An Accelerated Proximal Gradient Algorithm for Frame-Based Image Restoration via the Balanced Approach

Zuowei Shen * Kim-Chuan Toh † Sangwoon Yun ‡

March 1, 2011

Abstract

Frame-based image restoration by using balanced approach has been developed over the last decade. Many recently developed algorithms for image restoration can be viewed as an acceleration of the proximal forward-backward splitting algorithm. Accelerated proximal gradient algorithms studied by Nesterov, Nemirovski, and others have been demonstrated to be efficient in solving various regularized convex optimization problems arising in compressed sensing, machine learning, and control. In this paper, we adapt the accelerated proximal gradient (APG) algorithm to solve the ℓ_1-regularized linear least squares problem in the balanced approach in frame-based image restoration. This algorithm terminates in $O(1/\sqrt{\epsilon})$ iterations with an ϵ-optimal solution, and we demonstrate that this single algorithmic framework can universally handle several image restoration problems, such as image deblurring, denoising, inpainting, and cartoon-texture decomposition. Our numerical results suggest that the APG algorithms are efficient and robust in solving large-scale image restoration problems. The algorithms we implemented are able to restore 512×512 images in various image restoration problems in less than 50 seconds on a modest PC. We also compare the numerical performance of our proposed algorithms applied to image restoration problems by using one frame-based system with that by using cartoon and texture systems for image deblurring, denoising, and inpainting.

Key words. Balanced approach, analysis based approach, synthesis based approach, image restoration, iteration complexity, ℓ_1-regularized convex minimization, proximal gradient algorithms

*Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (matzuows@nus.edu.sg).

†Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (mattohkc@nus.edu.sg); and Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576.

‡Korea Institute for Advanced Study, Hoegiro 87, Dongdaemun-gu, Seoul 130-722, Korea(yswcs@kias.re.kr).
This author was supported in part by BSRP (2010-0510-1-3) through NRF funded by the Ministry of Education, Science and Technology.

1 Introduction

Image restoration is often formulated as an inverse problem. The objective is to find the unknown true image $u \in \Re^n$ from an observed image (or measurements) $b \in \Re^\ell$ defined by

$$b = Au + \eta, \tag{1}$$

where η is a white Gaussian noise with variance σ^2, and $A \in \Re^{\ell \times n}$ is a linear operator, typically a convolution operator in image deconvolution, a projection in image inpainting and the identity map in image denoising.

Our approach for image restoration is based on *tight frames*. For simplicity, we denote images by vectors in $\Re^n$ by concatenating their columns. Tight frames are redundant system in $\Re^n$. In particular, suppose $W \in \Re^{m \times n}$ (with $m \geq n$) satisfies $W^T W = I$, where I is the identity matrix. Then, the rows of W form a *tight frame* in $\Re^n$. Thus, for every vector $u \in \Re^n$,

$$u = W^T(Wu).$$

The components of the vector Wu are called the canonical coefficients representing u. In this paper, the tight frame system W used is generated from piecewise linear B-spline framelet constructed via the unitary extension principle in [40]. We refer interested readers to [21, 40] and the references therein for the general wavelet frame theory and its corresponding construction. The details in the construction of W from a given wavelet tight frame system can be found in, for example, [6, 7, 8, 9, 13, 14, 15, 16, 17].

In the case of redundant tight frame systems, the mapping from the image u to its coefficients is not one-to-one, i.e., the representation of u in the frame domain is not unique. Therefore, there are two formulations for the sparse approximation of the underlying images, namely analysis based and synthesis based approaches. The analysis based approach was first proposed in [23, 24]. In this approach, we assume that the analyzed coefficient vector Wu can be sparsely approximated, and it is usually formulated as a linear least squares problem involving a penalty on the term $\|Wu\|_1$. The synthesis based approach was first introduced in [22, 25, 26, 27, 28]. In this approach, the underlying image u is assumed to be synthesized from a sparse coefficient vector x with $u = W^T x$, and it is usually formulated as a linear least squares problem involving a penalty on the term $\|x\|_1$.

The balanced approach was first used in [15, 16] for high resolution image reconstruction. It was further developed for various image restoration problems in [6, 7, 8, 9, 13, 14, 17]. Although the synthesis based, analysis based and balanced approaches are developed independently in the literature, the balanced approach can be motivated from our desire to balance the analysis based and synthesis based approaches.

Before giving the mathematical formulations of the above three approaches, we set up some notations. For any $x \in \Re^n$, $\|x\|_p^p = \sum_{j=1}^n |x_j|^p$, $1 \leq p < \infty$. For simplicity, we write $\|x\| = \|x\|_2$ and $|x|$ denotes the vector obtained from x by taking absolute values of its components. We use $\|x\|_D$ to denote the D-norm, where D is a symmetric positive definite matrix, defined by $\|x\|_D = \sqrt{x^T D x}$. For any real symmetric matrices H_1 and H_2, $\lambda_{\max}(H_1)$ denotes the maximum eigenvalue of H_1, and we write $H_1 \succeq H_2$ (respectively, $H_1 \succ H_2$) to mean that $H_1 - H_2$ is positive semidefinite (respectively, positive definite). For any $m \times n$ real matrix A, $\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$. We use I to denote the identity matrix and e to denote the vector of ones.

The balanced approach can be formulated as the following ℓ_1-regularized linear least squares

problem:

$$\min_{x\in\Re^m} \quad \frac{1}{2}\|AW^Tx-b\|_D^2+\frac{\kappa}{2}\|(I-WW^T)x\|^2+\lambda^T|x|, \tag{2}$$

where $\kappa>0$, λ is a given nonnegative weight vector, and D is a given symmetric positive definite matrix. When $\kappa=0$, the problem (2) is reduced to the synthesis based approach:

$$\min_{x\in\Re^m} \quad \frac{1}{2}\|AW^Tx-b\|_D^2+\lambda^T|x|. \tag{3}$$

On the other extreme, when $\kappa=\infty$, the problem (2) is reduced to the analysis based approach. To see this, we note that the term $\|(I-WW^T)x\|$ must be 0 when $\kappa=\infty$. This implies that x is in the range of W, i.e., $x=Wu$ for some $u\in\Re^n$, so we can rewrite (2) as

$$\min_{x\in\text{Range}(W)} \quad \frac{1}{2}\|AW^Tx-b\|_D^2+\lambda^T|x| \;=\; \min_{u\in\Re^n} \quad \frac{1}{2}\|Au-b\|_D^2+\lambda^T|Wu| \tag{4}$$

The problem (4) is the analysis based minimization problem. It is clear that when $0<\kappa<\infty$, the problem (2) balances between (3) and (4), hence is called the balanced approach.

The objective function of the balanced approach (2) is convex but may not be strictly convex and hence the optimal solution need not be unique. To avoid having a non-unique minimizer, we consider the following modified balanced approach:

$$\min_{x\in\Re^m} \quad \frac{1}{2}\|AW^Tx-b\|_D^2+\frac{\kappa}{2}\|(I-WW^T)x\|^2+\frac{\alpha}{2}\|x\|^2+\lambda^T|x|, \tag{5}$$

where $\alpha\geq 0$ and κ, λ, D are as defined in (2). If $\alpha=0$ then (5) reduces to (2). If $\alpha>0$, then the objective function is strictly convex and so the optimal solution is unique. The term $\frac{\alpha}{2}\|x\|^2$ is also known as the Tikhonov regularization [4] which has been used to handle the ill-conditioning of the operator A in image deconvolution. As we will show later in the paper, in the limit when $\alpha\downarrow 0$, the minimizer of the modified model (5) will converge to the minimizer of (2) with the least Euclidean norm.

We note that when the rows of W form an orthonormal basis, instead of being a redundant tight frame, the above three approaches are exactly the same, since $WW^T=I$ in this case. However, for a redundant tight frame system W, the analysis based, synthesis based and balanced approaches cannot be derived from one another. In general, it is hard to draw definitive conclusions as to which approach is better since each of the three approaches has its own favorable data sets and applications. Our balanced approach bridges the analysis based and synthesis based approaches in image restoration and it provides an additional approach in the rich literature of frame-based image restoration as shown in [6, 7, 8, 13, 14, 15, 16, 17].

Recently, the linearized Bregman iteration was proposed for solving the ℓ_1-minimization problems in compressed sensing by [10, 38, 43] and the nuclear norm minimization in matrix completion by [5]. The linearized Bregman iteration was then used to develop a fast algorithm for the synthesis based approach for frame-based image deblurring in [11]. Furthermore, the split Bregman iteration proposed in [30] was shown to be powerful in [30, 45] when it is applied to various PDE based image restoration approaches, e.g., ROF and nonlocal PDE models. The split Bregman iteration is further used to develop a fast algorithm for the analysis based approach in frame-based image restoration in [12]. While the balanced approach in frame-based image restoration gives satisfactory simulation results, as shown in [6, 7, 8, 13, 14, 15, 16, 17], when solved by a variant of

the proximal forward-backward splitting algorithm, but the numerical convergence speed achieved is not as fast as the synthesis based approach when solved by the linearized Bregman iteration, or the analysis based approach when solved by the split Bregman iteration.

The main goals of this paper are as follows. (i) to find a fast algorithm, for the balanced approach (2) and the modified one (5) in frame-based image restoration, whose convergence speed is competitive to that of the linearized Bregman iteration for the synthesis based approach and the split Bregman iteration for the analysis based approach, and (ii) to demonstrate that a single simple algorithm, with proper implementation based on a natural and robust stopping condition together with an optimal estimation of the Lipschitz constant of the gradient of the linear least squares term, can efficiently and robustly solve various types of image restoration problems, such as image deblurring, denoising, inpainting, and cartoon-texture image decomposition. With such an algorithm, one is then free to choose either the synthesis based approach, analysis based approach or balanced approach according to his/her priorities and applications without the constraint of computational consideration. In this paper, we demonstrate that by adapting the accelerated proximal gradient (APG) algorithm of Beck and Teboulle [1] to solve the problem (5) (and so (2)) on large-scale problems arising in various image restoration problems, we are able to achieve the objectives outlined above. We also give a practical sub-gradient based stopping condition for our adapted APG algorithm besides the commonly used stopping condition that is based on the relative error of the iterates (which can cause early termination). This sub-gradient based stopping condition is easy to check without incurring extra computational cost; see Section 4.1.

The APG algorithm of Beck and Teboulle is based on several variants of APG algorithms that were studied earlier on by Nesterov and Nemirovski; see [34, 35, 36, 37]. More recently, Tseng [42] gave a unified study of APG algorithms, for convex-concave optimization, that included all the previously proposed algorithms. The APG algorithms of Nesterov, and Beck and Teboulle, have been adapted to solve ℓ_1-regularized linear least squares problems arising in signal/image processing [1], compressed sensing [2, 33] and nuclear norm regularized linear least squares problems [41]. Compared to projected gradient and proximal forward-backward splitting algorithms, APG algorithms have an attractive iteration complexity of $O(1/\sqrt{\epsilon})$ for achieving ϵ-optimality; see Section 2. Also APG algorithms are simple and use only the soft-thresholding operator when applied to ℓ_1-regularized linear least squares problems, just like algorithms such as the linearized Bregman iteration, the split Bregman iteration and the proximal forward-backward splitting algorithm.

We also extend the APG algorithm for (5) (and hence (2)) to the balanced approach in image restoration of two-layered images. Real images usually have two layers, referring to cartoons (the piecewise smooth part of the image) and textures (the oscillating pattern part of the image). The layers usually have sparse approximations under different tight frame systems. Therefore, those two different layers should be considered separately. One natural idea is to use two tight frame systems that can sparsely represent cartoons and textures separately. The corresponding image restoration problem can be formulated as the following ℓ_1-regularized linear least squares problem:

$$\min_{x_1\in\Re^{m_1},x_2\in\Re^{m_2}} \quad \frac{1}{2}\|A(\sum_{i=1}^{2} W_i^T x_i) - b\|_D^2 + \sum_{i=1}^{2}\left(\frac{\kappa_i}{2}\|(I - W_i W_i^T)x_i\|^2 + \frac{\alpha_i}{2}\|x_i\|^2\right) + \sum_{i=1}^{2}\lambda_i^T|x_i|. \quad (6)$$

where, for $i = 1, 2$, $W_i^T W_i = I$, $\kappa_i > 0$, $\alpha_i \geq 0$, λ_i is a given nonnegative weight vector, and D is a given symmetric positive definite matrix.

The paper is organized as follows. In Section 2, we adapt the APG algorithm of Beck and Teboulle for solving (5) and (6). We give the iteration complexity of the algorithm. We show

that each cluster point of the iterates generated by the proposed algorithm is an optimal solution when it is applied to solve (5) with $\alpha = 0$ (i.e. (2)) and (6) with $\alpha_i = 0$. We also show that the iterates generated by the proposed algorithm converge to a unique optimal solution when it is applied to solve (5) with $\alpha > 0$ and (6) with $\alpha_i > 0$, and this unique optimal solution converges to an optimal solution of (2) (and (6) with $\alpha_i = 0$) with the minimal Euclidean norm as $\alpha \to 0$. In Section 3, we estimate the Lipschitz constants of the gradients of the smooth parts of the objective functions in (5) and (6), especially for the problems arising in image inpainting, since the Lipschitz constant is crucial for the convergence speed of our proposed algorithm. In Section 4, we give a natural and robust stopping condition of our proposed algorithm based on the sub-gradient of the objective function and present some preliminary numerical results for solving (5) and (6) on a set of large-scale problems arising from several types of image restoration problems, such as image deblurring, denoising, inpainting, and cartoon-texture image decomposition. Comparison of numerical performance of our proposed algorithms applied to image problems by using one frame-based system with that by using cartoon and texture systems is also given. Finally, we give our conclusions in Section 5.

2 Accelerated Proximal Gradient Algorithm

In this section we adapt the APG algorithm of Beck and Teboulle [1] for solving (5) and (6). We also state their iteration complexities and analyze their convergence.

In what follows,

$$f(x) = \frac{1}{2}\|AW^T x - b\|_D^2 + \frac{\kappa}{2}\|(I - WW^T)x\|^2 + \frac{\alpha}{2}\|x\|^2, \tag{7}$$

and

$$F(x) = F(x_1, x_2) = \frac{1}{2}\|A\Big(\sum_{i=1}^{2} W_i^T x_i\Big) - b\|_D^2 + \sum_{i=1}^{2}\left(\frac{\kappa_i}{2}\|(I - W_i W_i^T)x_i\|^2 + \frac{\alpha_i}{2}\|x_i\|^2\right). \tag{8}$$

Note that the gradient of f is given by

$$\nabla f(x) = WA^T D(AW^T x - b) + \kappa(I - WW^T)x + \alpha x. \tag{9}$$

and the gradient of F is given by

$$\nabla F(x) = (\nabla_{x_1} F(x)^T, \nabla_{x_2} F(x)^T)^T, \tag{10}$$

where

$$\nabla_{x_j} F(x) = W_j A^T D(AW_1^T x_1 + AW_2^T x_2 - b) + \kappa_j(I - W_j W_j^T)x_j + \alpha_j x_j,\ j = 1, 2.$$

For any $y \in \Re^m$, consider the approximation of $f(x) + \lambda^T|x|$ by replacing f with its linear approximation at y:

$$\ell_f(x; y) := f(y) + \langle \nabla f(y),\, x - y\rangle + \lambda^T|x|. \tag{11}$$

Since ∇f is Lipschitz continuous on $\Re^m$, i.e.,

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \quad \forall\ x,\ y\ \in \Re^m, \tag{12}$$

for some $L > 0$, this together with the convexity of f implies that

$$f(x) + \lambda^T|x| - \frac{L}{2}\|x-y\|^2 \le \ell_f(x;y) \le f(x) + \lambda^T|x| \quad \forall\ x,\ y\ \in \Re^n. \tag{13}$$

The main step of the APG algorithm for solving (5) uses the following subproblem:

$$\min_x \ell_f(x;y) + \frac{L}{2}\|x-y\|^2. \tag{14}$$

Since the objective function of the above subproblem is strictly convex, the solution is unique. By ignoring the constant term in y, (14) can be rewritten as follows:

$$\min_x \frac{L}{2}\|x-g\|^2 + \lambda^T|x|, \tag{15}$$

where $g = y - \nabla f(y)/L$.

For a given nonnegative vector $\nu \in \Re^m$, we define the soft-thresholding map $s_\nu : \Re^m \to \Re^m$ as follows:

$$s_\nu(x) := \mathrm{sgn}(x) \odot \max\{|x| - \nu, 0\} \tag{16}$$

where $\odot$ denotes the component-wise product, i.e., $(x \odot y)_i = x_i y_i$, and sgn is the signum function defined by

$$\mathrm{sgn}(t) := \begin{cases} +1 & \text{if } t > 0; \\ 0 & \text{if } t = 0; \\ -1 & \text{if } t < 0. \end{cases}$$

Then $s_{\lambda/L}(g)$ is the unique solution of (14) and (15).

We now describe formally the APG algorithm adapted from [1] for solving (5).

APG algorithm:

For a given nonnegative vector λ, choose $x^0 = x^{-1} \in \Re^n$, $t^0 = t^{-1} = 1$. For $k = 0, 1, 2, ...$, generate x^{k+1} from x^k according to the following iteration:

Step 1. Set $y^k = x^k + \frac{t^{k-1}-1}{t^k}(x^k - x^{k-1})$.

Step 2. Set $g^k = y^k - \nabla f(y^k)/L$,

Step 3. Set $x^{k+1} = s_{\lambda/L}(g^k)$.

Step 4. Compute $t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2}$.

When the APG algorithm with $t^k = 1$ for all k is applied to solve the problem (2), it is the popular *iterative shrinkage/thresholding* (IST) algorithms [20, 27, 28, 31] and it is also the proximal forward-backward splitting (PFBS) algorithm developed in [6, 7, 8, 9, 13, 14, 15, 16, 17] for the balanced approach in frame-based image restoration. The IST and PFBS algorithms have been developed and analyzed independently by many researchers. These algorithms only require gradient evaluations and soft-thresholding operations, so the computation at each iteration is very cheap. But, for any $\epsilon > 0$, those algorithms terminate in $O(L/\epsilon)$ iterations with an ϵ-optimal solution [1, 41]. Hence the sequence $\{x^k\}$ converges slowly. On the other hand, we show in this section that the APG algorithm adapted here gets an ϵ-optimal solution in $O(\sqrt{L/\epsilon})$ iterations.

Thus the APG algorithm accelerates the PFBS algorithm used in [6, 7, 8, 9, 13, 14, 15, 16, 17] for the balanced approach in frame-based image restoration.

We first prove that the problem (2) (i.e. (5) with $\alpha = 0$) has an optimal solution and the problem (5) with $\alpha > 0$ has a unique optimal solution.

Lemma 2.1 *There exists an optimal solution for the problem (5). In addition, the optimal solution of (5) with $\alpha > 0$ is unique.*

Proof. Note that the objective function for the problem (5) is proper, lower semicontinuous, and level bounded [39]. Therefore, by [39, Theorem 1.9], the set of minimizers is nonempty and compact. In addition, the objective function for (5) with $\alpha > 0$ is strictly convex. Hence the optimal solution for the problem (5) with $\alpha > 0$ is unique. ■

The following lemma gives an explicit bound on optimal solution set of (5). In what follows, $\mathcal{X}^*$ denotes the set of optimal solutions of (5).

Lemma 2.2 *For each positive vector λ, the optimal solution set $\mathcal{X}^*$ is bounded. In addition, for any $x^* \in \mathcal{X}^*$, we have*

$$\|x^*\|_1 \leq \chi \tag{17}$$

where

$$\chi = \begin{cases} \min\{\|b\|_D^2/2, \frac{\alpha}{2}\|x_{LS}\|^2 + \lambda^T|x_{LS}|\}/\lambda_{\min} & \text{if } A \text{ is surjective} \\ \|b\|_D^2/(2\lambda_{\min}) & \text{otherwise.} \end{cases}$$

with $\lambda_{\min} = \min_{i=1,\ldots,n}\{\lambda_i\}$ and $x_{LS} = WA^T(AA^T)^{-1}b$.

Proof. Considering the objective value of (5) at $x = 0$, we obtain that, for any $x^* \in \mathcal{X}^*$,

$$\lambda_{\min}\|x^*\|_1 \leq f(x^*) + \lambda^T|x^*| \leq \frac{1}{2}\|b\|_D^2.$$

Hence $\|x^*\|_1 \leq \|b\|_D^2/(2\lambda_{\min})$. In addition, if A is surjective, then by considering the objective value of (5) at $x = x_{LS}$, we obtain that for any $x^* \in \mathcal{X}^*$, $\lambda_{\min}\|x^*\|_1 \leq f(x^*) + \lambda^T|x^*| \leq \frac{\alpha}{2}\|x_{LS}\|^2 + \lambda^T|x_{LS}|$. ■

The following theorem gives an upper bound on the number of iterations for the APG algorithm for solving (5) to achieve ϵ-optimality. This theorem follows from [1, Theorem 4.1] or [42, Corollary 2]. Hence we omit its proof.

Theorem 2.1 *Let $\{x^k\}$, $\{y^k\}$, $\{t^k\}$ be the sequences generated by Algorithm APG when applied to solve (5). Then, for any $k \geq 1$, we have*

$$f(x^k) + \lambda^T|x^k| - f(x^*) - \lambda^T|x^*| \leq \frac{2L\|x^* - x^0\|^2}{(k+1)^2}, \qquad \forall x^* \in \mathcal{X}^*. \tag{18}$$

Hence

$$f(x^k) + \lambda^T|x^k| - f(x^*) - \lambda^T|x^*| \leq \epsilon \quad \text{whenever } k \geq \sqrt{\frac{2L}{\epsilon}}\left(\|x^0\| + \chi\right) - 1, \tag{19}$$

where χ is defined as in Lemma 2.2.

The next theorem shows that the sequence $\{x^k\}$ generated by Algorithm APG is bounded and each cluster point of the sequence is an optimal solution when it is applied to solve (5).

Theorem 2.2 *Let $\{x^k\}$ be the sequence generated by Algorithm APG when applied to solve (5) with a positive vector λ. Then $\{x^k\}$ is bounded and each cluster point of the sequence $\{x^k\}$ is an optimal solution. In addition, if $\alpha > 0$, then the sequence $\{x^k\}$ converges to a unique optimal solution.*

Proof. Since $f(x^k) \geq 0$, this together with (18) implies that we have, for any $k \geq 1$ and some $x^* \in \mathcal{X}^*$,

$$\lambda_{\min}|x^k| \leq f(x^k) + \lambda^T|x^k| \leq f(x^*) + \lambda^T|x^*| + \frac{2L\|x^* - x^0\|^2}{(k+1)^2}.$$

Hence $\{x^k\}$ is bounded. Next, we let $\bar{x}$ be a cluster point of the sequence $\{x^k\}$. Then there is a subset $\mathcal{K} \subseteq \{0, 1, ...\}$ such that $\{x^k\}_{\mathcal{K}}$ is a subsequence of $\{x^k\}$ converging to $\bar{x}$. Taking the limit as $k \in \mathcal{K}, k \to \infty$ in the inequality (18) and using the continuity of the function $f(x) + \lambda^T|x|$, we get

$$f(\bar{x}) + \lambda^T|\bar{x}| \leq f(x^*) + \lambda^T|x^*| \qquad \forall x^* \in \mathcal{X}^*.$$

Therefore $\bar{x}$ is an optimal solution for the problem (5).

If $\alpha > 0$, then f is strictly convex. Hence there is a unique optimal solution for (5). In this case, the sequence $\{x^k\}$ generated by Algorithm APG converges to the unique optimal solution. ■

The following theorem shows that the unique solution of (5) converges to a solution of (2) with the minimal Euclidean norm as $\alpha \downarrow 0$.

Theorem 2.3 *Let x^*_α be the unique solution of the problem (5) and x^*_0 be the unique solution of the following problem:*

$$\min_{x \in \mathcal{X}^*_0} \|x\|, \tag{20}$$

where $\mathcal{X}^*_0$ is the set of solutions of (2). Then $\|x^*_\alpha\|$ is a nonincreasing function of α and

$$\lim_{\alpha \to 0} \|x^*_\alpha - x^*_0\| = 0. \tag{21}$$

Proof. Let $h(x) := \frac{1}{2}\|AW^Tx - b\|^2_D + \frac{\kappa}{2}\|(I - WW^T)x\|^2$. Consider any scalars α_1, α_2 with $\alpha_1 > \alpha_2$. Since $x^*_{\alpha^1}$ and $x^*_{\alpha^2}$ are solutions of the problems (5) with $\alpha = \alpha^1$ and $\alpha = \alpha^2$ respectively, we have

$$h(x^*_{\alpha^1}) + \frac{\alpha^1}{2}\|x^*_{\alpha^1}\|^2 + \lambda^T|x^*_{\alpha^1}| \leq h(x^*_{\alpha^2}) + \frac{\alpha^1}{2}\|x^*_{\alpha^2}\|^2 + \lambda^T|x^*_{\alpha^2}|$$
$$h(x^*_{\alpha^2}) + \frac{\alpha^2}{2}\|x^*_{\alpha^2}\|^2 + \lambda^T|x^*_{\alpha^2}| \leq h(x^*_{\alpha^1}) + \frac{\alpha^2}{2}\|x^*_{\alpha^1}\|^2 + \lambda^T|x^*_{\alpha^1}|.$$

Adding the above two inequalities and rearranging terms yield

$$\frac{1}{2}(\alpha^1 - \alpha^2)(\|x^*_{\alpha^1}\|^2 - \|x^*_{\alpha^2}\|^2) \leq 0. \tag{22}$$

Hence (22) implies that $\|x^*_{\alpha^1}\| \leq \|x^*_{\alpha^2}\|$. Therefore $\|x^*_\alpha\|$ is nonincreasing. Also for all $\alpha \geq 0$, we have

$$h(x^*_\alpha) + \lambda^T|x^*_\alpha| + \frac{\alpha}{2}\|x^*_\alpha\|^2 \leq h(x^*_0) + \lambda^T|x^*_0| + \frac{\alpha}{2}\|x^*_0\|^2 \leq h(x^*_\alpha) + \lambda^T|x^*_\alpha| + \frac{\alpha}{2}\|x^*_0\|^2.$$

Thus for all $\alpha \geq 0$,

$$\|x_\alpha^*\| \leq \|x_0^*\|. \tag{23}$$

Now, we consider any sequence $\{x_{\alpha^k}^*\}$ with $\lim_{k\to\infty} \alpha^k = 0$. By (23), $\{x_{\alpha^k}^*\}$ is bounded and hence it has a convergent subsequence, i.e., there is a subset $\mathcal{K} \subseteq \{0, 1, ...\}$ such that $\{x_{\alpha^k}^* | k \in \mathcal{K}\}$ is a subsequence of $\{x_{\alpha^k}^*\}$ converging to a limit point $\tilde{x}$. Moreover, by (23),

$$\|\tilde{x}\| \leq \|x_0^*\|. \tag{24}$$

Using the continuity of $h(x) + \frac{\alpha}{2}\|x\|^2 + \lambda^T|x|$ and taking the limit as $k \in \mathcal{K}, k \to \infty$ in the following inequality:

$$h(x_{\alpha^k}^*) + \frac{\alpha^k}{2}\|x_{\alpha^k}^*\|^2 + \lambda^T|x_{\alpha^k}^*| \leq h(x_0^*) + \frac{\alpha^k}{2}\|x_0^*\|^2 + \lambda^T|x_0^*|$$

yield

$$h(\tilde{x}) + \lambda^T|\tilde{x}| \leq h(x_0^*) + \lambda^T|x_0^*|.$$

Hence $\tilde{x}$ is a solution of (2). This together with (24) and the uniqueness of x_0^* implies that $\tilde{x} = x_0^*$.

Finally, we prove (21) by contradiction. Suppose that x_α^* does not converge to x_0^* as $\alpha \downarrow 0$. Then there exists a $\sigma > 0$ and a sequence $\{\alpha^k\}$ with $\lim_{k\to\infty} \alpha^k = 0$ such that

$$\|x_{\alpha^k}^* - x_0^*\| \geq \sigma, \quad \forall k.$$

Since the sequence $\{x_{\alpha^k}^*\}$ is bounded, there is a convergent subsequence that must converge to x_0^* by the discussions above. This leads to a contradiction. ∎

Remark 1 *It was shown in [3, 29, 44] that there exists an $\alpha^* > 0$ such that, for all $\alpha \leq \alpha^*$, the unique solution of the following ℓ_2-regularized ℓ_1-minimization problem:* $\min_{x\in\Re^m} \{\|x\|_1 + \frac{\alpha}{2}\|x\|^2 : Ax = b\}$, *is also the solution of the following ℓ_1-minimization problem:* $\min_{x\in\Re^m} \{\|x\|_1 : Ax = b\}$, *with the minimal Euclidean norm. However there may not exist such an α^* for our problem (5). In fact, if the minimizer x_α^* of (5) with $\alpha > 0$ is non-zero, it will never be a minimizer of (2). We can prove this statement by contradiction. Let $h(x)$ be defined as in the proof of Theorem 2.3. Suppose x_α^* is also a minimizer of (2) and $(x_\alpha^*)_j \neq 0$. From the optimality conditions of x_α^* for (5) and (2), we get $(\nabla h(x_\alpha^*))_j + \alpha(x_\alpha^*)_j + \lambda_j \operatorname{sgn}(x_\alpha^*)_j = 0 = (\nabla h(x_\alpha^*))_j + \lambda_j \operatorname{sgn}(x_\alpha^*)_j$, and hence $(x_\alpha^*)_j = 0$, which is a contradiction.*

We observe that the APG algorithm is as simple as the IST and PFBS algorithms, and yet it has a better iteration complexity. Hence the main advantage of the APG algorithm over the algorithms just mentioned is its $O(\sqrt{L/\epsilon})$ iteration complexity. This advantage is also demonstrated by the numerical experiments in Section 4.

By using $F(x)$ instead of $f(x)$, the APG algorithm for solving (5) can be easily extended to solve (6). Hence the $O(\sqrt{L/\epsilon})$ iteration complexity and the convergence properties still hold for the APG algorithm when applied to solve (6). We omit the details here for the extended problem (6).

3 Estimation of Lipschitz Constants

The Lipschitz constant is crucial for the iteration complexity of the APG algorithm described in Section 2. From Theorem 2.1, we see that a tighter Lipschitz constant for ∇f will lead to a better

iteration complexity. In this section, we estimate the Lipschitz constants for the gradients of $f(x)$ and $F(x)$, arising from image restoration problems.

From (9), we have

$$\nabla^2 f(x) = WA^T DAW^T + \kappa(I - WW^T) + \alpha I.$$

Since $W^T W = I$ and $D \succ 0$, $\nabla^2 f(x) \succeq 0$ and so the Lipschitz constant L of ∇f can be taken to be the following upper bound: $L \leq \lambda_{\max}(\nabla^2 f(x)) \leq \lambda_{\max}(A^T DA) + \kappa + \alpha$. This leads to the following proposition.

Proposition 3.1 *For the problem (5), the Lipschitz constant L of ∇f defined in (9) has the following upper bound:*

$$L \leq \lambda_{\max}(A^T DA) + \kappa + \alpha. \tag{25}$$

For the problem (6), where one uses a two-frames system, we have the following result on the Lipschitz constant L of ∇F defined in (10).

Proposition 3.2 *For the problem (6), the Lipschitz constant L of ∇F defined in (10) has the following upper bound:*

$$L \leq \lambda_{\max}\left(\begin{pmatrix} W_1 \\ W_2 \end{pmatrix} A^T DA \begin{pmatrix} W_1^T & W_2^T \end{pmatrix}\right) + \max\{\kappa_1 + \alpha_1, \kappa_2 + \alpha_2\}. \tag{26}$$

Proof. From (10), we have

$$\begin{aligned} \nabla^2 F(x_1, x_2) &= \begin{pmatrix} W_1 A^T DAW_1^T + \kappa_1(I - W_1 W_1^T) + \alpha_1 I & W_1 A^T DAW_2^T \\ W_2 A^T DAW_1^T & W_2 A^T DAW_2^T + \kappa_2(I - W_2 W_2^T) + \alpha_2 I \end{pmatrix} \\ &= \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} A^T DA \begin{pmatrix} W_1^T & W_2^T \end{pmatrix} + \begin{pmatrix} \kappa_1(I - W_1 W_1^T) + \alpha_1 I & 0 \\ 0 & \kappa_2(I - W_2 W_2^T) + \alpha_2 I \end{pmatrix}. \end{aligned}$$

Since $W_1^T W_1 = I$, $W_2^T W_2 = I$, and $D \succ 0$, $\nabla^2 F(x_1, x_2) \succeq 0$ and so the Lipschitz constant L has the following upper bound:

$$\lambda_{\max}(\nabla^2 F(x_1, x_2)) \leq \lambda_{\max}\left(\begin{pmatrix} W_1 \\ W_2 \end{pmatrix} A^T DA \begin{pmatrix} W_1^T & W_2^T \end{pmatrix}\right) + \max\{\kappa_1 + \alpha_1, \kappa_2 + \alpha_2\}.$$

■

The above propositions derived an upper bound for the Lipschitz constants involved in a general frame-based image restoration problem. Next we consider the special case of the *image inpainting* problem using the model (5) with $D = I$. Note that for the image inpainting problem, A is a diagonal matrix with diagonals equal to 1 if the corresponding pixel values are known, but 0 otherwise. In this case, by using Proposition 3.1, one can derive straightforwardly the upper bound, $1 + \kappa + \alpha$, from (25) for the Lipschitz constant of ∇f. However, for the image inpainting problem (5) with $D = I$, we can obtain a tighter Lipschitz constant for ∇f, as shown in the next proposition.

Proposition 3.3 *For the image inpainting problem (5) with $D = I$, the Lipschitz constant L of ∇f defined in (9) is bounded by $\max\{1, \kappa\} + \alpha$. Note that A is a diagonal matrix with diagonals equal to 1 if the corresponding pixel values are known, but 0 otherwise.*

Proof. By permuting the rows of W^T if necessary, we have

$$W^T = \begin{pmatrix} W_a^T \\ W_b^T \end{pmatrix}, \quad W = \begin{pmatrix} W_a & W_b \end{pmatrix}, \quad A = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}.$$

Hence

$$\begin{aligned} \nabla^2 f(x) &= \begin{pmatrix} W_a & W_b \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} W_a^T \\ W_b^T \end{pmatrix} + \kappa(I - WW^T) + \alpha I \\ &= W_a W_a^T + \kappa(I - W_a W_a^T - W_b W_b^T) + \alpha I. \end{aligned}$$

We have two cases to consider.

Case 1. If $0 \le \kappa \le 1$. Then

$$\begin{aligned} \nabla^2 f(x) &\preceq W_a W_a^T + \kappa(I - W_a W_a^T) + \alpha I \;=\; \kappa I + (1-\kappa) W_a W_a^T + \alpha I \\ &\preceq \kappa I + (1-\kappa) I + \alpha I \;=\; I + \alpha I. \end{aligned}$$

Case 2. If $\kappa \ge 1$. Then

$$\nabla^2 f(x) \;=\; \kappa I - (\kappa - 1) W_a W_a^T - \kappa W_b W_b^T + \alpha I \;\preceq\; \kappa I + \alpha I.$$

Therefore $\lambda_{\max}(\nabla^2 f(x)) \le \max\{1, \kappa\} + \alpha$. Thus the Lipschitz constant of ∇f can be taken to be $\max\{1, \kappa\} + \alpha$. ∎

Similarly, when one uses a two-frames system for the image inpainting problem in (6) with $D = I$, the Lipschitz constant L of ∇F defined in (10) can be bounded by the shaper constant $1 + (\kappa + \alpha)_{\max} + \max\{0, 1 - \kappa_{\min}\}$, instead of the obvious bound of $2 + (\kappa + \alpha)_{\max}$ derived from (26). Here, $\kappa_{\min} = \min\{\kappa_1, \kappa_2\}$ and $(\kappa + \alpha)_{\max} = \max\{\kappa_1 + \alpha_1, \kappa_2 + \alpha_2\}$.

Proposition 3.4 *For the cartoon and texture image inpainting problem (6) with $D = I$, the Lipschitz constant L of ∇F defined in (10) is bounded by $1 + (\kappa + \alpha)_{\max} + \max\{0, 1 - \kappa_{\min}\}$.*

Proof. By permuting the rows of W_1^T and W_2^T if necessary, we have

$$W_1^T = \begin{pmatrix} W_{1a}^T \\ W_{1b}^T \end{pmatrix}, \quad W_2^T = \begin{pmatrix} W_{2a}^T \\ W_{2b}^T \end{pmatrix}, \quad A = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}.$$

Hence

$$\begin{aligned} &\nabla^2 F(x_1, x_2) \\ &= \begin{pmatrix} W_{1a} W_{1a}^T + \kappa_1(I - W_{1a} W_{1a}^T - W_{1b} W_{1b}^T) + \alpha_1 I & W_{1a} W_{2a}^T \\ W_{2a} W_{1a}^T & W_{2a} W_{2a}^T + \kappa_2(I - W_{2a} W_{2a}^T - W_{2b} W_{2b}^T) + \alpha_2 I \end{pmatrix}. \end{aligned}$$

We have two cases to consider.

Case 1. If $0 \le \kappa_{\min} \le 1$. Then

$$\begin{aligned}
\nabla^2 F(x_1,x_2) &\preceq \begin{pmatrix} W_{1a}W_{1a}^T + \kappa_1(I - W_{1a}W_{1a}^T) + \alpha_1 I & W_{1a}W_{2a}^T \\ W_{2a}W_{1a}^T & W_{2a}W_{2a}^T + \kappa_2(I - W_{2a}W_{2a}^T) + \alpha_2 I \end{pmatrix} \\
&= \begin{pmatrix} (1-\kappa_1)W_{1a}W_{1a}^T + (\kappa_1+\alpha_1)I & W_{1a}W_{2a}^T \\ W_{2a}W_{1a}^T & (1-\kappa_2)W_{2a}W_{2a}^T + (\kappa_2+\alpha_2)I \end{pmatrix} \\
&\preceq \begin{pmatrix} (\kappa_1+\alpha_1)I & 0 \\ 0 & (\kappa_2+\alpha_2)I \end{pmatrix} + (1-\kappa_{\min}) \begin{pmatrix} W_{1a}W_{1a}^T & W_{1a}W_{2a}^T \\ W_{2a}W_{1a}^T & W_{2a}W_{2a}^T \end{pmatrix} + \kappa_{\min} \begin{pmatrix} 0 & W_{1a}W_{2a}^T \\ W_{2a}W_{1a}^T & 0 \end{pmatrix} \\
&\preceq (\kappa+\alpha)_{\max} \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} + (1-\kappa_{\min}) \begin{pmatrix} W_{1a} \\ W_{2a} \end{pmatrix} \begin{pmatrix} W_{1a}^T & W_{2a}^T \end{pmatrix} + \kappa_{\min} \begin{pmatrix} 0 & W_{1a}W_{2a}^T \\ W_{2a}W_{1a}^T & 0 \end{pmatrix}.
\end{aligned}$$

This implies

$$\begin{aligned}
\lambda_{\max}(\nabla^2 F(x_1,x_2)) &\le (\kappa+\alpha)_{\max} + (1-\kappa_{\min})\lambda_{\max}\left(\begin{pmatrix} W_{1a} \\ W_{2a} \end{pmatrix} \begin{pmatrix} W_{1a}^T & W_{2a}^T \end{pmatrix} \right) + \kappa_{\min}\|W_{1a}W_{2a}^T\|_2 \\
&\le 2 + (\kappa+\alpha)_{\max} - \kappa_{\min}.
\end{aligned}$$

Case 2. If $\kappa_{\min} \ge 1$. Then

$$\begin{aligned}
&\nabla^2 F(x_1,x_2) \\
&= \begin{pmatrix} (\kappa_1+\alpha_1)I - (\kappa_1-1)W_{1a}W_{1a}^T - \kappa_1 W_{1b}W_{1b}^T & W_{1a}W_{2a}^T \\ W_{2a}W_{1a}^T & (\kappa_2+\alpha_2)I - (\kappa_2-1)W_{2a}W_{2a}^T - \kappa_2 W_{2b}W_{2b}^T \end{pmatrix} \\
&\preceq \begin{pmatrix} (\kappa_1+\alpha_1)I & 0 \\ 0 & (\kappa_2+\alpha_2)I \end{pmatrix} + \begin{pmatrix} 0 & W_{1a}W_{2a}^T \\ W_{2a}W_{1a}^T & 0 \end{pmatrix}.
\end{aligned}$$

This implies that

$$\lambda_{\max}(\nabla^2 F(x_1,x_2)) \le (\kappa+\alpha)_{\max} + \|W_{1a}W_{2a}^T\|_2 \le (\kappa+\alpha)_{\max} + 1.$$

Therefore $\lambda_{\max}(\nabla^2 F(x_1,x_2)) \le 1 + (\kappa+\alpha)_{\max} + \max\{0, 1-\kappa_{\min}\}$. Thus the Lipschitz constant of ∇F can be taken to be $1 + (\kappa+\alpha)_{\max} + \max\{0, 1-\kappa_{\min}\}$. ■

4 Numerical Implementation

In this section, we give a brief description of an acceleration strategy for the APG algorithm and describe the stopping condition for the APG algorithm. We also report some numerical results for solving a collection of ℓ_1-regularized linear least squares problems of the forms (5) and (6) arising in image restoration. In Section 4.1, we discuss a stopping condition for the APG algorithm. In Section 4.2, we apply the APG algorithm to frame-based image inpainting via the balanced approach. The carton and texture image decomposition and noise removal are also presented as

special cases. In Section 4.3, we apply the APG algorithm to the frame-based image deblurring via the balanced approach.

In order to further accelerate the convergence speed of the APG algorithm, we adopt the continuation strategy that is used in [31, 41]. We briefly describe this strategy here. If the both problems (5) and (6) are to be solved with the target parameter value $\bar{\lambda}$, we propose solving a sequence of problems of the forms (5) and (6) defined by a decreasing sequence $\{\lambda^0, \lambda^1, \ldots, \lambda^\ell = \bar{\lambda}\}$ with a given finite positive integer ℓ. When a new problem, associated with λ^{j+1}, is to be solved, the approximate solution for the current problem with $\lambda = \lambda^j$ is used as the starting point. Our computational experience showed that the performance of this continuation strategy is generally superior to that of directly applying the APG algorithm to the problem with the specified target value $\bar{\lambda}$. In our numerical experiments in sections 4.2 and 4.3, we set the initial $\lambda^0 = 10\lambda$, and update $\lambda^k = \max\{\upsilon\lambda^{k-1}, \lambda\}$, where υ is a real number in the interval $(0, 1)$, once at every 3 consecutive iterations or whenever the condition (30) is satisfied with Tol $= 10^{-2}$. We set $\upsilon = 0.8$ for the problem (5) and set $\upsilon = 0.7$ for (6). The reduction factors were found after some experimentation.

We have implemented the APG algorithms in MATLAB. All runs are performed on an Intel Xeon 3.20GHz PC with 4GB RAM, running Linux and MATLAB (Version 7.6). Throughout the experiments, we choose the initial iterate to be $x^0 = 0$.

4.1 A stopping condition for the APG algorithm

The natural stopping condition for the unconstrained convex minimization problem (5) is that $\delta(x) := \text{dist}(0; \partial(f(x) + \lambda^T|x|))$ is sufficiently small, where $\partial(\cdot)$ denotes the sub-differential map. Here $\text{dist}(x; S)$ denotes the distance between a point x and a set S. Since

$$\partial|x_i| = \begin{cases} \{+1\} & \text{if } x_i > 0; \\ [-1, 1] & \text{if } x_i = 0; \\ \{-1\} & \text{if } x_i < 0, \end{cases}$$

an upper bound on $\delta(x)$ is given by

$$\sqrt{\sum_{i=1}^{n} \Big(\max\{|(\nabla f(x))_i + \lambda_i u_i| : u_i \in \partial|x_i|\} \Big)^2}. \tag{27}$$

In the course of running the APG algorithm, one can actually get a good upper bound on $\delta(x)$ without incurring extra computational cost as follows. At the k-th iteration, let $g^k = y^k - \nabla f(y^k)/L$. From the subproblem (14), we can observe that

$$\partial(\lambda^T|x^{k+1}|) \ni L(g^k - x^{k+1}) = L(y^k - x^{k+1}) - \nabla f(y^k).$$

Thus we have $L(y^k - x^{k+1}) + \nabla f(x^{k+1}) - \nabla f(y^k) \in \partial(f(x^{k+1}) + \lambda^T|x^{k+1}|)$. Since ∇f is Lipschitz continuous with Lipschitz constant L, we have the following upper bound for $\delta(x^{k+1})$ with

$$\delta(x^{k+1}) \le \|L(y^k - x^{k+1}) + \nabla f(x^{k+1}) - \nabla f(y^k)\| \le 2L\|y^k - x^{k+1}\|.$$

Note that we choose not to evaluate $\nabla f(x^{k+1})$ when computing an upper bound for $\delta(x^{k+1})$ since evaluating the gradient is costly. From the above, we derive the following stopping condition for the APG algorithm:

$$\frac{2L\|y^{k-1} - x^k\|}{\max\{1, \|x^k\|\}} \le \text{Tol}, \tag{28}$$

where Tol is a moderately small tolerance. In addition, we stop the APG algorithm when

$$\frac{|\|AW^T x^k - b\|_D - \|AW^T x^{k-1} - b\|_D|}{\|AW^T x^k - b\|_D} \leq \text{Tol}. \tag{29}$$

For image deblurring problems, we use $0.2 \times$ Tol instead of Tol for (29) in order to prevent our algorithm from stopping prematurely. The scaling constant 0.2 was found after some experimentation. We also stop the APG algorithm when the relative error of the iterates satisfies the following condition:

$$\frac{\|x^k - x^{k-1}\|}{\max\{1, \|x^k\|\}} \leq \text{Tol}. \tag{30}$$

Similar stopping conditions for the unconstrained convex minimization problem (6) can be obtained simply by replacing $f(x)$ with $F(x_1, x_2)$.

Throughout the experiments, we choose Tol $= 5 \times 10^{-4}$ for the stopping conditions (28)-(30). The matrices W and W_1 are set to the tight frame generated by piecewise linear B-spline framelets. For W_2, we set it to be the tight frame generated by a local discrete cosine transform.

Our experiments are conducted for the models (5) and (6) where we set $\alpha = 0.1e^T\lambda/m^2$ and $\alpha_i = 0.1e^T\lambda_i/m_i^2$, respectively. The value for α is chosen so that the term $\frac{\alpha}{2}\|x\|^2$ does not dominate the l_1-term $\lambda^T|x|$. (Similar reason applies to the choice of the parameter values for α_1, α_2 in (6).) We set the parameters $\kappa = \kappa_1 = \kappa_2 = 1$ throughout. Our numerical experience showed that this choice produces the best result in terms of the number of APG iterations taken and the PSNR values obtained. In fact this choice is consistent with the results in Propositions 3.3 and 3.4. Indeed, for the image inpainting problem, $\kappa = 1$ would produce the smallest Lipschitz constant $L = 1 + \alpha$ for ∇f in the problem (5). A similar statement also holds for the image inpainting problem involving the problem (6). It is also interesting to note that the parameter κ is also chosen to be one in [6, 7, 8, 9, 13, 14, 15, 16, 17], although the derivations of the algorithms are different from those we have presented here. In [8, 14, 15, 16, 17], each algorithm was developed by using both the properties of frames and the underlying characteristics of image restoration problems. It was then proven that the limit of the minimizers to a sequence of modified problems is a minimizer of (2) with $\kappa = 1$. In this paper, we formulate the image restoration problem in terms of the minimization problems (5) or (6) as motivated by results of [6, 7, 8, 9, 13, 14, 15, 16, 17], and develop a fast algorithm to find the minimizer.

Finally, note that for the numerical results we obtain in the next few subsections, we have also conducted the same set of numerical experiments on (5) (respectively (6)) under the same set of parameter values for κ (κ_i) and λ (λ_i) but with α (α_i) set to 0. We observe the numerical results are almost the same and thus we do not report those results here.

4.2 Image Inpainting

In this section, we apply the APG algorithm to the image inpainting problem, which refers to the problem of filling-in the missing part in a damaged image. Let Ω be the region of known pixels of an observed image. Our aim is to recover the original image u from the observed image b, which is given by

$$b = P_\Omega(u + \sigma\xi), \tag{31}$$

where P_Ω is projection, or more precisely, a diagonal matrix with diagonals 1 if the corresponding pixels are known, or 0 otherwise, the elements of ξ are i.i.d. standard Gaussian random variables,

and σ is the noise level contained in the image. The image inpainting problem arises, for examples, in removing scratches in photos, in restoring ancient drawings, and in filling in the missing pixels of images transmitted through a noisy channel. In this problem, we need to extract information such as edges and textures from the observed data to fill in the missing part so that shapes and patterns are consistent with our human vision.

We propose to recover the true image u by solving either (5) or (6), with

$$A = P_\Omega, \quad D = I. \tag{32}$$

To measure the quality of the restored image, we use the PSNR value defined by

$$\text{PSNR} = -20 \log_{10} \frac{\|u - \tilde{u}\|}{255N}$$

where u and $\tilde{u}$ are the original and restored images, respectively, and N is the total number of pixels in u.

In Table 1, we report the numerical performance of the APG algorithm applied to (5) and (6) for the image inpainting problem. In the table, we report for each image, the number of iterations taken by the APG algorithm; the PSNR value of the restored image; and the CPU time (in seconds). As indicated in the table, the APG algorithm took no more than 27 iterations and 25 seconds to solve the model (5) for all the images. For the model (6), the APG algorithm took no more than 35 iterations and 40 seconds to solve all the problems. Though the second model (6) is more expensive to solve, the PSNR values of the restored images are consistently better than those obtained from the first model (5) for the last five images which have obvious cartoon and texture structures. As mentioned in the Introduction, for an image that has a two-layer structure (the cartoon and texture parts with very different characteristics), it is more appropriate to represent the layers by two different tight frames to capture their respective characteristics. For the last five images we considered in Table 1, they have obvious two-layer structures. Hence, it is not surprising that when the two different parts are represented by appropriate tight frame systems, the restored images can have better qualities. The restored `barbara512` images are shown in Figure 1.

In this paper, we will mainly compare our numerical results with those obtained by various algorithms in [12]. Although [12] focuses on the split Bregman iterations for image restoration via the analysis based approach, it also provides a rather complete comparison with other frame-based image restoration methods in the literature. For brevity, we summarize the comparison in Table 2. As we can see from the table, the APG algorithm solved the inpainting problem for the image `peppers256` in 22 iterations via the model (5) and achieved a PSNR value of 33.69. In contrast, the proximal forward-backward splitting (PFBS) algorithm took 329 iterations to solve the same problem in [12] and achieved a slightly better PSNR value of 33.82. Since the computational cost of each iteration of the APG algorithm is almost the same as that for the PFBS algorithm, we see that for this example, the APG algorithm is much more efficient than the PFBS algorithm for image inpainting problems based on the balanced approach. Similarly, we can also see that the APG algorithm is more efficient than the split Bregman iteration applied to the analysis based problem (4) for the inpainting problem involving the cartoon-texture image `barbara512`. For this case, the former took 31 iterations to achieve a PSNR value of 33.82, whereas the latter took 67 iterations and obtained a PSNR value of 33.77.

Table 1: Numerical results for the APG algorithm in solving (5) and (6) arising from image inpainting without noise (i.e., $\sigma = 0$ in (1)).

inpainting	one system			two systems		
$\sigma = 0$	$\lambda = 0.03$			$\lambda_1 = \lambda_2 = 0.01$		
	iter	psnr	time	iter	psnr	time
peppers256	22	33.69	3.36	29	33.66	6.45
goldhill256	24	32.21	3.75	32	32.09	7.14
boat256	23	30.99	3.61	29	30.87	6.53
camera256	23	30.13	3.56	29	30.44	6.45
bridge256	26	31.30	4.12	33	31.08	7.41
bowl256	23	34.38	3.56	35	36.02	7.60
barbara512	27	31.33	22.48	31	33.82	34.10
baboon512	26	29.12	22.05	32	29.10	35.64
fingerprint512	25	26.51	21.13	34	28.00	37.67
zebra512	25	28.47	20.54	33	29.32	35.96

Table 2: Comparison of the APG algorithm in solving (5) with other algorithms. The numbers are in the format: iteration count/PSNR value/CPU time in seconds.

image	problem type	APG on (5)	[12]		
			split Bregman on (4)	linearized Bregman on (3)	proximal forward backward on (2)
`pepper256`	inpainting	22/33.69/3.4	51/33.86/10.1		329/33.82/46.0
`barbara512`	inpainting cartoon texture	31/33.82/34.1	67/33.77/71.5		
`barbara512`	denoising cartoon texture	29/28.39/31.5	55/29.01/71.6		
`goldhill256`	deblur `average`, `9` $\sigma = 3$	27/26.41/5.0	19/26.40/14.6	11/26.21/7.1	171/26.21/107.3
`boat256`	deblur `disk`, `4` $\sigma = 3$	28/25.46/5.8	18/25.30/13.8	12/25.32/7.7	155/25.00/99.9

Next, we consider the problem of denoising. That is, $P_\Omega = I$, but with $\sigma = 20$ in (31). Note that the pixel values in the image u are in the interval $[0, 255]$. The numerical results obtained by the APG algorithm are presented in Table 3. From the table, we see that the APG algorithm took no more than 17 iterations and 15 seconds to solve the problems (5) for all the images. For the model (6), it took no more than 32 iterations and 35 seconds to solve all the problems. Observe that the number of iterations taken to solve the problem (5) is consistently less than that for the

problem (6). This is not surprising since the Lipschitz constant for ∇f in (5) is smaller than that for the corresponding function in (6).

Table 3: Numerical results for the APG algorithm in solving (5) and (6) arising from image denoising with $\sigma = 20$.

denoising	one system			two systems		
	$\lambda = 0.11$			$\lambda_1 = \lambda_2 = 0.08$		
	iter	psnr	time	iter	psnr	time
bowl256	16	28.77	2.50	29	29.13	6.31
barbara512	17	27.38	14.24	29	28.39	31.49
baboon512	17	25.81	14.47	29	25.40	31.50
fingerprint512	17	27.76	14.23	32	27.51	34.84
zebra512	17	28.24	14.18	29	28.09	31.34

From Table 2, we see that the APG algorithm is more efficient than the split Bregman iteration developed in [12] for the denoising problem involving the cartoon-texture image `barbara512`. In this case, the former took 29 iterations to achieve a PSNR value of 28.39 whereas the latter took 55 iterations and obtained a PSNR value of 29.01.

Our next experiment is to consider the image decomposition problem, i.e., to decompose a given image into its cartoon and texture parts. In this case, $P_\Omega = I$ and $\sigma = 0$. For this problem, it is meaningful to consider only the model (6). Table 4 reports the numerical results for the APG algorithm in solving 5 image decomposition problems. The restored `barbara512` images are shown in Figure 2.

To summarize, we conclude that comparing with the numerical results of [8, 12], the APG algorithm is much faster than the PFBS algorithm in obtaining comparable quality in the restored images for frame-based image inpainting by using the balanced approach. It is comparable to the split Bregman iteration for frame-based image inpainting by using the analysis based approach. The large number of images we used in our simulations also shows that the performance of the APG algorithm is independent of the images and it is robust.

Table 4: Numerical results for the APG algorithm in solving (6) arising from image decomposition.

decomposition	one system			two systems		
$\sigma = 0$				$\lambda_1 = 0.007, \lambda_2 = 0.01$		
	iter	psnr	time	iter	psnr	time
peppers256				28	41.64	6.20
goldhill256				28	39.99	6.37
boat256				27	40.81	6.12
camera256				28	41.96	6.21
bridge256				27	39.58	6.25
bowl256				28	43.46	6.21
barbara512				29	41.88	32.17

Table 4: Numerical results for the APG algorithm in solving (6) arising from image decomposition.

decomposition	one system			two systems		
$\sigma = 0$				$\lambda_1 = 0.007, \lambda_2 = 0.01$		
	iter	psnr	time	iter	psnr	time
baboon512				27	39.39	30.84
fingerprint512				27	41.06	30.41
zebra512				28	42.95	30.75

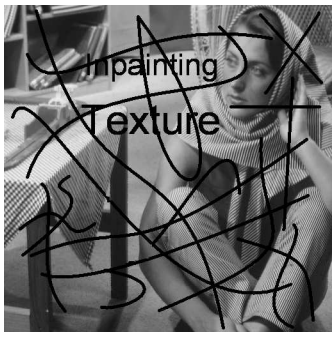
4.3 Image Deblurring

Images can be blurred for various reasons which can significantly degrade the visual quality of images. For example, it can be blurred by the motions. Motion blurring caused by camera shake has been one of the prime causes of poor image quality in digital imaging, especially when using telephoto lenses or long shuttle speeds. In many imaging applications, there is simply not enough light to produce a clear image by using a short shutter speed. As a result, the image will appear blurry due to the relative motion between the camera and the scene. In this section, we use a frame-based deblurring method via the balanced approach to restore images degraded by some convolution operators. The aim is to find the underlying image $u = W^T x$ from its blurred observation b in (1), where A is a convolution operator, by applying the APG algorithm to solve the problem (5) with D being a positive definite matrix depending on A. As mentioned in Introduction, real images usually have two layers. Hence we also consider the problem (6). Note that the observed image in (1) is the transformed image Au with noise corruption given by $\eta = \sigma\xi$, where the elements of ξ are i.i.d. standard Gaussian random variables and σ^2 is the noise variance.

Since A is usually ill-conditioned, we choose a symmetric positive definite matrix D to approximate $(AA^T)^{-1}$ so that $A^T DA$ may have a good condition number. The choice of D can be done as follows. The blurring operator A is usually a convolution operator, and it is a Toeplitz-like or block-Toeplitz-like matrix with a suitable boundary condition. Hence A can be efficiently approximated by a circulant matrix or a fast transform based matrix C [18, 32]. In this paper, we use convolution matrices with circular or Neumann boundary conditions to approximate A, and $D = (CC^T)^{-1}$ is usually a good approximation for $(AA^T)^{-1}$. In order for the approximation to be numerically stable and robust against noise, we choose $D = (CC^T + \theta I)^{-1}$, where θ is a small positive number. Note that, for any $y \in \Re^\ell$, $(CC^T + \theta I)^{-1}y$ can be computed efficiently by fast Fourier transform or Discrete cosine transform [32]. For our numerical experiments, we consider 4 blurring kernels: (a) 7×7 disk kernel; (b) 15×30 motion kernel; (c) 15×15 Gaussian kernel with standard deviation 2; and (d) 9×9 average kernel.

Figure 1: Image inpainting results ($\sigma = 0$).

(a) original: 512×512



(b) observed image

(c) inpainted from (5): one system

(d) inpainted from (6): two systems

Figure 2: Image decomposition results.

(a) original: 512×512

(b) decomposition via (6)

(c) cartoon part

(d) texture part

Table 5: Numerical results for the APG algorithm in solving (5) and (6) arising from image deblurring with noise level $\sigma = 3$.

deblurring		one system				two systems			
		$\lambda = 0.003$				$\lambda_1 = \lambda_2 = 0.003$			
	blur	θ	iter	psnr	time	θ	iter	psnr	time
peppers256	disk,3	0.40	28	28.17	5.63	0.24	34	27.57	8.70
peppers256	motion,15,30	0.50	28	27.28	5.49	0.30	28	26.94	6.96
peppers256	gaussian,15,2	0.30	22	25.79	4.18	0.18	21	25.71	5.26
peppers256	average,9	0.35	28	27.77	5.27	0.21	28	27.28	6.78
goldhill256	disk,3	0.40	27	27.21	5.42	0.24	36	27.06	9.11
goldhill256	motion,15,30	0.50	27	26.86	5.29	0.30	29	26.72	7.24
goldhill256	gaussian,15,2	0.30	22	26.48	4.23	0.18	21	26.47	5.23
goldhill256	average,9	0.35	27	26.41	5.03	0.21	31	26.32	7.51
boat256	disk,3	0.40	28	26.43	5.65	0.24	36	25.92	9.19
boat256	motion,15,30	0.50	27	25.90	5.30	0.30	28	25.56	7.01
boat256	gaussian,15,2	0.30	23	25.15	4.40	0.18	22	25.04	5.49
boat256	average,9	0.35	27	25.28	5.10	0.21	30	25.03	7.33
camera256	disk,3	0.40	28	26.98	5.56	0.24	35	26.37	8.83
camera256	motion,15,30	0.50	28	26.44	5.54	0.30	29	26.10	7.22
camera256	gaussian,15,2	0.30	22	25.08	4.22	0.18	22	24.95	5.49
camera256	average,9	0.35	28	25.39	5.24	0.21	31	25.07	7.48
bridge256	disk,3	0.40	28	25.41	5.65	0.24	39	25.17	9.93
bridge256	motion,15,30	0.50	28	25.02	5.52	0.30	32	24.80	7.94
bridge256	gaussian,15,2	0.30	24	24.42	4.69	0.18	22	24.36	5.51
bridge256	average,9	0.35	28	24.32	5.28	0.21	31	24.19	7.60
bowl256	disk,3	0.40	26	29.04	5.15	0.24	34	29.10	8.53
bowl256	motion,15,30	0.50	26	29.31	5.00	0.30	29	29.51	7.15
bowl256	gaussian,15,2	0.30	22	28.49	4.22	0.18	21	28.60	5.19
bowl256	average,9	0.35	25	28.80	4.61	0.21	31	28.97	7.46
barbara512	disk,3	0.40	29	25.34	29.38	0.24	36	25.80	44.32
barbara512	motion,15,30	0.50	28	25.01	27.97	0.30	34	25.57	41.07
barbara512	gaussian,15,2	0.30	22	24.21	21.67	0.18	20	24.20	24.08
barbara512	average,9	0.35	26	24.12	24.92	0.21	29	24.17	34.12
baboon512	disk,3	0.40	29	23.47	29.61	0.24	39	23.43	49.16
baboon512	motion,15,30	0.50	28	23.44	28.13	0.30	37	23.33	45.84
baboon512	gaussian,15,2	0.30	24	22.19	23.61	0.18	22	22.15	27.35
baboon512	average,9	0.35	28	22.33	26.94	0.21	33	22.24	40.32
fingerprint512	disk,3	0.40	29	27.40	29.96	0.24	37	27.81	47.27
fingerprint512	motion,15,30	0.50	30	25.49	30.14	0.30	32	25.55	40.02
fingerprint512	gaussian,15,2	0.30	26	27.30	26.11	0.18	24	27.41	30.84
fingerprint512	average,9	0.35	31	24.52	30.16	0.21	30	24.69	35.91
zebra512	disk,3	0.40	29	27.26	29.99	0.24	39	27.41	48.15
zebra512	motion,15,30	0.50	30	25.99	30.53	0.30	36	26.22	43.38
zebra512	gaussian,15,2	0.30	25	26.00	25.10	0.18	27	25.99	32.66
zebra512	average,9	0.35	29	24.72	28.57	0.21	30	24.75	35.73

Table 6: Numerical results for the APG algorithm in solving (5) and (6) arising from image deblurring with noise level $\sigma = 5$.

deblurring		one system				two systems			
		$\lambda = 0.003$				$\lambda_1 = \lambda_2 = 0.003$			
	blur	θ	iter	psnr	time	θ	iter	psnr	time
peppers256	disk,3	1.00	28	26.68	5.56	0.60	26	26.29	6.51
peppers256	motion,15,30	1.00	31	25.63	6.05	0.60	30	25.43	7.46
peppers256	gaussian,15,2	1.00	25	25.23	4.77	0.60	22	25.10	5.44
peppers256	average,9	1.00	32	26.34	5.96	0.60	28	25.81	6.80
goldhill256	disk,3	1.00	27	26.57	5.32	0.60	23	26.53	5.84
goldhill256	motion,15,30	1.00	31	25.76	6.01	0.60	29	25.73	7.21
goldhill256	gaussian,15,2	1.00	25	26.12	4.79	0.60	22	26.08	5.41
goldhill256	average,9	1.00	29	25.61	5.33	0.60	27	25.55	6.48
boat256	disk,3	1.00	27	25.43	5.34	0.60	24	25.14	6.06
boat256	motion,15,30	1.00	31	24.69	5.97	0.60	29	24.47	7.19
boat256	gaussian,15,2	1.00	25	24.59	4.72	0.60	22	24.46	5.46
boat256	average,9	1.00	30	24.40	5.61	0.60	27	24.20	6.52
camera256	disk,3	1.00	28	25.66	5.51	0.60	29	25.27	7.36
camera256	motion,15,30	1.00	32	25.03	6.21	0.60	30	24.76	7.42
camera256	gaussian,15,2	1.00	25	24.56	4.69	0.60	23	24.39	5.76
camera256	average,9	1.00	30	24.30	5.47	0.60	27	24.03	6.56
bridge256	disk,3	1.00	29	24.63	5.76	0.60	26	24.46	6.59
bridge256	motion,15,30	1.00	31	23.95	6.02	0.60	31	23.83	7.72
bridge256	gaussian,15,2	1.00	26	23.91	4.95	0.60	23	23.83	5.72
bridge256	average,9	1.00	32	23.49	5.90	0.60	28	23.34	6.75
bowl256	disk,3	1.00	26	28.29	5.07	0.60	24	28.41	6.02
bowl256	motion,15,30	1.00	29	28.01	5.60	0.60	30	28.35	7.43
bowl256	gaussian,15,2	1.00	24	27.64	4.49	0.60	22	27.69	5.53
bowl256	average,9	1.00	28	28.16	5.13	0.60	25	28.36	6.02
barbara512	disk,3	1.00	26	24.47	26.14	0.60	32	24.80	38.95
barbara512	motion,15,30	1.00	30	24.14	29.62	0.60	30	24.41	36.50
barbara512	gaussian,15,2	1.00	23	24.04	22.36	0.60	20	24.03	24.23
barbara512	average,9	1.00	28	23.77	26.49	0.60	25	23.81	29.46
baboon512	disk,3	1.00	28	22.51	28.24	0.60	37	22.46	45.97
baboon512	motion,15,30	1.00	31	22.48	30.89	0.60	31	22.38	37.65
baboon512	gaussian,15,2	1.00	25	21.79	24.67	0.60	22	21.74	26.65
baboon512	average,9	1.00	30	21.59	28.53	0.60	27	21.50	31.82
fingerprint512	disk,3	1.00	31	26.75	32.17	0.60	26	27.16	32.49
fingerprint512	motion,15,30	1.00	34	24.17	34.03	0.60	39	24.19	47.13
fingerprint512	gaussian,15,2	1.00	27	26.48	26.96	0.60	25	26.42	30.71
fingerprint512	average,9	1.00	34	23.40	32.82	0.60	33	23.49	39.22
zebra512	disk,3	1.00	29	26.02	29.32	0.60	28	26.01	34.19
zebra512	motion,15,30	1.00	33	24.62	32.84	0.60	35	24.73	42.14
zebra512	gaussian,15,2	1.00	28	25.16	27.38	0.60	26	25.07	31.86
zebra512	average,9	1.00	32	23.65	30.64	0.60	28	23.58	33.88

Figure 3: Image deblurring results when $\sigma = 5$ and motion blurring kernel is used.

(a) original: 512×512

(b) noisy blurred image

(c) deblurred via (5)

(d) deblurred via (6)

Tables 5 and 6 report the number of iterations, the PSNR values, and the CPU times for the APG algorithm to solve image deblurring problems via (5) and (6), corresponding to the cases $\sigma = 3$ and $\sigma = 5$, respectively. As shown in the tables, it took the AGP algorithm no more than 40 iterations to solve all the image deblurring problems via (5) or (6), and all the 512×512 problems are solved in less than 50 seconds. The restored `barbara512` images are shown in Figure 3.

Next, we compare the performance of the APG algorithm with the split Bregman algorithm and PFBS algorithm in [12] on some image deblurring problems. In Table 2, for the image `goldhill256`, which is blurred by the 9×9 average kernel and contaminated with noise level $\sigma = 3$, the APG algorithm is able to solve the model (5) in 27 iterations and achieved a PSNR value of 26.41. This result is competitive to that obtained by the split Bregman algorithm in [12] applied to the analysis based problem (4), where it took 19 iterations to achieve a PSNR value of 26.40. The linearized Bregman algorithm applied to the synthesis based problem (3) took 11 iterations to achieve a PSNR value of 26.21. For the problem (2) in the balanced approach, the PFBS algorithm took 171 iterations to achieve a PSNR value of 26.21. Clearly, the APG algorithm is much more efficient than the PFBS algorithm for deblurring problems based on the balanced approach.

For the image `boat256`, which is blurred by the disk kernel of radius 4, and contaminated with noise level $\sigma = 3$ in Table 2, the APG algorithm took 28 iterations to solve the problem (5) and achieved a PSNR value of 25.46. Our APG algorithm is again competitive to the split Bregman and linearized Bregman algorithms. In [12], the split Bregman and linearized Bregman algorithms

took 18 and 12 iterations to solve the analysis and synthesis based problems, (4) and (3), and achieved the PSNR values of 25.30 and 25.32, respectively. For the balanced model (2), the PFBS algorithm took 155 iterations to solve the problem and obtained a PSNR value of 25.00.

5 Conclusions

In this paper we have considered ℓ_1-regularized linear least squares problems arising from frame-based image restoration via the balanced approach. We have proposed an accelerated proximal gradient algorithm with continuation strategy for solving this convex nonsmooth minimization problem, and give a natural and robust stopping condition based on the sub-gradient of the objective function without incurring extra computational cost. This leads to a set of frame-based image restoration algorithms that are applied to solve large-scale instances in several image restoration problems, such as image deblurring, inpainting, denoising, and cartoon-texture image decomposition. The numerical simulation results show that our algorithms are fast, efficient and robust for frame-based image restoration via the balanced approach. In fact, all the 512×512 images we considered in image inpainting, denoising, deblurring, and cartoon-texture decomposition are successfully restored in less than 50 seconds on a modest PC.

References

[1] A. Beck and M. Teboulle, *Fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci., 2 (2009), pp. 183–202.

[2] S. Becker, J. Bobin, and E. J. Candés, *NESTA: a fast and accurate first-order method for sparse recovery*, SIAM J. Imaging Sci., 4 (2011), pp. 1–39.

[3] S. Becker, E. J. Candès, and M. Grant, *Templates for convex cone problems with applications to sparse signal recovery*, Preprint, 2010.

[4] J. Biemond, R. Lagendijk, and R. M. Mersereau, *Iterative methods for image deblurring*, Proc. of IEEE., 78 (1990), pp. 856–883.

[5] J.-F. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982.

[6] J.-F. Cai, R. H. Chan, L. Shen, and Z. Shen, *Restoration of chopped and nodded images by framelets*, SIAM J. Sci. Comput., 30 (2008), pp. 1205–1227.

[7] J.-F. Cai, R. H. Chan, L. Shen, and Z. Shen, *Convergence analysis of tight framelet approach for missing data recovery*, Adv. Comput. Math., 31 (2009), pp. 87–113.

[8] J.-F. Cai, R. H. Chan, and Z. Shen, *A framelet-based image inpainting algorithm*, Appl. Comput. Harmon. Anal., 24 (2008), pp. 131–149.

[9] J.-F. Cai, R. H. Chan, and Z. Shen, *Simultaneous cartoon and texture inpainting*, Inverse Probl. Imaging, 4 (2010), pp. 379–395.

[10] J.-F. Cai, S. Osher, and Z. Shen, *Linearized Bregman Iterations for Compressed Sensing*, Math. Comp., 78 (2009), pp. 1515–1536.

[11] J.-F. Cai, S. Osher, and Z. Shen, *Linearized Bregman Iterations for Frame-Based Image Deblurring*, SIAM J. Imaging Sci., 2 (2009), pp. 226–252.

[12] J.-F. Cai, S. Osher, and Z. Shen, *Split Bregman methods and frame based image restoration*, Multiscale Model. Simul., 8 (2009), pp. 337–369.

[13] J.-F. Cai and Z. Shen, *Framelet Based Deconvolution*, J. Computational Mathematics, 28 (2010), pp. 289–308.

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

[15] R. H. Chan, T. F. Chan, L. Shen, and Z. Shen, *Wavelet algorithms for high-resolution image reconstruction*, SIAM J. Sci. Comput., 24 (2003), pp. 1408–1432.

[16] R. H. Chan, S. D. Riemenschneider, L. Shen, and Z. Shen, *Tight frame: an efficient way for high-resolution image reconstruction*, Appl. Comput. Harmon. Anal., 17 (2004), pp. 91–115.

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

[18] R. H.-F. Chan and X.-Q. Jin, *An Introduction to Iterative Toeplitz Solvers*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2007.

[19] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Rev., 43 (2001), pp. 129–159.

[20] I. Daubechies, M. De Friese, and C. De Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Comm. on Pure and Applied Math., 57 (2004), pp. 1413–1457.

[21] I. Daubechies, B. Han, A. Ron, and Z. Shen, *Framelets: MRA-based constructions of wavelet frames*, Appl. Comput. Harmon. Anal., 14 (2003), pp. 1–46.

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

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

[24] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, *Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)*, Appl. Comput. Harmon. Anal., 19 (2005), pp. 340–358.

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

[26] M. Fadili, J.-L. Starck, and F. Murtagh, *Inpainting and zooming using sparse representations*, The Computer Journal, 52 (2009), pp. 64–79.

[27] M. Figueiredo and R. Nowak, *An EM algorithm for wavelet-based image restoration*, IEEE Trans. Image Proc., 12 (2003), pp. 906–916.

[28] M. Figueiredo and R. Nowak, *A bound optimization approach to wavelet-based image deconvolution*, IEEE Intern. Conf. on Image Processing-ICIP'05, 2005.

[29] M. Friedlander and P. Tseng, *Exact regularization of convex programs*, SIAM J. Optim., 18 (2007), pp. 1326–1350.

[30] T. Goldstein and S. Osher, *The split Bregman algorithm for L1 regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.

[31] E. T. Hale, W. Yin, and Y. Zhang, *Fixed-point continuation for ℓ_1-minimization: Methodology and convergence*, SIAM J. Optim., 19 (2008), pp. 1107–1130.

[32] X.-Q. Jin, *Developments and applications of block Toeplitz iterative solvers*, vol. 2 of Combinatorics and Computer Science, Kluwer Academic Publishers Group, Dordrecht, The Netherlands, 2002.

[33] Z. Lu, *Gradient based method for cone programming with application to large-scale compressed sensing*, Technical Report, 2008.

[34] A. Nemirovski and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*, Wiley, New York, 1983.

[35] Y. Nesterov, *A method of solving a convex programming problem with convergence rate $O(1/k^2)$*, Soviet Mathematics Doklady, 27 (1983), pp. 372–376.

[36] Y. Nesterov, *On an approach to the construction of optimal methods of minimization of smooth convex functions*, Èkonom. i. Mat. Metody, 24 (1988), pp. 509–517.

[37] Y. Nesterov, *Smooth minimization of nonsmooth functions*, Math. Prog., 103 (2005), pp. 127–152.

[38] S. Osher, Y. Mao, B. Dong, and W. Yin, *Fast Linearized Bregman Iteration for Compressed Sensing and Sparse Denoising*, Communications in Mathematical Sciences, 8 (2010), pp. 93–111.

[39] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Springer-Verlag, New York, 1998.

[40] A. Ron and Z. Shen, *Affine systems in $L_2(\Re^d)$: the analysis of the analysis operator*, J. Funct. Anal., 148 (1997), pp. 408–447.

[41] K.-C. Toh and S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems*, Pacific J. Optim., 6 (2010), pp. 615–640.

[42] P. Tseng, *On accelerated proximal gradient methods for convex-concave optimization*, 2008, submitted to SIAM J. Optim.

[43] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168.

[44] W. Yin, *Analysis and generalizations of the linearized Bregman method*, CAAM Technical Report TR09-02, Rice University, 2009.

[45] X. Zhang, M. Burger, X. Bresson, and S. Osher, *Bregmanized Nonlocal Regularization for Deconvolution and Sparse Reconstruction*, CAM Report (09-03), UCLA, 2009.