$, $\mathbf{P}_\Gamma$, and $\mathbf{A}$ are all continuous operators. Next we find out what functional $\mathbf{y}^*$ minimizes.

Note that by (11) and the chain rule, we have

$$\nabla\mathrm{env}_{\iota_{\mathcal{I}}}(\mathbf{A}^T\mathbf{y}^{(n)}) = \mathbf{A}(\mathbf{A}^T\mathbf{y}^{(n)} - \mathrm{prox}_{\iota_{\mathcal{I}}}(\mathbf{A}^T\mathbf{y}^{(n)})). \tag{31}$$

Using (21), (23) and (31), we obtain

$$\begin{aligned}\mathbf{y}^{(n+1)} &= \mathrm{prox}_\xi\big(\mathbf{A}\mathrm{prox}_{\iota_{\mathcal{I}}}(\mathbf{A}^T\mathbf{y}^{(n)})\big)\\ &= \mathrm{prox}_\xi\left\{\mathbf{y}^{(n)} - \Big[(\mathbf{y}^{(n)} - \mathbf{A}\mathbf{A}^T\mathbf{y}^{(n)}) + \mathbf{A}\big(\mathbf{A}^T\mathbf{y}^{(n)} - \mathrm{prox}_{\iota_{\mathcal{I}}}(\mathbf{A}^T\mathbf{y}^{(n)})\big)\Big]\right\}\\ &= \mathrm{prox}_\xi\Big\{\mathbf{y}^{(n)} - \nabla\big[\frac{1}{2}\|(\mathbf{I} - \mathbf{A}\mathbf{A}^T)\mathbf{y}^{(n)}\|_2^2 + \mathrm{env}_{\iota_{\mathcal{I}}}(\mathbf{A}^T\mathbf{y}^{(n)})\big]\Big\}.\end{aligned} \tag{32}$$

Thus by letting

$$F_1(\mathbf{y}) = \xi(\mathbf{y}), \quad F_2(\mathbf{y}) = \frac{1}{2}\|(\mathbf{I} - \mathbf{A}\mathbf{A}^T)\mathbf{y}\|_2^2 + \mathrm{env}_{\iota_{\mathcal{I}}}(\mathbf{A}^T\mathbf{y}), \quad \text{and} \quad r = 1, \tag{33}$$

we see that (32) is equivalent to the PFBS iteration (13) for the minimization of

$$\min_{\mathbf{y}}\{F_1(\mathbf{y}) + F_2(\mathbf{y})\} = \min_{\mathbf{y}}\left\{\frac{1}{2}\|(\mathbf{I} - \mathbf{A}\mathbf{A}^T)\mathbf{y}\|_2^2 + \mathrm{env}_{\iota_{\mathcal{I}}}(\mathbf{A}^T\mathbf{y}) + \xi(\mathbf{y})\right\}. \tag{34}$$

Note that by (10) and (22),

$$\mathrm{env}_{\iota_{\mathcal{I}}}(\mathbf{A}^T\mathbf{y}) = \min_{\mathbf{x}\in\mathcal{I}}\frac{1}{2}\|\mathbf{A}^T\mathbf{y} - \mathbf{x}\|_2^2 = \frac{1}{2}\|\mathbf{P}_\Lambda(\mathbf{A}^T\mathbf{y} - \mathbf{g})\|_2^2.$$

Using this and (20), (34) can be rewritten as

$$\min_{\mathbf{y}\in\mathcal{T}}\left\{\frac{1}{2}\|\mathbf{P}_\Lambda(\mathbf{g} - \mathbf{A}^T\mathbf{y})\|_2^2 + \frac{1}{2}\|(\mathbf{I} - \mathbf{A}\mathbf{A}^T)\mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1\right\}. \tag{35}$$

The roles of each term in the cost functional (35) are clear. The first term is to fit the data in the image domain, and the constraint is to fit the data in the transformed domain. The third term is to ensure the sparsity of the transformed coefficients. As mentioned in the introduction, the second term penalizes the distance between the coefficients $\mathbf{y}$ and the range of $\mathbf{A}$, i.e. the distance between $\mathbf{y}$ and the canonical coefficients of the tight frame transform. Since canonical coefficients of a transform is often linked to the smoothness of the underlying function, e.g. a weighted norm of the canonical framelet coefficients is equivalent to some Besov norm of the underlying function, see [5], the second term together with the third term guarantee the regularity of $\mathbf{f}$. Altogether, we see that the cost functional (35) balances the data fidelity in both the image and the transformed domains, and the regularity and sparsity of the limit $\mathbf{y}^*$ in the transformed domain, which in turn guarantee the regularity of solution in the image domain.

We now prove that the limit $\mathbf{y}^*$ minimizes (35).

Theorem 2 *Assume that the thresholding parameter vector* $\mathbf{u} > \mathbf{0}$ *entrywise. Then the sequence* $\mathbf{y}^{(n)}$ *defined by* (14) *converges to a minimizer of* (35).

Proof: Since $\mathbf{y}^{(n)}$ is the PFBS iteration for (35) with $F_1(\mathbf{y})$, $F_2(\mathbf{y})$, and r given in (33), the limit $\mathbf{y}^*$ will be a minimizer of (35) if we can verify the conditions in Proposition 1. Clearly, $F_1 = \xi(\mathbf{y})$ is proper, semi-continuous and convex. It is obvious that $F_2 = \frac{1}{2}\|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{y}\|_2^2 + \mathrm{env}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{y})$ is proper, continuous, and convex. Moreover, F_2 is differentiable since both the 2-norm and the envelope function are differentiable. Therefore, the only thing remaining to prove is that the gradient of F_2 is 1-Lipschitz continuous.

Note that

$$\begin{aligned}
& \|\nabla(\frac{1}{2}\|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{z}\|_2^2 + \mathrm{env}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{z})) - \nabla(\frac{1}{2}\|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{w}\|_2^2 + \mathrm{env}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{w}))\|_2^2 \\
= \; & \|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)(\mathbf{z}-\mathbf{w}) + \mathbf{A}\big(\mathbf{A}^T(\mathbf{z}-\mathbf{w}) - (\mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{z}) - \mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{w}))\big)\|_2^2 \\
= \; & \|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)(\mathbf{z}-\mathbf{w})\|_2^2 + \|\mathbf{A}\big(\mathbf{A}^T(\mathbf{z}-\mathbf{w}) - (\mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{z}) - \mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{w}))\big)\|_2^2. \qquad (36)
\end{aligned}$$

The last inequality is due to the fact that $(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{s}$ is orthogonal to $\mathbf{A}\mathbf{t}$ for any $\mathbf{s}$ and $\mathbf{t}$. For the second term in the last equality in (36), we have the following estimation,

$$\begin{aligned}
& \|\mathbf{A}\big(\mathbf{A}^T(\mathbf{z}-\mathbf{w}) - (\mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{z}) - \mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{w}))\big)\|_2^2 \\
= \; & \|\mathbf{A}^T(\mathbf{z}-\mathbf{w}) - (\mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{z}) - \mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{w}))\|_2^2 \\
= \; & \|(\mathbf{A}^T\mathbf{z} - \mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{z})) - (\mathbf{A}^T\mathbf{w} - \mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{w}))\|_2^2 \\
\leq \; & \|\mathbf{A}^T\mathbf{z} - \mathbf{A}^T\mathbf{w}\|_2^2 = \|\mathbf{A}\mathbf{A}^T(\mathbf{z}-\mathbf{w})\|_2^2, \qquad (37)
\end{aligned}$$

where the inequality follows from (27). Again, due to the fact that $(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{s}$ is orthogonal to $\mathbf{A}\mathbf{t}$ for any $\mathbf{s}$ and $\mathbf{t}$, by (36) and (37) we have

$$\begin{aligned}
& \|\nabla(\frac{1}{2}\|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{z}\|_2^2 + \mathrm{env}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{z})) - \nabla(\frac{1}{2}\|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{w}\|_2^2 + \mathrm{env}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{w}))\|_2^2 \\
= \; & \|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)(\mathbf{z}-\mathbf{w})\|_2^2 + \|\mathbf{A}\big(\mathbf{A}^T(\mathbf{z}-\mathbf{w}) - (\mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{z}) - \mathrm{prox}_{\iota_\mathcal{I}}(\mathbf{A}^T\mathbf{w}))\big)\|_2^2 \\
\leq \; & \|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)(\mathbf{z}-\mathbf{w})\|_2^2 + \|\mathbf{A}\mathbf{A}^T(\mathbf{z}-\mathbf{w})\|_2^2 \\
= \; & \|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)(\mathbf{z}-\mathbf{w}) + \mathbf{A}\mathbf{A}^T(\mathbf{z}-\mathbf{w})\|_2^2 = \|\mathbf{z}-\mathbf{w}\|_2^2.
\end{aligned}$$

Thus the gradient of F_2 is indeed 1-Lipschitz continuous. Therefore, by Proposition 1, the limit $\mathbf{y}^*$ is a minimizer of (35). □

We note that (35) is the same as (26) except that we swap the minimization order of $\mathbf{y}$ and $\mathbf{f}$, i.e., (35) coincides with

$$\min_{\mathbf{y}\in\mathcal{T}}\left\{\min_{\mathbf{f}\in\mathcal{I}}\left\{\frac{1}{2}\|\mathbf{A}\mathbf{f}-\mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1\right\}\right\}. \qquad (38)$$

To prove that, notice that $\mathbf{A}^T\mathbf{A} = \mathbf{I}$, so we can decompose $\mathbf{y}$ orthogonally into $\mathbf{y} = \mathbf{A}\mathbf{A}^T\mathbf{y} + (\mathbf{I} - \mathbf{A}\mathbf{A}^T)\mathbf{y}$. Hence (38) can be rewritten into

$$\min_{\mathbf{y}\in\mathcal{T}}\left\{\min_{\mathbf{f}\in\mathcal{I}}\left\{\frac{1}{2}\|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{y} + \mathbf{A}(\mathbf{A}^T\mathbf{y}-\mathbf{f})\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1\right\}\right\}.$$

Since $\mathbf{A}(\mathbf{A}^T\mathbf{y}-\mathbf{f})$ and $(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{y}$ are orthogonal to each other, and $\|\mathbf{A}(\mathbf{A}^T\mathbf{y}-\mathbf{f})\|_2 = \|\mathbf{A}^T\mathbf{y}-\mathbf{f}\|_2$, we obtain

$$\min_{\mathbf{y}\in\mathcal{T}}\left\{\frac{1}{2}\|(\mathbf{I}-\mathbf{A}\mathbf{A}^T)\mathbf{y}\|^2 + \min_{\mathbf{f}\in\mathcal{I}}\left\{\frac{1}{2}\|\mathbf{A}^T\mathbf{y}-\mathbf{f}\|_2^2\right\} + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1\right\}.$$

By the definition of the functions $\mathrm{env}_{\iota_\mathcal{I}}(\cdot)$ and $\xi(\cdot)$, this functional is the same as (34), which we have already proved to be equivalent to (35).

We close this section by summarizing what we have proved. The inpainting problem (2) in both domains is solved by finding a vector $\mathbf{f} \in \mathcal{I}$ and a vector $\mathbf{y} \in \mathcal{T}$ such that $\mathbf{Af}$ is close to $\mathbf{y}$ and $\mathbf{y}$ is sparse. We tackle this problem by minimizing the following functional

$$\min_{(\mathbf{f},\mathbf{y})\in(\mathcal{I}\otimes\mathcal{T})} \left\{ \frac{1}{2}\|\mathbf{Af}-\mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 \right\},$$

where the first term reflects the closeness between $\mathbf{Af}$ and $\mathbf{y}$, while the second term reflects the sparsity of $\mathbf{y}$ with respect to the weight $\mathbf{u}$. The minimization problem is solved by either (26) or (38). Moreover, its minimizer can be obtained by our iterative algorithm (5).

2.3 The term $\frac{1}{2}\|(\mathbf{I}-\mathbf{AA}^T)\mathbf{y}\|_2^2$ in (35)

In this subsection, we discuss the role of the term $\frac{1}{2}\|(\mathbf{I}-\mathbf{AA}^T)\mathbf{y}\|_2^2$ in (35). This discussion will lead to the conclusion that this term balances the synthesis and analysis based approaches in the literature, and is preferable as observed in [8,9]. To simplify the discussion, we focus on the problem of image inpainting. In this case, (35) becomes

$$\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{P}_\Lambda(\mathbf{g}-\mathbf{A}^T\mathbf{y})\|_2^2 + \frac{1}{2}\|(\mathbf{I}-\mathbf{AA}^T)\mathbf{y}\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}. \tag{39}$$

An algorithm for this model of inpainting is developed in [8]. In fact, this formulation and the corresponding algorithm are analyzed extensively in [8].

As mentioned before, the term $\frac{1}{2}\|(\mathbf{I}-\mathbf{AA}^T)\mathbf{y}\|_2^2$ controls the distance of the coefficient $\mathbf{y}$ to the range of the transform $\mathbf{A}$. Another possible choice of the cost functional is to omit the term $\frac{1}{2}\|(\mathbf{I}-\mathbf{AA}^T)\mathbf{y}\|_2^2$ that leads to the model

$$\min_{\mathbf{y}} \left\{ \frac{1}{2}\|\mathbf{P}_\Lambda(\mathbf{g}-\mathbf{A}^T\mathbf{y})\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}. \tag{40}$$

In this formulation, the sparsest coefficients in the transform is first sought, and then the recovered image is synthesized by the sparsest coefficient via the inverse transform. This is called the *synthesis* based approach, and was proposed in, e.g., [28, 34, 35]. An algorithm for this minimization was proposed in [35]. This approach fully explores the sparsity the redundant system that may bring in. On the other hand, since the smoothness of the underlying solutions normally connects to the canonical coefficients of the transform, one may prefer the distance to the range to be zero whenever the smoothness of the solution is important. This leads to another model

$$\min_{\mathbf{y}\in\mathrm{Range}(\mathbf{A})} \left\{ \frac{1}{2}\|\mathbf{P}_\Lambda(\mathbf{g}-\mathbf{A}^T\mathbf{y})\|_2^2 + \|\mathrm{diag}(\mathbf{u})\mathbf{y}\|_1 \right\}. \tag{41}$$

In this formulation, one seeks the recovered image such that it has the sparsest analyzed coefficients via the direct transform. This is called the *analysis* based approach, and was proposed in, e.g., [34]. An algorithm for this model and its analysis is developed in [9]. The synthesis and analysis based approaches are different as pointed out in [33]. Which one to choose depends on applications, in particular, which priority to seek, the sparsity or smoothness of underlying solutions. When the priority is not so evident, it is better to use the balanced model (39). We see from Figure 2 that the analysis based approach (41) and our proposed algorithm (5) (i.e. model (39)) are preferable in both simulations there. Finally, we remark that for the analysis based approaches, it is hard to find algorithms based on soft-thresholding by the PFBS iteration, as the proximity operator of

$\|\text{diag}(\mathbf{u})\mathbf{Af}\|_1$ is not soft-thresholding again. One has to employ iterative algorithms to find the proximity operator, hence burden the computational cost of the algorithm. We forgo the detailed discuss and the interested read should consult [9] for details.

Furthermore, in [9], the three models (39), (40) and (41) are formally unified into one by variation the relative weight of the term $\frac{1}{2}\|(\mathbf{I}-\mathbf{AA}^T)\mathbf{y}\|_2^2$ in (39). This leads to the following model:

$$\min_{\mathbf{y}}\left\{\frac{1}{2}\|\mathbf{P}_\Lambda(\mathbf{g}-\mathbf{A}^T\mathbf{y})\|_2^2+\frac{\mu}{2}\|(\mathbf{I}-\mathbf{AA}^T)\mathbf{y}\|_2^2+\|\text{diag}(\mathbf{u})\mathbf{y}\|_1\right\}. \tag{42}$$

When $\mu = 0$, (42) is the synthesis based approach (40). As the relative weight μ of the term $\frac{1}{2}\|(\mathbf{I}-\mathbf{AA}^T)\mathbf{y}\|_2^2$ increases, the distance of the coefficient $\mathbf{y}$ to the range of $\mathbf{A}$ decreases. In the limit case of $\mu = +\infty$, the distance must be 0, hence (42) becomes the analysis based approach (41). Therefore, the term $\frac{1}{2}\|(\mathbf{I}-\mathbf{AA}^T)\mathbf{y}\|_2^2$ balances the synthesis and analysis based approaches. Notice that, when $\mu = 1$, (42) is (39). Therefore, the model (39) is an intermediate between the synthesis and analysis based approaches. Minimization algorithms for (42) was proposed in [9], and the model (42) was extended to simultaneously cartoon and texture image inpainting using two tight frames. The interested read should consult [9] for details.

The recent development of Bregman iteration, especially the linearized Bregman iteration can also lead to some useful algorithms for some special cases of inpainting, although it is a different set of models. We omit the discussion here and the interested reader should consult [10–12] for linearized Bregman iteration and its applications.

3 Applications

In this section, we consider an application which falls into our model (2), i.e. inpainting in both the image and transformed domains. It is a super-resolution image reconstruction problem where the images are obtained by taking pictures of the same scene or part of it using different zooms. However, to better understand super-resolution image reconstruction, we start with a simpler problem first.

3.1 Super-Resolution Image Reconstruction with Multiple Sensors

In this subsection, we introduce super-resolution image reconstruction with multiple sensors. For this case, $\Lambda = \emptyset$, and the problem is an inpainting problem in the transformed domain. For simplicity, we only consider 1D case. It can be extended to 2D straightforwardly through tensor product, and we refer the readers to [6, 16] for more details.

Let $\mathbf{f}$ be the desired high-resolution image. We are going to reconstruct $\mathbf{f}$ by taking its low-resolution images using K multiple sensors of the same resolution but with different subpixel displacements. Before we write out the formulation of the model in [6] for a general K, we first illustrate the model for the simplest case $K = 2$. In such situation, the high-resolution image $\mathbf{f}$ (in discrete form) is modeled as a average-sampling of an underlying image f (in a continuous form). More precisely, we assume that

$$\mathbf{f}[i]=\frac{1}{L}\int_{(i-\frac{1}{2})L}^{(i+\frac{1}{2})L} f(t)\,dt$$

where L is the length of the average-sampling. Two low-resolution images are modeled by

$$\mathbf{g}_0[i]=\frac{1}{2L}\int_{2(i-\frac{1}{2})L}^{2(i+\frac{1}{2})L} f(t)\,dt \quad\text{and}\quad \mathbf{g}_1[i]=\frac{1}{2L}\int_{(2(i-\frac{1}{2})+1)L}^{(2(i+\frac{1}{2})+1)L} f(t)\,dt,$$

respectively. The length of the average-sampling for low-resolution images is $2L$. If the function f is constant on the interval $[(i-\frac{1}{2})L,(i+\frac{1}{2})L]$ for all i. Then we have

$$\mathbf{g}_0[i]=\frac{1}{4}\mathbf{f}[2i-1]+\frac{1}{2}\mathbf{f}[2i]+\frac{1}{4}\mathbf{f}[2i+1] \quad \text{and} \quad \mathbf{g}_1[i]=\frac{1}{4}\mathbf{f}[2i]+\frac{1}{2}\mathbf{f}[2i+1]+\frac{1}{4}\mathbf{f}[2i+2].$$

We combine the low-resolution images $\mathbf{g}_0$ and $\mathbf{g}_1$ into a new image $\mathbf{g}$ in the following way

$$\mathbf{g}[2i]=\mathbf{g}_0[i] \quad \text{and} \quad \mathbf{g}[2i+1]=\mathbf{g}_1[i].$$

Above formulation for $K=2$ can be easily extended for a large value K. $\mathbf{f}$ then satisfies the following equation:

$$(\mathbf{I}\otimes\mathbf{P}_{\mathcal{K}})(\mathbf{h}_0^K\odot\mathbf{f})=(\mathbf{I}\otimes\mathbf{P}_{\mathcal{K}})\mathbf{g},$$

where $\mathbf{I}$ is the identity matrix of size $\frac{N}{K}\times\frac{N}{K}$, the symbol $\otimes$ denotes matrix tensor product, the symbol $\odot$ denotes the discrete convolution,

$$\mathbf{h}_0^K=\frac{1}{K}[\frac{1}{2},\underbrace{1,\cdots,1,\cdots,1}_{K-1},\frac{1}{2}], \tag{43}$$

$\mathcal{K}$ is the set of indices of available sensors in K multiple sensors, $\mathbf{P}_{\mathcal{K}}$ is the K-by-K diagonal matrix defined by

$$\mathbf{P}_{\mathcal{K}}[i,j]=\begin{cases}1, & \text{if } i=j\in\mathcal{K},\\ 0, & \text{otherwise,}\end{cases}$$

and $\mathbf{g}$, called the *observed image*, is obtained by interlacing the low-resolution images, see the middle columns in Figures 3 and 4.

From the low-pass filter $\mathbf{h}_0^K$, one can construct filters $\mathbf{h}_1^K,\cdots,\mathbf{h}_r^K$ to forming a tight framelet system by the unitary extension principle, see [19] for detail. We will use non-downsampled multilevel framelet decomposition with symmetric boundary condition. For any filter $\mathbf{h}=[\mathbf{h}(i)]_{i=-I}^{I}$, we define a matrix $\mathbf{S}^{(\ell)}(\mathbf{h})$ of size $\frac{N}{K^\ell}\times\frac{N}{K^\ell}$ as

$$\begin{aligned}\mathbf{S}^{(\ell)}(\mathbf{h}) \;=\; & \begin{bmatrix} \mathbf{h}[0] & \cdots & \mathbf{h}[-I] & & 0\\ \vdots & \ddots & \ddots & \ddots & \\ \mathbf{h}[I] & \ddots & \ddots & \ddots & \mathbf{h}[-I]\\ & \ddots & \ddots & \ddots & \vdots\\ 0 & & \mathbf{h}[I] & \cdots & \mathbf{h}[0]\end{bmatrix} + \left[\begin{array}{cccc|c} \mathbf{h}[1] & \mathbf{h}[2] & \cdots & \mathbf{h}[I] & \\ \mathbf{h}[2] & \iddots & \iddots & & \\ \vdots & \iddots & & & 0\\ \mathbf{h}[I] & & & & \\ \hline & & 0 & & 0\end{array}\right]\\ & + \left[\begin{array}{c|cccc} 0 & & & 0 & \\ \hline & & & & \mathbf{h}[-I]\\ 0 & & & \iddots & \vdots\\ & & \iddots & \iddots & \mathbf{h}[-2]\\ & \mathbf{h}[-I] & \cdots & \mathbf{h}[-2] & \mathbf{h}[-1]\end{array}\right].\end{aligned} \tag{44}$$

We can now define the non-downsampled filtering matrix $\mathbf{H}_{i,K}^{(\ell)}$ with kernel $\mathbf{h}_i^K$ at level ℓ by

$$\mathbf{H}_{i,K}^{(\ell)}=\mathbf{S}^{(\ell)}(\mathbf{h}_i^K)\otimes\mathbf{I}^{(\ell)}, \tag{45}$$

where $\mathbf{I}^{(\ell)}$ is the identity matrix of size $K^\ell\times K^\ell$. For any vector $\mathbf{w}$, the filtering process $\mathbf{v}=\mathbf{H}_{i,K}^{(\ell)}\mathbf{w}$ can be understood as follows. We first break up $\mathbf{w}$ into $\mathbf{w}_t$ which are defined by $\mathbf{w}_t=[\mathbf{w}[j]]$ with

$j \mod K^\ell = t$ for $t = 0, \cdots, K^\ell - 1$. Then we apply the filtering by $\mathbf{v}_t = \mathbf{S}^{(\ell)}(\mathbf{h}_i^K)\mathbf{w}_t$ for each t. Finally, we combine $\mathbf{v}_t$ and permute them back by $\mathbf{v}[t + sK^\ell] = \mathbf{v}_t[s]$. Then $\mathbf{v} = \mathbf{H}_{i,K}^{(\ell)}\mathbf{w}$.

With these notations, the transform we used in super-resolution image reconstruction is

$$\mathbf{A} := \mathbf{A}_{K,L} = \begin{bmatrix} \prod_{\ell=0}^{L-1} \mathbf{H}_{0,K}^{(L-\ell)} \\ \mathbf{H}_{1,K}^{(L)} \prod_{\ell=1}^{L-1} \mathbf{H}_{0,K}^{(L-\ell)} \\ \vdots \\ \mathbf{H}_{r,K}^{(L)} \prod_{\ell=1}^{L-1} \mathbf{H}_{0,K}^{(L-\ell)} \\ \vdots \\ \vdots \\ \mathbf{H}_{1,K}^{(1)} \\ \vdots \\ \mathbf{H}_{r,K}^{(1)} \end{bmatrix}. \tag{46}$$

If the size of the image to be recovered is N, then the size of the coefficients in the transformed domain is $M = rNL + N$. If all K low-resolution images are available, then the set of indices of known data is

$$\Gamma = \{1, 2, \cdots, r(L-1)N + N\}, \quad \text{and} \quad \Lambda = \emptyset. \tag{47}$$

Sometimes, not all K sensors are available to give us the low-resolution images. In that case, the set Γ will even be smaller. To be precise, let the available low-resolution images be from the sensors $1 \leq k_1, k_2, \cdots, k_s \leq K$ with $1 \leq s \leq K$. Then the set of indices of known data is

$$\Gamma = \{k_1, k_2, \cdots, k_s\} \oplus \{0, K, 2K, \cdots, N-K\} \oplus \{0, N, 2N, \cdots, r(L-1)N\}, \quad \text{and} \quad \Lambda = \emptyset, \tag{48}$$

where $\oplus$ is defined as $\Omega_1 \oplus \Omega_2 = \{a + b : a \in \Omega_1, b \in \Omega_2\}$.

With above $\mathbf{A}$, Γ, and Λ, the problem of super-resolution image reconstruction with multiple sensors can be formulated by the model (2). Therefore, the proposed iterative algorithms (5) and (14) in the image and transformed domains can be applied to the problem. Theorems 1 and 2 guarantee the convergence of the corresponding algorithms.

3.2 Super-Resolution Image Reconstruction from Different Zooms

In this subsection, we consider restoring a high-resolution image from low-resolution images of different zooms. We will see that it is an inpainting problem on both domains with $\Lambda \neq \emptyset$ and $\Gamma \neq \emptyset$.

The problem setting is that p images $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_p\}$ of a scene or part of it are taken with increasing zoom ratio, such that all p images have the same number of pixels but with different resolutions. The image $\mathbf{p}_p$ has the highest resolution but covers the smallest part of the scene, and $\mathbf{p}_1$ is the least zoomed image but covers the entire scene. The problem is to obtain the whole scene covered by $\mathbf{p}_1$ at the resolution same as that of $\mathbf{p}_p$. Figure 1 depicts the case where $p = 3$ and the areas covered by the observed images $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$. Figure 5(a)–(c) give $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ for real images "Boat" and "Goldhill".

We now formulate the problem using the model (2) with appropriate transform matrix $\mathbf{A}$ and index sets Λ and Γ. For simplicity, we only formulate it in 1D. It can easily be extended to 2D images. Details can be found in [38]. Let the vectors $\{\mathbf{p}_i\}_{i=1}^p$ be all of length P. Let $\mathbf{f} \in \mathbb{R}^N$ be the solution (i.e. the scene at the highest resolution—same resolution as $\mathbf{p}_p$'s.). Note that $\mathbf{p}_p$ has the

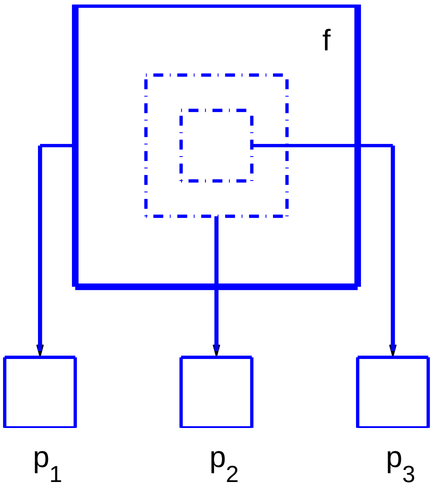


Figure 1: A diagram showing how the images $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ of different resolutions are taken from part of the scene or the whole scene.

same resolution as $\mathbf{f}$ but just covers part of the scene. Thus $\mathbf{p}_p$ is a portion of $\mathbf{f}$ cropped directly from $\mathbf{f}$. Hence $\mathbf{p}_p$ can be viewed as the given known data in the image domain. Let $T_p + 1$ be the starting pixel position of $\mathbf{p}_p$ in $\mathbf{f}$. Then in the image domain, the set Λ in (2) is

$$\Lambda = \{T_p + 1, \cdots, T_p + P\}. \tag{49}$$

The other image $\{\mathbf{p}_i\}_{i=1}^{p-1}$ can be seen as the low-resolution images obtained from part of the high-resolution image $\mathbf{f}$. We assume that the zoom factors between $\mathbf{p}_i$ and $\mathbf{p}_p$ are integers K_i. It is clear that $K_i > K_j$ if $i < j$. Using the super-resolution image reconstruction model we discussed in Section 3.1, $\mathbf{p}_i$ are just part of the low-resolution images of $\mathbf{f}$ obtained by one of the sensors of the K_i sensor array. As in (46), we define $\mathbf{A}_{K_i,L_i}$ to be the L_ith level transform matrix associated with the K_i sensor array. In particular, we restrict $L_i = 2$ for $2 \leq i < p$. Then the set of indices of known coefficients in $\mathbf{A}_{K_i,L_i}\mathbf{f}$ is

$$\Gamma_i = \{k_i\} \oplus \{T_i K_i, \cdots, (T_i + P - 1)K_i\} \oplus \{0, N, 2N, \cdots, r_i(L_i - 1)N\}.$$

where $r_i + 1$ is the number of framelet filters derived from $\mathbf{h}_0^{K_i}$. Since the scene is the same as that of $\mathbf{p}_1$, we have $T_1 = 0$ and $P = N/K_1$.

Therefore, if we define the tight frame system by

$$\mathbf{A} := \frac{1}{\sqrt{p-1}} \begin{bmatrix} \mathbf{A}_{K_1,L_1} \\ \vdots \\ \mathbf{A}_{K_{p-1},L_{p-1}} \end{bmatrix}, \tag{50}$$

then the set of indices of the known data in the transformed domain is

$$\Gamma = \bigcup_{i=1}^{p-1} (\{R_i\} \oplus \Gamma_i), \tag{51}$$

where $R_1 = 0$ and $R_i = \sum_{j=1}^{i-1}(r_j L_j + 1)N$ for $i > 1$ is the starting position of the $\mathbf{A}_{K_i,L_i}\mathbf{f}$ in $\mathbf{Af}$. Hence, the problem of recovering $\mathbf{f}$ from different $\mathbf{p}_i$ can be cast into the model (2) with $\mathbf{A}$ in (50), Λ in (49), and Γ in (51). We can use our algorithm (5) to find $\mathbf{f}$. The convergence of the algorithm (5) is guaranteed by By Theorems 1.

3.3 Numerical Results

We now illustrate the applicability of our proposed algorithm (5) for the applications presented in the last two subsections. We use the "Boat" and "Goldhill" images of size 256×256 as the original images in our numerical tests. The objective quality of the reconstructed images is evaluated by the peak signal-to-noise ratio (PSNR) which is defined as

$$\mathrm{PSNR} = 20 \log_{10} \left(\frac{255\sqrt{N}}{\|\mathbf{f} - \mathbf{f}^{(\infty)}\|_2} \right), \tag{52}$$

with $\mathbf{f}$ being the original image, $\mathbf{f}^{(\infty)}$ the reconstructed image by (5), and N the number of pixels in $\mathbf{f}^{(\infty)}$. The initial seed $\mathbf{f}^{(0)}$ is chosen as zero. The maximum number of iteration is set to 100 and the iteration process is stopped when the reconstructed image achieves the highest PSNR value.

We first consider super-resolution image reconstruction with multiple sensors. We used a 4×4 sensor array (i.e. $K = 4$). In this case, we choose the parameter $r = 5$ in (46). The six filters $\mathbf{h}_i^4$, $i = 0, 1, 2, 3, 4, 5$, are

$$\begin{aligned}
&\mathbf{h}_0^4 = \frac{1}{4}\left[\frac{1}{2}, 1, 1, 1, \frac{1}{2}\right], \mathbf{h}_1^4 = \frac{\sqrt{2}}{8}\left[1, 0, 0, 0, -1\right], \mathbf{h}_2^4 = \frac{1}{4}\left[-\frac{1}{2}, 1, -1, , 1, -\frac{1}{2}\right]\\
&\mathbf{h}_3^4 = \frac{1}{4}\left[\frac{1}{2}, 1, 0, -1, -\frac{1}{2}\right], \mathbf{h}_4^4 = \frac{\sqrt{2}}{8}\left[1, 0, -2, 0, 1\right], \mathbf{h}_5^4 = \frac{1}{4}\left[-\frac{1}{2}, 1, 0, -1, \frac{1}{2}\right].
\end{aligned}$$

The sum of the absolute values of the elements in $\mathbf{h}_i^4$, denoted by c_i, $i = 0, 1, 2, 3, 4, 5$, are 1, $\frac{\sqrt{2}}{2}$, 1, $\frac{3}{4}$, $\frac{\sqrt{2}}{2}$, and $\frac{3}{4}$, respectively. The parameter L in (46) is 4. To determine the thresholding parameters u_i in (4), we first compute a threshold β which basically is a scaled version of the threshold given in [31], that is, $\beta = \frac{1}{64}\sigma\sqrt{2\log N}$. The parameter σ is the estimated standard deviation of the Gaussian noise in the observed image. The factor $\frac{1}{64}$ is empirically chosen based on our experiments. We choose $u_i = c_p\beta$ if the corresponding framelet coefficient y_i is produced by the subblock $\mathbf{H}_{p,4}^{(j)} \prod_{\ell=1}^{j-1} \mathbf{H}_{0,4}^{(j-\ell)}$ in (46), where $1 \le i \le 5$, and $1 \le j \le L$. Figures 3 and 4 show the reconstructed images when noise at signal-to-noise ratio 30dB is added to the observed images.

Next we test our algorithm for super-resolution image reconstruction from different zooms. In the test, we set $p = 3$, $K_1 = 4$, and $K_2 = 2$. The image $\mathbf{p}_1$ is the lowest-resolution image of the scene $\mathbf{f}$ as obtained from a 4×4 sensor array. The image $\mathbf{p}_2$ is a low-resolution image of part of $\mathbf{f}$ as obtained from a 2×2 sensor array. The image $\mathbf{p}_3$ is a part of $\mathbf{f}$. The images $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ all have the same size and are 1/16 of the size of $\mathbf{f}$. The images and their restored results are displayed in Figure 5.

4 Conclusion

In this paper, we have developed an algorithm for doing inpainting in both image and transformed domains. The proposed algorithm is motivated by the perfect reconstruction formula of a tight framelet system. This algorithm is further recognized as a proximal forward-backward splitting iteration of a minimization problem with the help of proximal operator and proximal envelop. By using the convergence result of PFBS iteration, we have proved the convergence of the algorithm, and obtained the functionals which the limits will minimize. We have illustrated how to apply the method to super-resolution image reconstruction from different zooms.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, *Image inpainting*, in Proceedings of SIGGRAPH, New Orleans, LA, 2000, pp. 417–424.

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

[3] M. Bertero and P. Boccacci, *Introduction to inverse problems in imaging*, Institute of Physics Pub., Bristol, 1998.

[4] M. Bertero, P. Boccacci, F. D. Benedetto, and M. Robberto, *Restoration of chopped and nodded images in infrared astronomy*, Inverse Problem, 15 (1999), pp. 345–372.

[5] 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.

[6] N. Bose and K. Boo, *High-resolution image reconstruction with multisensors*, International Journal of Imaging Systems and Technology, 9 (1998), pp. 294–304.

[7] J.-F. Cai, R. Chan, L. Shen, and Z. Shen, *Restoration of chopped and nodded images by framelets*, SIAM Journal on Scientific Computing, 30(3) (2008), pp. 1205–1227.

[8] J.-F. Cai, R. Chan, and Z. Shen, *A framelet-based image inpainting algorithm*, Applied and Computational Harmonic Analysis, 24 (2008), pp. 131–149.

[9] J.-F. Cai, R. Chan, and Z. Shen, *Simultaneous Cartoon and Texture Inpainting*, preprint, 2008.

[10] J.-F. Cai, S. Osher, and Z. Shen, *Linearized Bregman iterations for compressed sensing*, Mathematics of Computations, to appear.

[11] J.-F. Cai, S. Osher, and Z. Shen, *Convergence of the linearized Bregman iteration for ℓ_1-norm minimization*, Mathematics of Computations, to appear.

[12] J.-F. Cai, S. Osher, and Z. Shen, *Linearized Bregman iterations for frame-based image deblurring*, SIAM Journal on Imaging Sciences, to appear.

[13] E. J. Candès and D. L. Donoho, *New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities*, Comm. Pure Appl. Math., 57 (2004), pp. 219–266.

[14] E. 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.

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

[16] 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.

[17] ——, *Wavelet deblurring algorithms for spatially varying blur from high-resolution image reconstruction*, Linear Algebra and its Applications, 366 (2003), pp. 139–155.

[18] R. Chan, S. D. Riemenschneider, L. Shen, and Z. Shen, *High-resolution image reconstruction with displacement errors: A framelet approach*, International Journal of Imaging Systems and Technology, 14 (2004), pp. 91–104.

[19] ——, *Tight frame: The efficient way for high-resolution image reconstruction*, Applied and Computational Harmonic Analysis, 17 (2004), pp. 91–115.

[20] R. Chan, L. Shen, and Z. Shen, *A framelet-based approach for image inpainting*, Tech. Report 2005-4, The Chinese University of Hong Kong, Feb. 2005.

[21] R. Chan, Z. Shen, and T. Xia, *A framelet algorithm for enchancing video stills*, Applied and Computational Harmonic Analysis, 23 (2007), pp. 153–170.

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

[23] T. Chan and J. Shen, *Mathematical models for local nontexture inpaintings*, SIAM J. Applied Mathematics, 62 (2001), pp. 1019–1043.

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

[25] P. Combettes and V. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, 4 (2005), pp. 1168–1200.

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

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

[28] I. Daubechies, G. Teschke, and L. Vese, *Iteratively solving linear inverse problems under general convex constraints*, Inverse Probl. Imaging, 1 (2007), pp. 29–46.

[29] A. H. Delaney and Y. Bresler, *A fast and accurate Fourier algorithm for iterative parallel-beam tomography*, IEEE Trans. Image Process., 5 (1996), pp. 740–753.

[30] M. N. Do and M. Vetterli, *The contourlet transform: an efficient directional multiresolution image representation*, IEEE Transactions on Image Processing, 14 (2005), pp. 2091–2106.

[31] D. Donoho and I. Johnstone, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, 81 (1994), pp. 425–455.

[32] M. Elad and A. Feuer, *Restoration of a single superresolution image from several blurred, noisy and undersampled measured images*, IEEE Transactions on Image Processing, 6 (1997), pp. 1646–1658.

[33] M. Elad, P. Milanfar, and R. Rubinstein, *Analysis versus synthesis in signal priors*, Inverse Problems, 23 (2007), pp. 947–968.

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

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

[36] O. G. Guleryuz, *Nonlinear approximation based image recovery using adaptive sparse reconstruction and iterated denoising: Part II - adaptive algorithms*, IEEE Transaction on Image Processing, (2006).

[37] J.-B. Hiriart-Urruty and C. Lemarechal, *Convex analysis and minimization algorithms*, vol. 305 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer-Verlag, Berlin, 1993.

[38] M. V. Joshi, S. Chaudhuri, and R. Panuganti, *Super-resolution imaging: use of zoom as a cue*, Image and Vision Computing, 22 (2004), pp. 1185–1196.

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

[40] M. K. Ng and N. Bose, *Analysis of displacement errors in high-resolution image reconstruction with multisensors*, IEEE Transactions on Circuits and Systems—I: Fundamental Theory and Applications, 49 (2002), pp. 806–813.

[41] M. K. 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.

[42] 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.

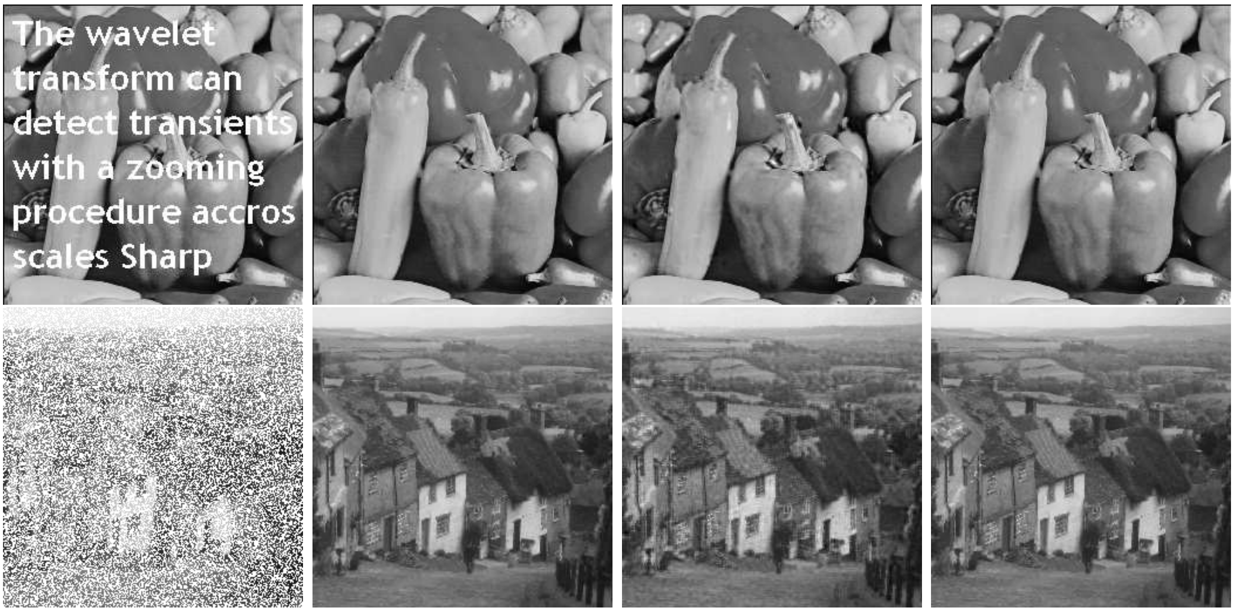


Figure 2: Inpainting in image domain for the “Pepper” and “Goldhill” images. Columns (from left to right) are the observed incomplete image, the recovered image by (5), the recovered image by the synthesis based approach, and the recovered image by the analysis based approach, respectively. In the first row, the PSNR values of the recovered images (from the second left to the right) are 33.82dB, 32.27dB, 33.73dB, respectively. In the second row, the PSNR values of the recovered images (from the second left to the right) are 29.81dB, 28.10dB, 29.80dB, respectively.

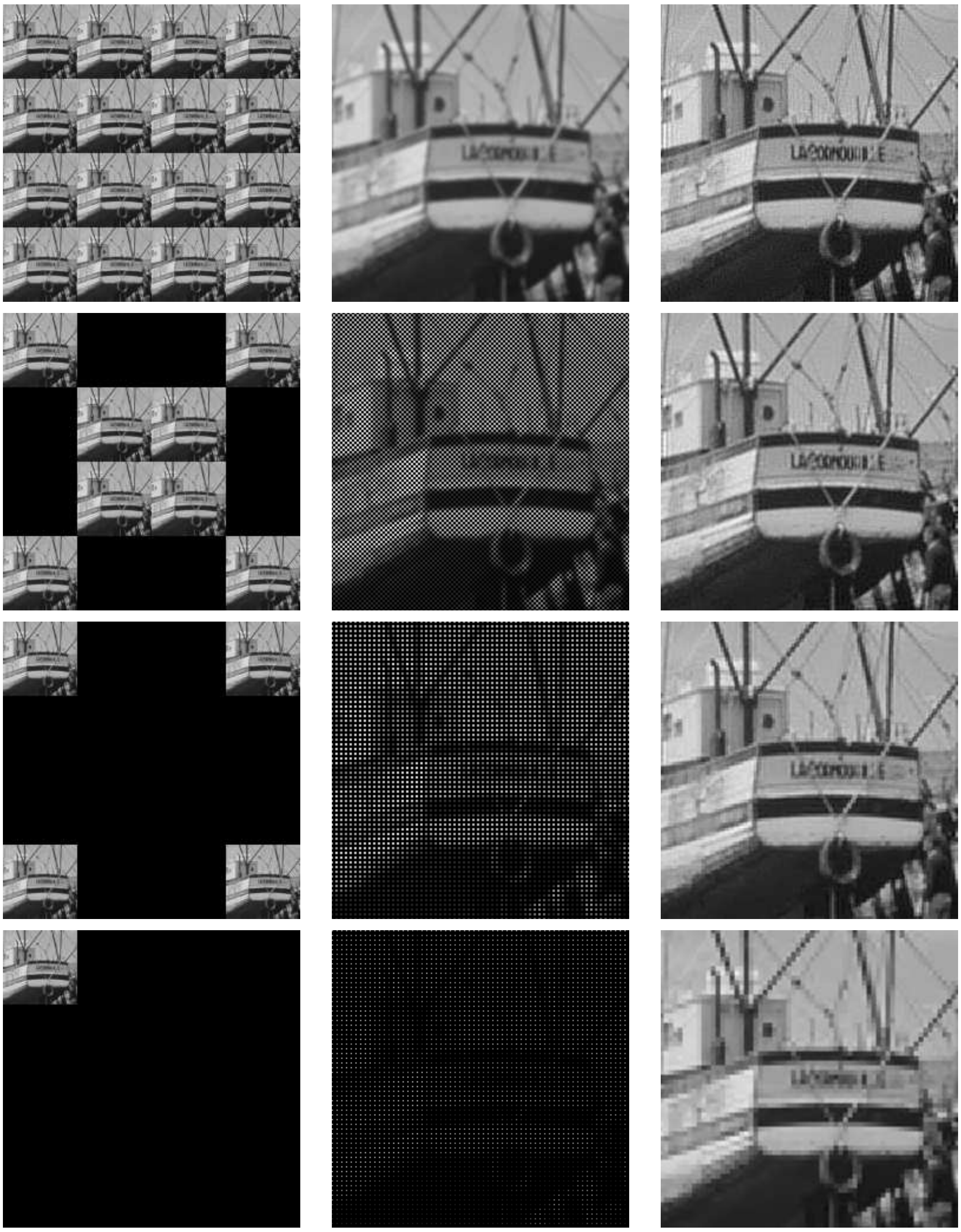


Figure 3: 4 × 4 sensor array for the "Boat" image. Columns (from left to right) show the available low-resolution images, the observed high-resolution images, and the reconstructed high-resolution images, respectively. The PSNR values of the reconstructed high-resolution images (from top to bottom) are 29.76dB, 29.01dB, 26.78dB, and 23.91dB for the cases where sixteen, eight, four, and one low-resolution images are available. The corresponding numbers of iterations are 62, 92, 100, and 100, respectively.

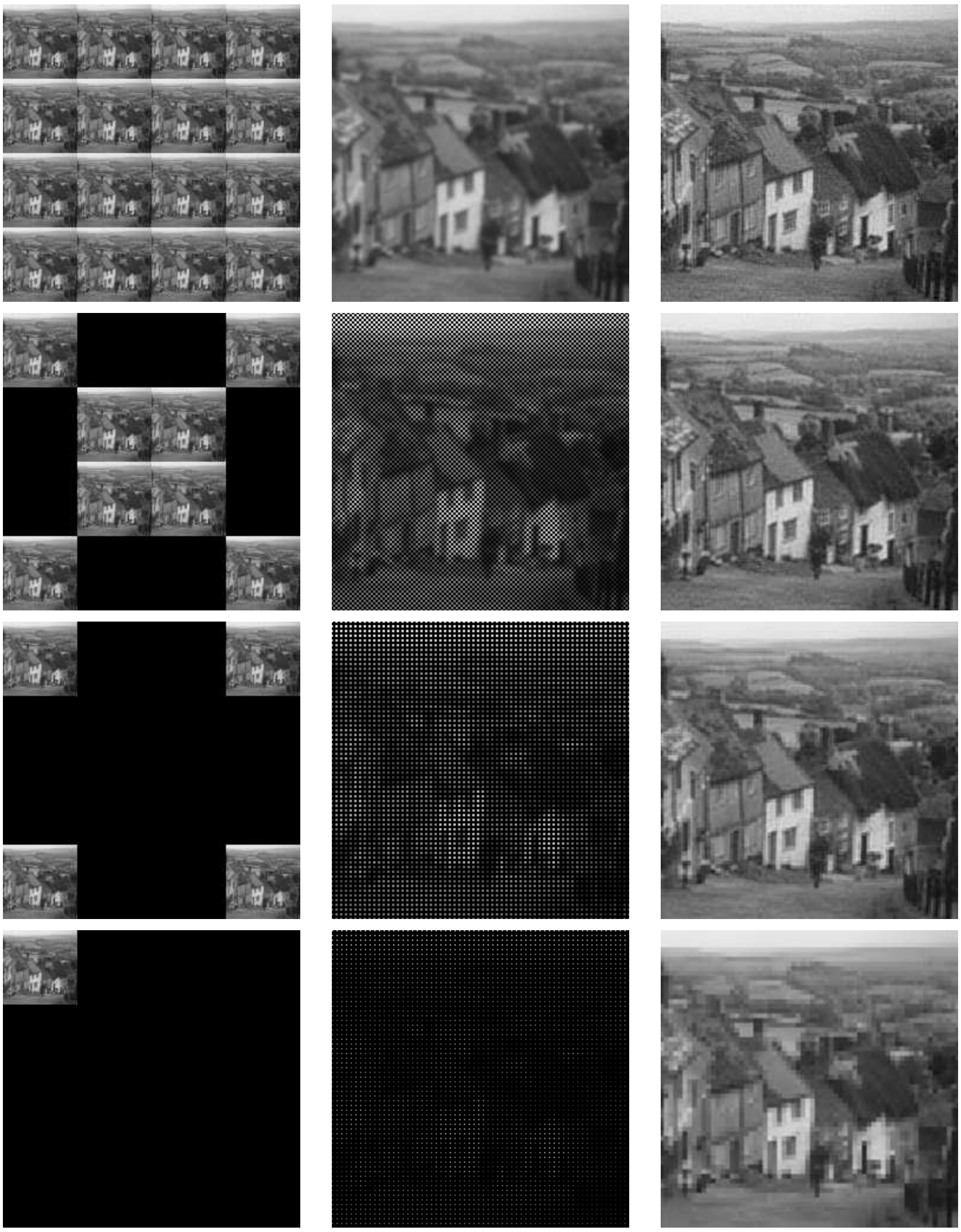
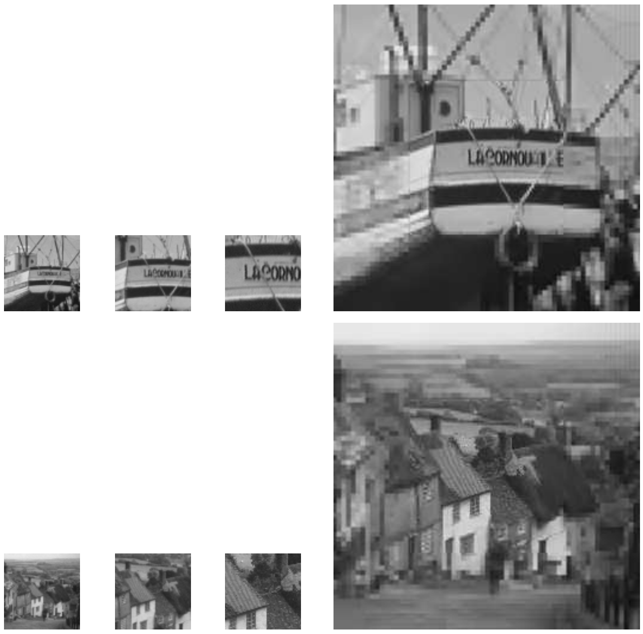


Figure 4: 4×4 sensor array for the "Goldhill" image. Columns (from left to right) show the available low-resolution images, the observed high-resolution images, and the reconstructed high-resolution images, respectively. The PSNR values of the reconstructed high-resolution images (from top to bottom) are 28.51dB, 27.93dB, 26.49dB, and 24.58dB for the cases where sixteen, eight, four, and one low-resolution images are available. The corresponding numbers of iterations are 92, 100, 100, and 100, respectively.



(a) $\mathbf{p}_1$ (b) $\mathbf{p}_2$ (c) $\mathbf{p}_3$ (d) reconstructed image

Figure 5: Reconstructed super-resolution images for images "Boat" (top row) and "Goldhill" (bottom row), respectively, from the related low-resolution images $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$.