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

Zuowei Shen * Kim-Chuan Toh † Sangwoon Yun ‡

December 3, 2009

Abstract

Frame based image restorations by using balanced approach have been developed over the last decade in [5, 6, 7, 8, 14, 15, 16, 17, 18]. The algorithm developed in these papers can be viewed as an acceleration of the proximal forward-backward splitting algorithm (see e.g.[5, 6, 7, 8, 14]). Accelerated proximal gradient algorithms studied in [1, 38, 44] have been demonstrated to be efficient in solving various regularized convex optimization problems arising in compressive sensing, machine learning, and control. In this paper, we apply the accelerated proximal gradient algorithm to the balanced approach in frame based image restoration which is formulated as an ℓ_1-regularized linear least squares problem. This algorithm terminates in $O(1/\sqrt{\epsilon})$ iterations with an ϵ-optimal solution, and it leads to a set of new frame based image restoration algorithms that can universally handle several image restoration problems, such as image deblurring, denoising, inpainting, and cartoon-texture decomposition. The numerical results suggest that our algorithm is efficient and robust in solving large-scale image restoration problems. Our algorithms are able to successfully restore 512×512 images in image deblurring, denoising, inpainting and cartoon-texture decomposition in less than 50 seconds on a modest PC. We also compare the numerical performance of our proposed algorithms applied to image 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, 2 Science Drive 2, Singapore 117543 (matzuows@nus.edu.sg).

†Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543 (mattohkc@nus.edu.sg); and Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576.

‡Korea Institute for Advanced Study, 207-43 Cheongnyangni 2-dong, Dongdaemun-gu, Seoul 130-722, Korea(yswcs@kias.re.kr).

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 in image denoising.

Our approach for the 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 [42]. We refer interested readers to [21, 42] and the references therein for the general wavelet frame theory and its corresponding constructions. The details in the construction of W from a given wavelet tight frame system can be found in, for example, [5, 6, 7, 8, 14, 15, 16, 17, 18].

Since tight frame systems are redundant systems, the mapping from the image u to its coefficient 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 [24, 25]. In that 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, 26, 27, 28, 29]. In that 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 [16, 17] for high resolution image reconstruction. It was further developed for various image restorations in [5, 6, 7, 8, 14, 15, 18]. Although the synthesis based, analysis based and the 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.

Next, we give the exact definitions of the above three approaches. Before that, we set up some notation. For any $x \in \Re^n$, $\|x\|_p = \left(\sum_{j=1}^n |x_j|^p\right)^{1/p}$, $1 \leq p < \infty$. For simplicity, we write $\|x\| = \|x\|_2$ and $|x|$ denotes the vector obtained from x by taking the absolute values of its components. Let $\|x\|_D$ 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 matrices A, $\|A\|_2 = \sqrt{\lambda_{\max}(A^T A)}$. The identity matrix is denoted by I and the matrix of zero entries is denoted by 0.

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 positive weight vector, $|x|$ denotes the ℓ_1 norm of the vector x, and D is a given symmetric positive definite matrix. When $\kappa=0$, the problem (2) is reduced to a 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 an analysis based approach. To see this, we note that the distance $\|(I-WW^T)x\|$ 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

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

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 a balanced approach.

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 in this case, $WW^T=I$. However, for redundant tight frame system W, the analysis based, the synthesis based and the balanced approaches cannot be derived from one another. In fact, it was observed in, for examples, [9, 24] that there is a gap between the analysis based and the synthesis based approaches. Both of them have their own favorable data sets and applications. In general, it is hard to draw definitive conclusions as to which approach is better, without specifying the applications and data sets. We further note that the ℓ_1-minimization problem arising from compressed sensing is akin to the synthesis based approach by nature. On the other hand, the TV-norm minimization problem in imaging restoration is, in many cases, an analysis based approach. For frame based image restorations, numerical simulation results in [12, 13] show that the analysis based approach tends to generate smoother images. This is because the coefficient Wu is quite often linked to the smoothness of the underlying image. However, the synthesis based approach tends to explore more on the sparse representation of the underlying solution in terms of the given frame system by utilizing the redundance. Our balanced approach bridge the analysis based and synthesis based approaches in image restorations and it provides an additional approach in the rich literature of frame based image restoration as shown in [5, 6, 7, 14, 15, 16, 17, 18].

Recently, the linearized Bregman iteration was proposed for solving the ℓ_1-minimization problems in compressed sensing by [11, 39, 45] and the nuclear norm minimization in matrix completion by [4]. The linearized Bregman iteration was then used to develop a fast algorithm for the synthesis based approach for frame based image deblurring in [12]. Furthermore, the split Bregman iteration proposed in [30] was shown to be powerful in [30, 46] when it is applied to various PDE based image restoration approaches, e.g., ROF model 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 restorations in [13]. While the balanced approach in frame based image restoration gives satisfactory simulation results, as shown in [5, 6, 7, 14, 15, 16, 17, 18], when solved by a variant of the proximal forward-backward splitting algorithm. But as shown in [5, 6, 7, 14, 15, 16, 17, 18], the numerical convergence speed achieved is not as fast as the synthesis based approach by using the linearized Bregman iteration, or the analysis based approach by using the split Bregman

iteration. The main goal of this paper is to develop fast algorithms for the balanced approach in frame based image restorations whose convergence speeds are competitive to those of the linearized Bregman iteration for the synthesis based approach and the split Bregman iteration for the analysis based approach. With this, one can free to chose either synthesis based approach, analysis based approach or balanced approach according to his priorities and applications.

Our accelerated algorithms are based on several variants of accelerated proximal gradient algorithms that were studied by Nesterov, Nemirovski, Beck and Teboulle, and others; see [1, 34, 35, 36, 38, 44] and references therein, and that were proposed to solve ℓ_1-regularized linear least squares problems arising in signal/image processing [1], compressed sensing [2, 33] and nuclear norm regularized linear squares problems [43]. These accelerated proximal gradient algorithms have an attractive iteration complexity of $O(1/\sqrt{\epsilon})$ for achieving ϵ-optimality; see Section 2. Also these accelerated proximal gradient algorithms are simple and use only the soft-thresholding operator, just like algorithms such as, the linearized Bregman iteration, the split Bregman iteration and the proximal forward-backward splitting algorithm. In this paper, we extend Beck and Teboulle's algorithm [1], which is proposed to solve linear inverse problems, to solve the problems (2), and more general (5), on large-scale problems arising in image restoration.

Next, we extend our algorithm for (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, these 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:

$$\min_{x_1,x_2} \quad \frac{1}{2}\|A(\sum_{i=1}^{2} W_i^T x_i) - b\|_D^2 + \sum_{i=1}^{2} \frac{\kappa_i}{2}\|(I - W_i W_i^T)x_i\|^2 + \sum_{i=1}^{2} \lambda_i^T |x_i|, \tag{5}$$

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

The paper is organized as follows. In Section 2, we introduce some gradient algorithms using proximal regularization and accelerated versions which can be applied to solve (2) and (5). We give the iteration complexity of algorithms. In Section 3, we estimate the Lipschitz constant of the gradient of the linear least squares of problems (2) and (5), especially for the problems arising in image inpainting since the Lipschitz constant is crucial for the convergence speed of our proposed algorithms. In Section 4, we present some preliminary numerical results for solving (2) and (5) on a set of large-scale problems arising from several topics in image restorations, 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 introduce accelerated proximal gradient algorithms for solving (2) and (5). We also give the analysis of their iteration complexity.

In what follows,

$$f(x) = \frac{1}{2}\|AW^T x - b\|_D^2 + \frac{\kappa}{2}\|(I - WW^T)x\|^2 \tag{6}$$

and, for (5),

$$f(x) = f(x_1, x_2) = \frac{1}{2}\|A\Big(\sum_{i=1}^{2} W_i^T x_i\Big) - b\|_D^2 + \sum_{i=1}^{2} \frac{\kappa_i}{2}\|(I - W_i W_i^T)x_i\|^2. \tag{7}$$

Note that the gradient of f is given by

$$\nabla f(x) = WA^T D(AW^T x - b) + \kappa(I - WW^T)x \tag{8}$$

and for (5),

$$\nabla f(x) = (\nabla_{x_1} f(x)^T, \nabla_{x_2} f(x)^T)^T, \tag{9}$$

where

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

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

$$\ell_f(x; y) := f(y) + \langle \nabla f(y),\, x - y\rangle + \lambda^T|x|. \tag{10}$$

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

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \quad \forall\ x,\ y\ \in \Re^m, \tag{11}$$

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

$$f(x) + \lambda^T|x| - \frac{L}{2}\|x - y\|^2 \le \ell_f(x; y) \le f(x) + \lambda^T|x| \quad \forall\ x,\ y\ \in \Re^n. \tag{12}$$

The main step of the accelerated proximal gradient algorithm for solving (2) uses the following subproblem:

$$\min_x \ell_f(x; y) + \frac{L}{2}\|x - y\|^2. \tag{13}$$

Since the objective function of the above subproblem is strictly convex, the solution is unique. By ignoring constant terms in y, (13) can be rewritten as follows:

$$\min_x \frac{L}{2}\|x - g\|^2 + \lambda^T|x|, \tag{14}$$

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

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

$$s_\nu(x) := \mathrm{sgn}(x) \odot \max\{|x| - \nu, 0\} \tag{15}$$

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 (13) and (14).

We now describe formally the accelerated proximal gradient (abbreviated as APG) algorithm for solving (2).

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 the problem (2), it is the popular *iterative shrinkage/thresholding* (IST) algorithms [20, 28, 29, 31] and it is also the proximal forward-backward splitting (PFBS) algorithm developed in [5, 6, 7, 8, 14, 15, 16, 17, 18] for the balanced approach in frame based image restorations. 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 $\epsilon > 0$, these algorithms terminate in $O(L/\epsilon)$ iterations with an ϵ-optimal solution [1, 43]. Hence the sequence $\{x^k\}$ converges slowly. On the other hand, we show in this section that the APG algorithm proposed here gets an ϵ-optimal solution in $O(\sqrt{L/\epsilon})$ iterations. Thus the APG algorithm accelerates the PFBS algorithm used in [5, 6, 7, 8, 14, 15, 16, 17, 18] for the balanced approach in frame based image restorations.

We first prove the following lemma which shows that the optimal solution set of (2) is bounded. In what follows, $\mathcal{X}^*$ denotes the set of optimal solutions.

Lemma 2.1 *For each positive vector λ, the optimal solution set $\mathcal{X}^*$ of (2) is bounded. In addition, for any $x^* \in \mathcal{X}^*$, we have*

$$\|x^*\|_1 \leq \chi \tag{16}$$

where

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

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

Proof. Considering the objective value of (2) 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 (2) at $x = x_{LS}$, we obtain that for any $x^* \in \mathcal{X}^*$, $\lambda_{\min}\|x^*\|_1 \leq f(x^*) + \lambda^T|x^*| \leq \lambda^T|x_{LS}|$. ■

The following theorem gives an upper bound on the number of iterations for the APG algorithm for solving (2) to achieve ϵ-optimality. This theorem can be proved by using [1, Theorem 4.1] or [44, Corollary 2]. We include its proof for completeness. For this, we note the following fact given by [1, Lemma 4.3]:

$$t^k \geq \frac{k+1}{2} \quad \forall k \geq 1. \tag{17}$$

Theorem 2.1 *Let $\{x^k\}$, $\{y^k\}$, $\{t^k\}$ be the sequences generated by APG. 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

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

where χ is defined as in Lemma 2.1.

Proof. Fix any $k \in \{0, 1, ...\}$ and any $x^* \in \mathcal{X}^*$. Let $s^k = s_{\lambda/L}(g^k)$ and $\hat{x} = ((t^k - 1)x^k + x^*)/t^k$. By the definition of s^k and Fermat's rule [41, Theorem 10.1],

$$s^k \in \arg\min_x \{\ell_f(x; y^k) + L\langle s^k - y^k,\, x\rangle\}.$$

Hence

$$\ell_f(s^k; y^k) + L\langle s^k - y^k,\, s^k\rangle \leq \ell_f(\hat{x}; y^k) + L\langle s^k - y^k,\, \hat{x}\rangle. \tag{20}$$

Since

$$\langle s^k - y^k,\, \hat{x}\rangle + \frac{1}{2}\|s^k - y^k\|^2 - \langle s^k - y^k,\, s^k\rangle \;=\; \frac{1}{2}\|\hat{x} - y^k\|^2 - \frac{1}{2}\|\hat{x} - s^k\|^2,$$

adding $\frac{L}{2}\|s^k - y^k\|^2 - L\langle s^k - y^k,\, s^k\rangle$ to both sides of the inequality (20) yields

$$\ell_f(s^k; y^k) + \frac{L}{2}\|s^k - y^k\|^2 \leq \ell_f(\hat{x}; y^k) + \frac{L}{2}\|\hat{x} - y^k\|^2 - \frac{L}{2}\|\hat{x} - s^k\|^2. \tag{21}$$

For notational convenience, let $F(x) = f(x) + \lambda^T|x|$ and $z^k = (1 - t^{k-1})x^{k-1} + t^{k-1}x^k$. The inequality (21) with $s^k = x^{k+1}$ and the first inequality in (12) implies that

$$\begin{aligned}
F(x^{k+1}) \;&\leq\; \ell_f(x^{k+1}; y^k) + \frac{L}{2}\|x^{k+1} - y^k\|^2 \;\leq\; \ell_f(\hat{x}; y^k) + \frac{L}{2}\|\hat{x} - y^k\|^2 - \frac{L}{2}\|\hat{x} - x^{k+1}\|^2 \\
&\leq\; \frac{t^k - 1}{t^k}\ell_f(x^k; y^k) + \frac{1}{t^k}\ell_f(x^*; y^k) \\
&\quad + \frac{L}{2(t^k)^2}\|(t^k - 1)x^k + x^* - t^k y^k\|^2 - \frac{L}{2(t^k)^2}\|(t^k - 1)x^k + x^* - t^k x^{k+1}\|^2 \\
&=\; \frac{t^k - 1}{t^k}\ell_f(x^k; y^k) + \frac{1}{t^k}\ell_f(x^*; y^k) + \frac{L}{2(t^k)^2}\|x^* - z^k\|^2 - \frac{L}{2(t^k)^2}\|x^* - z^{k+1}\|^2 \\
&\leq\; \frac{t^k - 1}{t^k}F(x^k) + \frac{1}{t^k}F(x^*) + \frac{L}{2(t^k)^2}\|x^* - z^k\|^2 - \frac{L}{2(t^k)^2}\|x^* - z^{k+1}\|^2
\end{aligned} \tag{22}$$

In the above, the second last inequality used that fact that $t^k \geq 1$ for all k and the convexity of ℓ_f, and the last inequality used (12).

Subtracting $F(x^*)$ from both sides of (22) and then multiplying both sides by $(t^k)^2$ yields

$$(t^k)^2(F(x^{k+1}) - F(x^*)) \;\leq\; (t^{k-1})^2(F(x^k) - F(x^*)) + \frac{L}{2}\|x^* - z^k\|^2 - \frac{L}{2}\|x^* - z^{k+1}\|^2. \tag{23}$$

In (23), we used the fact that $(t^{k-1})^2 = t^k(t^k - 1)$. From (23), $t^0 = 1$ and $z^0 = x^0$, we get

$$(t^k)^2(F(x^{k+1}) - F(x^*)) \leq \frac{L}{2}\|x^* - x^0\|^2. \tag{24}$$

By (17), we obtain (18). In addition, by using the inequality, $\|x^* - x^0\| \leq \|x^*\| + \|x^0\| \leq \|x^*\|_1 + \|x^0\|$ and Lemma 2.1, we get the required result in (19). ∎

We observe that the APG algorithm is as simple as the IST and the 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 is also demonstrated by the numerical simulations in Section 4.

By using $f(x_1, x_2)$ instead of $f(x)$, the APG algorithm for solving (2) can be easily extended to solve (5). Hence the $O(\sqrt{L/\epsilon})$ iteration complexity still holds for the APG algorithm for solving (5). We omit the detailed discussions here for the extended problem.

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 better iteration complexity. In this section, we estimate the Lipschitz constants for the gradients of $f(x)$ in (2) and (5), arising from image restoration problems.

From (8), we have

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

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

Proposition 3.1 *For the problem (2), the Lipschitz constant L of ∇f defined in (6) has the following upper bound:*

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

For the problem (5), where one uses a two frame system, we have the following result on the Lipschitz constant L of ∇f defined in (7).

Proposition 3.2 *For the problem (5), the Lipschitz constant L of ∇f defined in (7) has the following upper bound:*

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

Proof. Indeed, from (9), we have

$$\begin{aligned}\nabla^2 f(x_1,x_2) &= \begin{pmatrix} W_1A^TDAW_1^T+\kappa_1(I-W_1W_1^T) & W_1A^TDAW_2^T \\ W_2A^TDAW_1^T & W_2A^TDAW_2^T+\kappa_2(I-W_2W_2^T)\end{pmatrix} \\ &= \begin{pmatrix} W_1 \\ W_2\end{pmatrix} A^TDA\begin{pmatrix} W_1^T & W_2^T\end{pmatrix} + \begin{pmatrix} \kappa_1(I-W_1W_1^T) & 0 \\ 0 & \kappa_2(I-W_2W_2^T)\end{pmatrix}\end{aligned}$$

Since $W_1^TW_1 = I$, $W_2^TW_2 = I$, and $D \succ 0$, $\nabla^2 f(x_1,x_2) \succeq 0$, we can take the Lipschitz constant L to be the following upper bound:

$$\lambda_{\max}(\nabla^2 f(x_1,x_2)) \le \lambda_{\max}\left(\begin{pmatrix} W_1 \\ W_2\end{pmatrix} A^TDA\begin{pmatrix} W_1^T & W_2^T\end{pmatrix}\right) + \max\{\kappa_1,\kappa_2\}.$$

■

The above propositions derived an upper bound for the Lipschitz constant of ∇f for a general frame based image restoration problem. Next we consider the special case of *image inpainting* problem using the model (2) with $D = I$. Note that for the image inpainting problem, A is a diagonal matrix with diagonals 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, $1+\kappa$, from (25) for the Lipschitz constant of ∇f. However, for the image inpainting problem (2) with $D = I$, we can obtain a tighter Lipschitz constant for ∇f, as shown in the next proposition.

Proposition 3.3 *For the inpaiting problem (2) with $D = I$ the Lipschitz constant L of ∇f defined in (6) is bounded by* $\max\{1,\kappa\}$. *Note that A is a diagonal matrix with diagonals 1 if the corresponding pixel values are known, but 0 otherwise.*

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

$$W^T = \begin{pmatrix} W_1^T \\ W_2^T\end{pmatrix}, \quad W = \begin{pmatrix} W_1 & W_2\end{pmatrix}, \quad A = \begin{pmatrix} I & 0 \\ 0 & 0\end{pmatrix}.$$

Hence

$$\begin{aligned}\nabla^2 f(x) &= \begin{pmatrix} W_1 & W_2\end{pmatrix}\begin{pmatrix} I & 0 \\ 0 & 0\end{pmatrix}\begin{pmatrix} W_1^T \\ W_2^T\end{pmatrix} + \kappa(I-WW^T) \\ &= W_1W_1^T + \kappa(I - W_1W_1^T - W_2W_2^T).\end{aligned}$$

We have two cases to consider.

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

$$\begin{aligned}\nabla^2 f(x) &\preceq W_1W_1^T + \kappa(I-W_1W_1^T) = \kappa I + (1-\kappa)W_1W_1^T \\ &\preceq \kappa I + (1-\kappa)I = I.\end{aligned}$$

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

$$\nabla^2 f(x) = \kappa I - (\kappa-1)W_1W_1^T - \kappa W_2W_2^T \preceq \kappa I.$$

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

Similarly, when one uses a two frame system for the inpainting problem in (5) with $D = I$, the Lipschitz constant L of ∇f defined in (7) can be bounded by the shaper constant $1 + \kappa_{\max} + \max\{0, 1 - \kappa_{\min}\}$, instead of the obvious bound of $2 + \kappa_{\max}$ derived from (26). Here, $\kappa_{\min} = \min\{\kappa_1, \kappa_2\}$ and $\kappa_{\max} = \max\{\kappa_1, \kappa_2\}$.

Proposition 3.4 *For cartoon and texture inpainting problem (5) with $D = I$, the Lipschitz constant L of ∇f defined in (7) is bounded by $1 + \kappa_{\max} + \max\{0, 1 - \kappa_{\min}\}$.*

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

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

We have two cases to consider.

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

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

This implies

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

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

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

This implies that

$$\lambda_{\max}(\nabla^2 f(x_1, x_2)) \;\; \leq \;\; \kappa_{\max} + \|W_{1a} W_{2a}^T\|_2 \;\leq\; \kappa_{\max} + 1.$$

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

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 form (2) or (5) 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 by using 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 by using 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, 43]. We briefly describe this strategy. If the both problems (2) and (5) are to be solved with the target parameter value $\bar{\lambda}$, we propose solving a sequence of problems of the forms (2) and (5) 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 indicates 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 open 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 (2) and set $\upsilon = 0.7$ for (5). 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 (2) is that $\delta(x) := \text{dist}(0; \partial(f(x) + \lambda^T|x|))$ is sufficiently small, where $\partial(\cdot)$ denotes the sub-differential. Here $\text{dist}(x; S)$ denotes the distance between a point x and a set S. Since

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

the upper bound on $\delta(x)$ can be 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 (13), 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

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

Hence we have the following upper bound for $\delta(x^{k+1})$ with $\delta(x^{k+1}) \leq 2L\|y^k - x^{k+1}\|$. From the above, we derive the following stopping condition for the APG algorithm:

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

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 the 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 (5) 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 DCT.

We set the parameters $\kappa = \kappa_1 = \kappa_2 = 1$ throughout. Our numerical experience showed that this choice produces the best results 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 $L = 1$ for ∇f in the problem (2) while placing the maximum penalty on $\|(I - WW^T)x\|$ among all the choices of κ such that the associated Lipschitz constant for ∇f is fixed at $L = 1$. A similar statement also holds for the image inpainting problem involving the problem (5). It is also interesting to note that the parameter κ is also chosen to be one in [5, 6, 7, 8, 14, 15, 16, 17, 18], although the derivation of algorithms is different from here. In [7, 15, 16, 17, 18], each algorithm was developed by using both properties of frames and the nature of the image restoration problem. It was then proven that its limit is the minimizer of (2) with $\kappa = 1$. In this paper, we formulate the image restoration problem in terms of the minimization problems (2) or (5) as motivated by results of [5, 6, 7, 8, 14, 15, 16, 17, 18], and develop a fast algorithm to find the minimizer.

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

$$b = P_\Omega(u + \sigma\xi),$$

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 such that shapes and patterns are consistent with our human vision.

We propose to recover the true image u by solving either (2) or (5), with

$$A = P_\Omega, \quad D = I. \tag{31}$$

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

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

where u and $\tilde{u}$ are the original and restored images, respectively, and m, n are the dimensions of u.

In Table 1, we report the numerical performance of the APG algorithm applied to (2) and (5) for the image inpainting problem. In the table, we report 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 to solve the model (2) for all the images. For the model (5), the APG algorithm took no more than 35 iterations to solve all the problems. Though the second model (5) is more expensive to solve, the PSNR values of the restored images are consistently better than those obtained from the first model (2) 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 [13]. Although [13] focuses on the split Bregman iterations for image restoration by 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 (2) and achieved a PSNR value of 33.69. In contrast, the numerical result for proximal forward-backward splitting (PFBS) algorithm took 329 iterations to solve the same problem provided in [13] 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 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 (2) and (5) 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	3.38	29	33.66	6.39
goldhill256	24	32.21	3.79	32	32.09	7.10
boat256	23	30.99	3.63	29	30.87	6.51
camera256	23	30.13	3.58	29	30.44	6.47
bridge256	26	31.31	4.15	33	31.08	7.47
bowl256	23	34.38	3.53	35	36.02	7.66
barbara512	27	31.33	22.47	31	33.82	34.12
baboon512	26	29.12	22.23	32	29.10	35.79
fingerprint512	25	26.51	21.23	34	28.00	38.20
zebra512	25	28.47	20.93	33	29.32	36.43

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

image	problem type	APG on (2)	[13]		
			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	29/28.44/31.5	55/29.01/71.6		
`goldhill256`	deblur `average`, `9` $\sigma = 3$	27/26.45/5.1	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 $\sigma = 20$. 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 32 iterations to solve all the problems in less than 40 seconds. Observe that the number of iterations taken to solve the problem (2) is consistently less than that for the problem (5). This is not surprising since the Lipschitz constant for ∇f in (2) is smaller than that for the corresponding function in (5).

Table 3: Numerical results for the APG algorithm in solving (2) and (5) 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	2.48	29	29.29	6.33
barbara512	17	27.39	14.36	29	28.44	31.47
baboon512	17	25.81	14.37	29	25.39	31.72
fingerprint512	17	27.74	14.75	32	27.51	34.78
zebra512	17	28.19	14.35	29	28.10	31.43

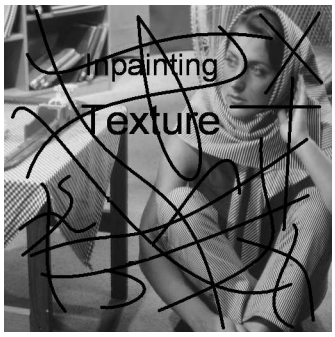
From Table 2, we see that the APG algorithm is more efficient than the split Bregman iteration developed in [13] for the denoising problem involving the cartoon-texture image `barbara512`. In this case, the former took 29 iterations to achieve a PSNR value of 28.44 whereas the latter took 55 iterations 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 (5). 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 [7, 13], 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 indicates that the performance of the APG algorithm is independent of the images and robust.

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

(a) original: 512×512



(b) observed image

(c) inpainted from (2): one system

(d) inpainted from (5): two systems

Table 4: Numerical results for the APG algorithm in solving (5) 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	6.34
goldhill256				28	39.99	6.39
boat256				27	40.81	6.13
camera256				28	41.96	6.29
bridge256				27	39.58	6.33
bowl256				28	43.46	6.20
barbara512				29	41.88	32.43
baboon512				27	39.39	31.33
fingerprint512				27	41.06	31.21
zebra512				28	42.95	31.53

Figure 2: Image decomposition results.

(a) original: 512×512

(b) decomposition via (5)

(c) cartoon part

(d) texture part

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 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 (2) with D being a positive definite matrix depending on A. As mentioned in the Introduction, real images usually have two layers. Hence we also consider the problem (5). 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 [19, 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 to 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 transforms or Discrete cosine transforms [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.

Table 5: Numerical results for the APG algorithm in solving (2) and (5) 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	5.64	0.24	34	27.51	8.81
peppers256	motion,15,30	0.50	28	27.37	5.55	0.30	28	26.99	7.04
peppers256	gaussian,15,2	0.30	22	25.82	4.31	0.18	21	25.74	5.29
peppers256	average,9	0.35	28	27.84	5.26	0.21	29	27.23	7.11
goldhill256	disk,3	0.40	27	27.28	5.46	0.24	37	27.05	9.45
goldhill256	motion,15,30	0.50	27	26.86	5.26	0.30	29	26.74	7.16
goldhill256	gaussian,15,2	0.30	22	26.51	4.27	0.18	22	26.52	5.51
goldhill256	average,9	0.35	27	26.45	5.12	0.21	31	26.31	7.55
boat256	disk,3	0.40	28	26.49	5.56	0.24	36	25.92	9.16
boat256	motion,15,30	0.50	27	25.95	5.24	0.30	29	25.61	7.33
boat256	gaussian,15,2	0.30	23	25.15	4.43	0.18	21	25.02	5.27
boat256	average,9	0.35	27	25.29	5.13	0.21	31	25.08	7.65
camera256	disk,3	0.40	28	26.98	5.63	0.24	35	26.39	8.89

Table 5: Numerical results for the APG algorithm in solving (2) and (5) 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
camera256	motion,15,30	0.50	28	26.44	5.42	0.30	29	26.05	7.25
camera256	gaussian,15,2	0.30	22	25.08	4.16	0.18	22	24.97	5.43
camera256	average,9	0.35	28	25.47	5.22	0.21	30	25.17	7.27
bridge256	disk,3	0.40	28	25.47	5.66	0.24	38	25.26	9.73
bridge256	motion,15,30	0.50	28	25.04	5.54	0.30	33	24.82	8.24
bridge256	gaussian,15,2	0.30	24	24.42	4.66	0.18	22	24.38	5.56
bridge256	average,9	0.35	28	24.35	5.32	0.21	32	24.23	7.77
bowl256	disk,3	0.40	26	29.09	5.14	0.24	35	29.08	8.87
bowl256	motion,15,30	0.50	26	29.34	5.03	0.30	29	29.57	7.24
bowl256	gaussian,15,2	0.30	22	28.54	4.24	0.18	21	28.66	5.28
bowl256	average,9	0.35	25	28.84	4.65	0.21	31	29.05	7.60
barbara512	disk,3	0.40	29	25.35	29.39	0.24	36	25.78	44.67
barbara512	motion,15,30	0.50	28	25.02	27.89	0.30	35	25.64	42.66
barbara512	gaussian,15,2	0.30	21	24.21	20.63	0.18	19	24.20	23.37
barbara512	average,9	0.35	26	24.12	25.03	0.21	30	24.18	35.92
baboon512	disk,3	0.40	29	23.49	29.62	0.24	39	23.47	48.54
baboon512	motion,15,30	0.50	28	23.44	28.04	0.30	37	23.34	45.52
baboon512	gaussian,15,2	0.30	24	22.20	23.46	0.18	22	22.16	27.08
baboon512	average,9	0.35	28	22.31	27.00	0.21	33	22.22	39.81
fingerprint512	disk,3	0.40	29	27.42	29.81	0.24	37	27.84	47.17
fingerprint512	motion,15,30	0.50	30	25.49	29.98	0.30	32	25.54	39.35
fingerprint512	gaussian,15,2	0.30	26	27.32	25.83	0.18	24	27.42	29.88
fingerprint512	average,9	0.35	31	24.48	29.89	0.21	30	24.67	36.16
zebra512	disk,3	0.40	29	27.28	29.57	0.24	39	27.46	48.40
zebra512	motion,15,30	0.50	30	25.99	29.86	0.30	36	26.21	43.75
zebra512	gaussian,15,2	0.30	25	25.98	24.80	0.18	27	25.96	32.97
zebra512	average,9	0.35	29	24.74	28.20	0.21	30	24.73	35.87

Table 6: Numerical results for the APG algorithm in solving (2) and (5) 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	5.47	0.60	26	26.37	6.54
peppers256	motion,15,30	1.00	31	25.72	5.90	0.60	30	25.50	7.50
peppers256	gaussian,15,2	1.00	25	25.27	4.72	0.60	22	25.12	5.55
peppers256	average,9	1.00	31	26.29	5.71	0.60	28	25.74	6.87
goldhill256	disk,3	1.00	26	26.63	5.05	0.60	23	26.57	5.91
goldhill256	motion,15,30	1.00	30	25.80	5.78	0.60	29	25.80	7.27
goldhill256	gaussian,15,2	1.00	24	26.15	4.51	0.60	21	26.11	5.25

Table 6: Numerical results for the APG algorithm in solving (2) and (5) 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
goldhill256	average,9	1.00	29	25.62	5.30	0.60	26	25.54	6.31
boat256	disk,3	1.00	27	25.49	5.29	0.60	26	25.17	6.60
boat256	motion,15,30	1.00	31	24.75	5.94	0.60	30	24.56	7.53
boat256	gaussian,15,2	1.00	25	24.62	4.69	0.60	22	24.48	5.52
boat256	average,9	1.00	30	24.42	5.50	0.60	26	24.20	6.30
camera256	disk,3	1.00	28	25.67	5.43	0.60	29	25.29	7.38
camera256	motion,15,30	1.00	32	25.04	6.10	0.60	29	24.71	7.28
camera256	gaussian,15,2	1.00	25	24.55	4.63	0.60	23	24.42	5.76
camera256	average,9	1.00	30	24.36	5.43	0.60	27	24.08	6.60
bridge256	disk,3	1.00	28	24.66	5.48	0.60	26	24.51	6.59
bridge256	motion,15,30	1.00	31	23.99	5.96	0.60	31	23.87	7.74
bridge256	gaussian,15,2	1.00	26	23.91	4.95	0.60	23	23.83	5.78
bridge256	average,9	1.00	31	23.50	5.71	0.60	28	23.38	6.80
bowl256	disk,3	1.00	26	28.36	5.01	0.60	23	28.42	5.75
bowl256	motion,15,30	1.00	29	28.05	5.50	0.60	29	28.35	7.10
bowl256	gaussian,15,2	1.00	24	27.68	4.46	0.60	22	27.71	5.48
bowl256	average,9	1.00	28	28.16	5.03	0.60	25	28.42	6.07
barbara512	disk,3	1.00	26	24.47	26.11	0.60	32	24.79	39.66
barbara512	motion,15,30	1.00	30	24.15	29.69	0.60	30	24.43	36.58
barbara512	gaussian,15,2	1.00	23	24.04	22.48	0.60	20	24.02	24.48
barbara512	average,9	1.00	28	23.78	26.67	0.60	25	23.82	29.88
baboon512	disk,3	1.00	28	22.52	28.32	0.60	37	22.49	46.11
baboon512	motion,15,30	1.00	31	22.49	30.94	0.60	31	22.38	37.89
baboon512	gaussian,15,2	1.00	25	21.79	24.17	0.60	22	21.74	27.02
baboon512	average,9	1.00	30	21.59	28.40	0.60	27	21.50	32.44
fingerprint512	disk,3	1.00	30	26.78	30.88	0.60	26	27.17	32.68
fingerprint512	motion,15,30	1.00	34	24.17	33.99	0.60	39	24.17	48.53
fingerprint512	gaussian,15,2	1.00	27	26.49	26.90	0.60	26	26.43	32.15
fingerprint512	average,9	1.00	34	23.38	32.87	0.60	33	23.45	39.87
zebra512	disk,3	1.00	29	26.04	29.01	0.60	28	26.03	34.77
zebra512	motion,15,30	1.00	33	24.62	32.68	0.60	35	24.72	42.26
zebra512	gaussian,15,2	1.00	28	25.16	27.37	0.60	25	25.04	30.69
zebra512	average,9	1.00	32	23.65	30.46	0.60	28	23.55	33.77

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 (2) and (5), corresponding to the cases $\sigma = 3$ and $\sigma = 5$, respectively. As indicated in the tables, it took the AGP algorithm no more than 40 iterations to solve all the image deblurring problems via (2) or (5), and all the 512×512 problems are solved in less than 50 seconds. The restored `barbara512` images are shown in Figure 3.

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 (2)

(d) deblurred via (5)

Next, we compare the performance of the APG algorithm with the split Bregman algorithm and PFBS algorithm in [13] 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 (2) in 27 iterations and achieved a PSNR value of 26.45. This result is competitive to the result obtained by the split Bregman algorithm in [13] 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 (2).

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 (2) and achieved a PSNR value of 25.46. Our APG algorithm is again competitive to the split Bregman and linearized Bregman algorithms. In [13], 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 for solving this convex nonsmooth minimization problem and given an estimation for the Lipschitz constant (which determines the convergence speed of our algorithm) of the gradient of the smooth part of the objective function. 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, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, 2008, SIAM J. Imaging Sci., to appear.

[2] S. Becker, J. Bobin, and E. J. Candés, *NESTA: a fast and accurate first-order method for sparse recovery*, Preprint, 2009.

[3] D. P. Bertsekas, *Nonlinear Programming*, 2nd edition, Athena Scientific, Belmont, 1999.

[4] J.-F. Cai, E. J. Candès, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, Preprint, 2008.

[5] 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.

[6] 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.

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

[8] J.-F. Cai, R. H. Chan, and Z. Shen, *Simultaneous cartoon and texture inpainting*, Preprint, 2008.

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

[10] J.-F. Cai, S. Osher, and Z. Shen, *Convergence of the Linearized Bregman Iteration for ℓ_1-norm Minimization*, Math. Comp., 78 (2009), pp. 2127–2136.

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

[12] 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.

[13] J.-F. Cai, S. Osher, and Z. Shen, *Split Bregman methods and frame based image restoration*, Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, to appear.

[14] J.-F. Cai and Z. Shen, *Framelet Based Deconvolution*, 2008, J. Computational Mathematics, to appear.

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

[16] 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.

[17] 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.

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

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

[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] D. Donoho, *Compressed sensing*, IEEE Trans. Info. Theory, 52 (2006), pp. 1289–1306.

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

[25] 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.

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

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

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

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

[30] T. Goldstein and S. Osher, *The Split Bregman Algorithm for L1 Regularized Problems*, CAM Report (08-29), UCLA, 2008.

[31] E. T. Hale, W. Yin, and Y. Zhang, *A fixed-point continuation method for ℓ_1-regularized minimization with applications to compressed sensing*, CAAM Technical Report TR07-07, Department of Computational and Applied Mathematics, Rice University, 2007.

[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*, Preprint, 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, *Introductory Lectures on Convex Optimization*, Kluwer Academic Publisher, Dordrecht, The Netherlands, 2004.

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

[39] S. Osher, Y. Mao, B. Dong, and W. Yin, *Fast Linearized Bregman Iteration for Compressed Sensing and Sparse Denoising*, CAM Reprots (08-37), UCLA, 2008.

[40] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, 1970.

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

[42] 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.

[43] K.-C. Toh and S. Yun, *An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems*, 2009, Pacific J. Optim., to appear.

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

[45] 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.

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