Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems *

Amir Beck† and Marc Teboulle ‡

June 11, 2009

Abstract

This paper studies gradient-based schemes for image denoising and deblurring problems based on the discretized total variation (TV) minimization model with constraints. We derive a fast algorithm for the constrained TV-based image deburring problem. To achieve this task we combine an acceleration of the well known dual approach to the denoising problem with a novel monotone version of a fast iterative shrinkage/thresholding algorithm (FISTA) we have recently introduced. The resulting gradient-based algorithm shares a remarkable simplicity together with a proven global rate of convergence which is significantly better than currently known gradient projections-based methods. Our results are applicable to both the anisotropic and isotropic discretized TV functionals. Initial numerical results demonstrate the viability and efficiency of the proposed algorithms on image deblurring problems with box constraints.

1 Introduction

In this paper we propose fast gradient-based algorithms for the constrained total variation (TV) based image denoising and deblurring problems. The total variation model has been introduced by Rudin-Osher and Fatemi (ROF) in [24] as a regularization approach capable of handling properly edges and removing noise in a given image. This model has proven to be successful in a wide range of applications in image processing. The discrete penalized version of the TV-based deburring model consists of solving an unconstrained convex minimization problem of the form,

$$\min_{\mathbf{x}} \|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|^2 + 2\lambda\|\mathbf{x}\|_{TV}, \tag{1.1}$$

where $\|\cdot\|$ is a norm in some given vector space, $\mathbf{b}$ is the observed noisy data, $\mathcal{A}$ is a linear map representing some blurring operator, $\|\cdot\|_{TV}$ is a discrete TV (semi)-norm, and $\mathbf{x}$ is

*This research is partially supported by the Israel Science Foundation, ISF grant #489-06.

†Department of Industrial Engineering and Management, Technion—Israel Institute of Technology, Haifa 32000, Israel. E-mail: becka@ie.technion.ac.il

‡School of Mathematical Sciences, Tel-Aviv University, Ramat-Aviv 69978, Israel, E-mail: teboulle@math.tau.ac.il

the desired unknown image to be recovered (see Section 2 for more precise details). The regularization parameter $\lambda > 0$ provides a tradeoff between fidelity to measurements and noise sensitivity. When $\mathcal{A}$ is the identity operator, the problem is called *denoising*.

Recently, intensive research has focused on developing efficient methods to solve problem (1.1). In particular, a main focus has been on the simpler denoising problem, with resulting algorithms that often cannot be readily extended to handle the more difficult deblurring problem, and a-fortriori when the problem also includes constraints.

One of the key difficulties in the TV-based image deblurring problem is the presence of the *nonsmooth* TV norm in its objective. Another difficulty is the inherent very large scale nature of the optimization problem. This renders the task of building fast and simple numerical methods difficult, and hence the continued motivation and research efforts for building adequate methods. The design of fast methods are usually based on sophisticated methods and often require heavy computations, such as solution of huge scale linear systems as well as high memory requirements. On the other hand, simpler methods relying on less demanding computational efforts, such as matrix-vector multiplications, and requiring lower memory requirements are more suitable and attractive to tackle large scale image processing problems. However, these simple algorithms which are usually based on first order information, often exhibit very slow convergence rate. Thus, a natural objective is to search and devise schemes that remain simple, but can exhibit much faster performance. This is the main objective of this paper where we present very simple and fast algorithms for constrained TV-based image denoising and deblurring problems.

The literature on numerical methods for solving (1.1) abounds and include for example methods based on: PDE and fixed point techniques, smoothing approximation methods, primal-dual Newton-based methods, dual methods, primal-dual active set methods, interior point algorithms and second-order cone programming, see for instance, ([7, 5, 6, 8, 13, 15, 24, 27], and references therein. This list is surely not exhaustive and is just given as an indicator of the intense research activities in the field.

One method of particular interest to this paper is the dual approach introduced by Chambolle [5, 6] who developed a globally convergent gradient-based algorithm for the denoising problem and which was shown to be faster than primal-based schemes. This dual-based method of [5] is the starting point of this paper for the constrained denoising problem. To tackle the more involved constrained TV-based deblurring problem, the latter approach is combined with another method which is based on our recent study [1]. In that paper we introduced what we called a fast iterative shrinkage/thresholding algorithm (FISTA), for minimizing the sum of two convex functions

$$\min_{\mathbf{x}}\{F(\mathbf{x}) \equiv f(\mathbf{x}) + g(\mathbf{x})\}, \tag{1.2}$$

where f is smooth and g is nonsmooth. One of the particular characteristics of FISTA is that it uses in each iteration information on the two previous iterates which are combined in a very special way. In [1] we proved that this scheme exhibits a global convergence rate $O(1/k^2)$, where k is the iteration counter, and which is significantly faster than standard gradient-based methods such as iterative shrinkage thresholding algorithms (ISTA), see e.g., [3, 10] and references therein. The scheme developed in [1] relies on an extension of a gradient

projection method which is not so well known and was invented by Nesterov in 1983 [20] for minimizing *smooth* convex functions. For the class of smooth problems Nesterov also showed that this rate of convergence is "optimal"[1]

Very recently, in [19], Nesterov has also independently studied a multistep accelerated gradient-based method for solving the convex nonsmooth problem (1.2). This algorithm is quite different and much more involved than FISTA, yet it also achieved the same fast convergence rate, see [1, 19] for more details. Finally, it is also interesting to note that recently a different a two-step algorithm, called TwIST, was introduced and studied in [3]. The numerical results in [3] demonstrate the potential advantages of TwIST as compared to other one-step methods.

Contributions

In this paper we propose very simple and fast gradient-based methods for TV-based denoising and deblurring problems. Our contributions are threefold. First we consider the dual-based approach of Chambolle [5, 6] extended to constrained problems and we introduce a fast gradient projection (FGP) method which is shown to accelerate the algorithm of [6]. Second, we introduce and analyze a monotone version of FISTA, called MFISTA, which is applied to the TV-based deblurring problem and is shown to exhibit fast theoretical and practical rate of convergence. Finally, our approach is quite general and can handle bound constraints as well as tackle both the isotropic and anisotropic TV functions. Our initial numerical experiments confirm the predicted underlying theoretical results and even beyond, showing that the proposed framework leads to algorithms that can perform well and are significantly faster in comparison to currently known gradient projections-based methods. A MATLAB implementation and documentation of the FGP and MFISTA methods for image denoising and deblurring problems are available at

`http://iew3.technion.ac.il/~becka/papers/tv_fista.zip`

Outline of the paper

In the next section we introduce the necessary notation and the TV-based regularization models. Section 3 develops the mathematical framework for gradient-based schemes and presents FISTA of [1]. In Section 4 we develop and analyze dual-based algorithms for the constrained denoising problem and introduce a fast gradient projection scheme. Building on these results, in Section 5 we tackle the constrained deblurring by introducing a novel monotone version of FISTA which is as simple as FISTA and is proven to preserve its fast global convergence rate, see the appendix for the proof. Finally, Section 6 describes initial numerical results for constrained TV-based deblurring problems which demonstrate the efficiency, competitiveness and viability of the proposed methods.

Notations The vector space $\mathbb{E}$ stands for a finite dimensional Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \| = \langle \cdot, \cdot \rangle^{1/2}$, so that its norm is self dual $\| \cdot \|_* \equiv \| \cdot \|$. Often, we will apply our results with $\mathbb{E} = \mathbb{R}^{m \times n}$, so that the inner product is given by $\langle \mathbf{x}, \mathbf{y} \rangle = \mathrm{Tr}(\mathbf{x}\mathbf{y}^T)$ and the norm of the matrix $\mathbf{x} \in \mathbb{R}^{m \times n}$ is the Frobenius norm $\|\mathbf{x}\| \equiv \|\mathbf{x}\|_F = \sqrt{\mathrm{Tr}(\mathbf{x}\mathbf{x}^T)}$.

[1] Loosely saying, "optimal" here means that for every method that exploits only function and gradient evaluation, it is always possible to find a "worst case" problem instance for which the rate of convergence of the method is indeed of the order of $1/k^2$.

Furthermore, we will also use some standard definitions/notations from convex analysis which can be found in [23].

2 The Discrete Total Variation Regularization Model

This short section introduces the necessary notations and material for the problems studied in this paper. We consider images that are defined on rectangle domains. Let $\mathbf{b} \in \mathbb{R}^{m\times n}$ be an observed noisy image, $\mathbf{x} \in \mathbb{R}^{m\times n}$ the true (original) image to be recovered, $\mathcal{A}$ an affine map representing a blurring operator, and $\mathbf{w} \in \mathbb{R}^{m\times n}$ a corresponding additive unknown noise satisfying the relation:

$$\mathbf{b} = \mathcal{A}(\mathbf{x}) + \mathbf{w}. \tag{2.1}$$

The problem of finding an $\mathbf{x}$ from the above relation is the basic discrete linear inverse problem which has been extensively studied in various signal/image recovery problems in the literature via regularization methods, see e.g., [4, 11, 26]. Here we are concerned with TV-based regularization, which, given $\mathcal{A}$ and $\mathbf{b}$ seeks to recover $\mathbf{x}$ by solving the convex nonsmooth minimization problem

$$\min_{\mathbf{x}}\{\|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|^2 + 2\lambda \mathrm{TV}(\mathbf{x})\}, \quad (\lambda > 0), \tag{2.2}$$

where $\mathrm{TV}(\cdot)$ stands for the discrete total variation semi-norm $\|\mathbf{x}\|_{TV}$ given in the introduction (cf. (1.1)). The identity map will be denoted by $\mathcal{I}$ and with $\mathcal{A} \equiv \mathcal{I}$ problem (2.2) reduces to the denoising problem.

Two popular choices for the discrete TV are the isotropic TV defined by (see [5]),

$$\begin{aligned} \mathbf{x} \in \mathbb{R}^{m\times n}, \quad \mathrm{TV}_I(\mathbf{x}) &= \sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\sqrt{(x_{i,j} - x_{i+1,j})^2 + (x_{i,j} - x_{i,j+1})^2} \\ &\quad + \sum_{i=1}^{m-1}|x_{i,n} - x_{i+1,n}| + \sum_{j=1}^{n-1}|x_{m,j} - x_{m,j+1}| \end{aligned}$$

and the l_1-based, anisotropic TV defined by

$$\begin{aligned} \mathbf{x} \in \mathbb{R}^{m\times n}, \quad \mathrm{TV}_{l_1}(\mathbf{x}) &= \sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\{|x_{i,j} - x_{i+1,j}| + |x_{i,j} - x_{i,j+1}|\} \\ &\quad + \sum_{i=1}^{m-1}|x_{i,n} - x_{i+1,n}| + \sum_{j=1}^{n-1}|x_{m,j} - x_{m,j+1}|, \end{aligned}$$

where in the above formulas we assumed the (standard) reflexive boundary conditions:

$$x_{m+1,j} - x_{m,j} = 0, \ \forall\, j \text{ and } x_{i,n+1} - x_{i,n} = 0, \ \forall i.$$

The algorithms developed in this paper can be applied to both the isotropic and anisotropic TV. Since the derivations and results for the isotropic and l_1-based cases are very similar, to avoid repetition, all of our derivations will consider the isotropic TV (TV_I), and the corresponding results for the l_1-based TV will be shortly outlined.

In this paper we also consider the more general constrained TV-based deblurring problem:

$$\min_{\mathbf{x}\in C}\|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|_F^2 + 2\lambda \mathrm{TV}(\mathbf{x}), \tag{2.3}$$

where C is a convex closed set. The unconstrained case corresponds to $C = \mathbb{R}^{m\times n}$. Note that the previous cited works and algorithms in the introduction do not handle constrained TV-based deblurring problems (although some of them can be modified to incorporate constraints). We will be especially interested in the case $C = B_{l,u}$ where $B_{l,u}$ is the $n-$dimensional cube given by

$$B_{l,u} = \{\mathbf{x} : l \leq x_{ij} \leq u,\ \forall i, j\}\,.$$

Bound constraints of course model the situation in which all pixels have lower and upper bounds (such as 0 and 255 or 0 and 1). We do not restrict the lower and upper bounds to be finite. For example, we might choose $l = 0, u = \infty$ which corresponds to nonnegativity constraints.

3 Gradient-Based Algorithms

3.1 The General Optimization Model

For the purpose of our analysis, we consider the following useful nonsmooth convex optimization model:

$$(P) \qquad \min\{F(\mathbf{x}) \equiv f(\mathbf{x}) + g(\mathbf{x}) : \mathbf{x} \in \mathbb{E}\} \tag{3.1}$$

with the following assumptions:

- $g : \mathbb{E} \to (-\infty, +\infty]$ is a proper closed convex function.
- $f : \mathbb{E} \to \mathbb{R}$ is continuously differentiable with Lipschitz continuous gradient $L(f)$:

  $$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L(f)\|\mathbf{x} - \mathbf{y}\| \quad \text{for every } \mathbf{x}, \mathbf{y} \in \mathbb{E},$$

  where $\|\cdot\|$ denotes the standard Euclidean norm and $L(f) > 0$ is the Lipschitz constant of ∇f.
- Problem (P) is solvable, i.e., $X_* := \operatorname{argmin} F \neq \emptyset$, and for $\mathbf{x}^* \in X_*$ we set $F_* := F(\mathbf{x}^*)$.

Problem (P) is rich enough to cover basic generic optimization problems. In particular the standard smooth convex constrained minimization problem:

$$\min\{f(\mathbf{x}) :\ \mathbf{x} \in C\} \tag{3.2}$$

is obtained by choosing $g(\mathbf{x}) \equiv \delta_C(\mathbf{x})$, with $C \subseteq \mathbb{E}$ some closed convex set and δ_C being the indicator function on C. Likewise, with $f(\mathbf{x}) \equiv 0$, the general nonsmooth convex minimization problem is obtained.

The proposed model naturally includes the problem formulations of the constrained TV-based denoising and deblurring which correspond to the choice

$$f(\mathbf{x}) \equiv \|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|^2 \ \text{ and } \ g(\mathbf{x}) \equiv 2\lambda \mathrm{TV}(\mathbf{x}) + \delta_C(\mathbf{x}),$$

and with the special choice $\mathcal{A} = \mathcal{I}$ to recover the denoising case.

For a recent account of other fundamental problems that can be cast as (P), and arising in various engineering applications, as well as corresponding relevant algorithms, we refer the reader to the recent comprehensive study of [9] and the literature therein.

3.2 Preliminaries: The Proximal Map

A common approach for solving problems of the form (3.1) is via simple fixed point iterations which naturally emerge from writing down the optimality condition for problem (3.1), see Section 3.3 below. A key player within this approach is the proximal map of Moreau [18] associated to a convex function. We recall its definition and some of its fundamental properties. Given a proper closed convex function $g : \mathbb{E} \to (-\infty, +\infty]$ and any scalar $t > 0$, the proximal map associated to g is defined by

$$\text{prox}_t(g)(\mathbf{x}) := \underset{\mathbf{u}}{\text{argmin}} \left\{ g(\mathbf{u}) + \frac{1}{2t}\|\mathbf{u} - \mathbf{x}\|^2 \right\}. \tag{3.3}$$

The proximal map associated with a closed proper convex function g enjoys and implies several important properties. The next result records two such important properties, for a proof see [18, Proposition 7].

Lemma 3.1. *Let $g : \mathbb{E} \to (-\infty, +\infty]$ be a closed proper convex function, and for any $t > 0$, let*

$$g_t(\mathbf{x}) := \inf_{\mathbf{u}} \left\{ g(\mathbf{u}) + \frac{1}{2t}\|\mathbf{u} - \mathbf{x}\|^2 \right\}. \tag{3.4}$$

Then,
(a) The infimum in (3.4) is attained at the unique point $prox_t(g)(\mathbf{x})$. As a consequence, the map $(I + t\partial g)^{-1}$ is single valued from $\mathbb{E}$ into itself and

$$(I + t\partial g)^{-1}(\mathbf{x}) \equiv prox_t(g)(\mathbf{x}),\ \forall \mathbf{x} \in \mathbb{E}. \tag{3.5}$$

(b) The function g_t is continuously differentiable on $\mathbb{E}$ with a $1/t$-Lipschitz gradient given by

$$\nabla g_t(x) = \frac{1}{t}(I - prox_t(g)(\mathbf{x})),\ \forall \mathbf{x} \in \mathbb{E}.$$

(c) In particular, if $g = \delta_C$, the indicator of a closed convex set $C \subset \mathbb{E}$, then $prox_t(g) = (I + t\partial\delta_C)^{-1} = P_C$, the Euclidean projection operator on C, and

$$g_t(\mathbf{x}) = \|P_C(\mathbf{x}) - \mathbf{x}\|^2.$$

3.3 The Basic Gradient-Based Model Algorithms

Writing down the optimality condition for problem (3.1), fix any scalar $t > 0$, then $\mathbf{x}^*$ solves the convex minimization problem (3.1) if and only if the following equivalent statements hold:

$$\begin{aligned} 0 &\in t\nabla f(\mathbf{x}^*) + t\partial g(\mathbf{x}^*), \\ 0 &\in t\nabla f(\mathbf{x}^*) - \mathbf{x}^* + \mathbf{x}^* + t\partial g(\mathbf{x}^*), \\ (I + t\partial g)(\mathbf{x}^*) &\in (I - t\nabla f)(\mathbf{x}^*), \\ \mathbf{x}^* &= (I + t\partial g)^{-1}(I - t\nabla f)(\mathbf{x}^*), \end{aligned} \tag{3.6}$$

where (3.6) follows from Lemma 3.1 (a). The equation (3.6) above naturally calls for the fixed point iterative scheme:

$$\mathbf{x}_0 \in \mathbb{E}, \quad \mathbf{x}_k = (I + t_k\partial g)^{-1}(I - t_k\nabla f)(\mathbf{x}_{k-1}),\ (t_k > 0). \tag{3.7}$$

The scheme (3.7) is nothing else but a special case of the so-called backward-forward splitting method introduced by Passty [21, see model (BF) in p. 384] for solving the inclusion (3.6). From Lemma 3.1(a), (3.7) can be re-written as

$$\begin{aligned} \mathbf{x}_k &= \operatorname{prox}_{t_k}(g)(\mathbf{x}_{k-1} - t_k \nabla f(\mathbf{x}_{k-1})) \\ &= \operatorname*{argmin}_{\mathbf{x}} \left\{ g(\mathbf{x}) + \frac{1}{2t_k} \left\| \mathbf{x} - (\mathbf{x}_{k-1} - t_k \nabla f(\mathbf{x}_{k-1})) \right\|^2 \right\}. \end{aligned} \tag{3.8}$$

This iterative scheme includes as special cases the following well known important algorithms:

- **Gradient Projection Method** When $g(\mathbf{x}) \equiv \delta_C(\mathbf{x})$, (3.8) reduces to the gradient projection algorithm, for solving a smooth constrained minimization problem, see e.g., [2].

$$\mathbf{x}_k = P_C \left(\mathbf{x}_{k-1} - t_k \nabla f(\mathbf{x}_{k-1}) \right). \tag{3.9}$$

- **Proximal Minimization Algorithm** When $f(\mathbf{x}) \equiv 0$, the problem consists of a nonsmooth convex minimization problem, and (3.8) reduces to the proximal minimization algorithm of Martinet [17],

$$\mathbf{x}_k = \operatorname*{argmin}_{x} \left\{ g(\mathbf{x}) + \frac{1}{2t_k} \|\mathbf{x} - \mathbf{x}_{k-1}\|^2 \right\}. \tag{3.10}$$

- **ISTA – Iterative Shrinkage/Thresholding Algorithm** When $g(\mathbf{x}) = \|\mathbf{x}\|_1$, the scheme (3.8) reduces to

$$\mathbf{x}_k = \mathcal{T}_{\lambda t_k} \left(\mathbf{x}_{k-1} - t_k \nabla f(\mathbf{x}_{k-1}) \right) \tag{3.11}$$

  where $\mathcal{T}_\alpha : \mathbb{E} \to \mathbb{E}$ is the shrinkage operator defined by

$$\mathcal{T}_\alpha(\mathbf{x})_i = (|x_i| - \alpha)_+ \operatorname{sgn}(x_i), \tag{3.12}$$

  which thanks to separability, is easily obtained by computing the proximal map in (3.8) of the one dimensional function $|x|$. In the special case $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$, the popular Iterative-Skrinkage-Thresholding Algorithm (ISTA) is recovered, see e.g., [9, 10] and references therein.

In the rest of this paper, even though our analysis will handle general convex nonsmooth regularizers and constraints, we will still refer to the corresponding method described by (3.7) or equivalently by (3.8), as ISTA.

A typical condition ensuring the convergence of the sequence $\mathbf{x}_k$ produced by the gradient projection algorithm and ISTA is to require $t_k \in (0, 2/L(f))$. For an extensive background, as well as extension, refinements and more general convergence results, see [12, Chapter 12] and [9], and references therein. The three algorithms just outlined above are known to be slow. In fact, they all share the same *sublinear* rate of convergence in function values, that is, the function values $F(\mathbf{x}_k)$ converge to F_* at a rate of $O(1/k)$, see e.g., [22, 1]. Specifically, if the stepsizes are all chosen as the constant $1/L$, where L is an upper bound on the Lipschitz constant of f, then for every $k \geq 1$,

$$F(\mathbf{x}_k) - F_* \leq \frac{L\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{2k}, \quad \forall \mathbf{x}^* \in X_*.$$

We show next that this nonasymptotic rate of convergence can be significantly improved.

3.4 A Fast Iterative Shrinkage Thresholding Algorithm: FISTA

As just seen above, when $g(\mathbf{x}) \equiv \delta_C(\mathbf{x})$, the iteration scheme (3.8) consists of minimizing a smooth convex function over the convex set C via the gradient projection method. In this smooth setting, in 1983 Nesterov [20] introduced a method that achieves a rate of convergence of $O(1/k^2)$, and which clearly is a significant improvement over the theoretical sublinear rate of convergence of the classical gradient projection method. The method is as simple as the gradient projection method and requires only one gradient evaluation at each step. Motivated by the latter, in our recent work [1], we were able to extend Nesterov's method to handle the more general convex nonsmooth minimization model (P), and which is now recalled below. For convenience, and ease of comparison with [1], in the rest of this paper we use the following short hand notation for the proximal map. For any $L > 0$,

$$p_L(\mathbf{y}) \equiv \text{prox}_{1/L}(g)\left(\mathbf{y} - \frac{1}{L}\nabla f(\mathbf{y})\right) = \underset{\mathbf{x}\in\mathbb{E}}{\text{argmin}}\left\{\frac{L}{2}\left\|\mathbf{x} - \left(\mathbf{y} - \frac{1}{L}\nabla f(\mathbf{y})\right)\right\|^2 + g(\mathbf{x})\right\}. \tag{3.13}$$

FISTA
Input: An upper bound $L \geq L(f)$ on the Lipschitz constant $L(f)$ of ∇f.
Step 0. Take $\mathbf{y}_1 = \mathbf{x}_0 \in \mathbb{E}$, $t_1 = 1$.
Step k. ($k \geq 1$) Compute

$$\mathbf{x}_k = p_L(\mathbf{y}_k), \tag{3.14}$$

$$t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \tag{3.15}$$

$$\mathbf{y}_{k+1} = \mathbf{x}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}_k - \mathbf{x}_{k-1}). \tag{3.16}$$

Each iterate of FISTA depends on the previous two iterates and not only on the last iterate as in ISTA. In addition, the operator p_L is employed on a very special linear combination of the two previous iterates $\{\mathbf{x}_{k-1}, \mathbf{x}_k\}$. Note that FISTA is as simple as ISTA and shares the same computational demand of ISTA, namely the computation of $p_L(\cdot)$, the remaining additional steps being computationally negligible. Thus, despite the presence of the nonsmooth term g, and as proven in [1], the rate of convergence of function values of FISTA to the optimal function value is of the order $O(1/k^2)$ as opposed to the slower $O(1/k)$ rate of convergence of ISTA. We recall here the precise convergence result for FISTA.

Theorem 3.1 ([1, Theorem 4.1]). *Let $\{\mathbf{x}_k\}$ be generated by FISTA. Then for any $k \geq 1$*

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{2L\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{(k+1)^2}, \quad \forall \mathbf{x}^* \in X_*.$$

In the smooth case, i.e., with $g(\mathbf{x}) \equiv \delta_C(\mathbf{x})$ this result recovers the fast gradient projection method of [20] devised for smooth convex constrained minimization, which is exactly FISTA with (3.14) replaced by

$$\mathbf{x}_k \equiv P_C\left(\mathbf{y}_k - \frac{1}{L}\nabla f(\mathbf{y}_k)\right).$$

For future reference, in the sequel, this method will be called Fast Projected Gradient (FGP).

Remark 3.1. For simplicity, we have presented FISTA with a fixed step size based on a known Lipschitz constant $L(f)$. However, the algorithm can be easily modified with a variable step size to handle the case when the Lipschitz constant is not known, and is shown to preserve its rate of convergence, see [1] for details.

4 Total Variation-Based Denoising

4.1 The Dual Approach

We consider the TV-based denoising problem with constraints which consists of solving

$$\min_{\mathbf{x}\in C} \|\mathbf{x}-\mathbf{b}\|_F^2 + 2\lambda \mathrm{TV}(\mathbf{x}), \tag{4.1}$$

where the nonsmooth regularizer functionals TV is either the isotropic TV_I or anisotropic TV_{l_1}, and C is a closed convex subset of $\mathbb{E} \equiv \mathbb{R}^{m\times n}$, see Section 2. The derivations and results for the isotropic and l_1-based cases are very similar. To avoid repetition, all of our derivations will consider the isotropic TV function (TV_I), although the results for the l_1-based TV function will also be presented.

One of the intrinsic difficulties in problem (4.1) is the nonsmoothness of the TV function. To overcome this difficulty, Chambolle [5] suggested to consider a dual approach, and proposed a gradient-based algorithm for solving the resulting dual problem, which in the unconstrained case was shown to be a convex quadratic program (maximization of a concave quadratic function subject to linear constraints). Here we follow his approach and construct a dual of the constrained problem. Some notation is in order:

- $\mathcal{P}$ is the set of matrix-pairs $(\mathbf{p},\mathbf{q})$ where $\mathbf{p}\in\mathbb{R}^{(m-1)\times n}$ and $\mathbf{q}\in\mathbb{R}^{m\times(n-1)}$ that satisfy

$$\begin{aligned} p_{i,j}^2 + q_{i,j}^2 &\le 1, \quad i=1,\dots,m-1, j=1,\dots,n-1,\\ |p_{i,n}| &\le 1, \quad i=1,\dots,m-1,\\ |q_{m,j}| &\le 1, \quad j=1,\dots,n-1. \end{aligned}$$

- The linear operation $\mathcal{L}:\mathbb{R}^{(m-1)\times n}\times\mathbb{R}^{m\times(n-1)}\to\mathbb{R}^{m\times n}$ is defined by the formula

$$\mathcal{L}(\mathbf{p},\mathbf{q})_{i,j} = p_{i,j}+q_{i,j}-p_{i-1,j}-q_{i,j-1}, \quad i=1,\dots,m, j=1,\dots,n,$$

  where we assume that $p_{0,j}=p_{m,j}=q_{i,0}=q_{i,n}\equiv 0$ for every $i=1,\dots,m$ and $j=1,\dots,n$.

- The operator $\mathcal{L}^T:\mathbb{R}^{m\times n}\to\mathbb{R}^{(m-1)\times n}\times\mathbb{R}^{m\times(n-1)}$ which is adjoint to $\mathcal{L}$ is given by

$$\mathcal{L}^T(\mathbf{x}) = (\mathbf{p},\mathbf{q}),$$

  where $\mathbf{p}\in\mathbb{R}^{(m-1)\times n}$ and $\mathbf{q}\in\mathbb{R}^{m\times(n-1)}$ are the matrices defined by

$$\begin{aligned} p_{i,j} &= x_{i,j}-x_{i+1,j}, \quad i=1,\dots,m-1, j=1,\dots,n,\\ q_{i,j} &= x_{i,j}-x_{i,j+1}, \quad i=1,\dots,m, j=1,\dots,n-1. \end{aligned}$$

- P_C is the orthogonal projection operator on the set C. Thus, for example, if $C = B_{l,u}$ then $P_{B_{l,u}}$ is explicitly given by

$$P_{B_{l,u}}(\mathbf{x})_{ij} = \begin{cases} l & x_{ij} < l, \\ x_{ij} & l \le x_{ij} \le u, \\ u & x_{ij} > u. \end{cases}$$

Equipped with the necessary notation, we are now ready to derive a dual problem of (4.1) and state the relation between the primal and dual optimal solutions. This was done in the unconstrained case in [5], and is easily extended to the constrained case via standard Lagrangian duality. For completeness we include a proof.

Proposition 4.1. *Let $(\mathbf{p}, \mathbf{q}) \in \mathcal{P}$ be the optimal solution of the problem*

$$\min_{(\mathbf{p},\mathbf{q})\in\mathcal{P}} \left\{ h(\mathbf{p},\mathbf{q}) \equiv -\|H_C(\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q}))\|_F^2 + \|\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q})\|_F^2 \right\}, \tag{4.2}$$

where

$$H_C(\mathbf{x}) = \mathbf{x} - P_C(\mathbf{x}) \quad \textit{for every } \mathbf{x} \in \mathbb{R}^{m\times n}. \tag{4.3}$$

Then the optimal solution of (4.1) with $TV = TV_I$ is given by

$$\mathbf{x} = P_C(\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q})). \tag{4.4}$$

Proof: First note that the relations

$$\begin{aligned} \sqrt{x^2+y^2} &= \max_{p_1,p_2}\{p_1x + p_2y : p_1^2 + p_2^2 \le 1\}, \\ |x| &= \max_{p}\{px : |p| \le 1\} \end{aligned}$$

hold true. Hence, we can write

$$\mathrm{TV}_I(\mathbf{x}) = \max_{(\mathbf{p},\mathbf{q})\in\mathcal{P}} T(\mathbf{x},\mathbf{p},\mathbf{q}),$$

where

$$\begin{aligned} T(\mathbf{x},\mathbf{p},\mathbf{q}) &= \textstyle\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}[p_{i,j}(x_{i,j} - x_{i+1,j}) + q_{i,j}(x_{i,j} - x_{i,j+1})] \\ &\quad + \textstyle\sum_{i=1}^{m-1} p_{i,n}(x_{i,n} - x_{i+1,n}) + \sum_{j=1}^{n-1} q_{m,j}(x_{m,j} - x_{m,j+1}). \end{aligned}$$

With this notation we have

$$T(\mathbf{x},\mathbf{p},\mathbf{q}) = \mathrm{Tr}(\mathcal{L}(\mathbf{p},\mathbf{q})^T\mathbf{x}).$$

The problem (4.1) therefore becomes

$$\min_{\mathbf{x}\in C}\max_{(\mathbf{p},\mathbf{q})\in\mathcal{P}} \left\{ \|\mathbf{x} - \mathbf{b}\|_F^2 + 2\lambda\mathrm{Tr}(\mathcal{L}(\mathbf{p},\mathbf{q})^T\mathbf{x}) \right\}.$$

Since the objective function is convex in $\mathbf{x}$ and concave in $\mathbf{p}, \mathbf{q}$, we can exchange the order of the minimum and maximum (see for example [23, Corollary 37.3.2]) and get

$$\max_{(\mathbf{p},\mathbf{q})\in\mathcal{P}}\min_{\mathbf{x}\in C} \left\{ \|\mathbf{x} - \mathbf{b}\|_F^2 + 2\lambda\mathrm{Tr}(\mathcal{L}(\mathbf{p},\mathbf{q})^T\mathbf{x}) \right\},$$

which can be rewritten as

$$\max_{(\mathbf{p},\mathbf{q})\in\mathcal{P}} \min_{\mathbf{x}\in C} \left\{ \|\mathbf{x} - (\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q}))\|_F^2 - \|\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q})\|_F^2 + \|\mathbf{b}\|_F^2 \right\}. \tag{4.5}$$

The optimal solution of the inner minimization problem is

$$\mathbf{x} = P_C(\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q})), \tag{4.6}$$

where we recall that $P_C(\cdot)$ is the orthogonal projection operator onto the set C. Plugging the above expression for $\mathbf{x}$ back into (4.5), and omitting constant terms, we obtain that the dual problem is the same as

$$\max_{(\mathbf{p},\mathbf{q})\in\mathcal{P}} \left\{ \|P_C(\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q})) - (\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q}))\|_F^2 - \|\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q})\|_F^2 \right\},$$

which is equivalent to (4.2) with H as defined in (4.3). □

Note that in the unconstrained case $C = \mathbb{R}^{m\times n}$, problem (4.2) is equivalent to the dual problem derived in [5].

Remark 4.1. The only difference in the dual problem corresponding to the case $\mathrm{TV} = \mathrm{TV}_{l_1}$ (in comparison to the case $\mathrm{TV} = \mathrm{TV}_I$), is that the minimization in the dual problem is not done over the set $\mathcal{P}$, but rather over the set $\mathcal{P}_1$ which consists of all pairs of matrices $(\mathbf{p},\mathbf{q})$ where $\mathbf{p} \in \mathbb{R}^{(m-1)\times n}$ and $\mathbf{q} \in \mathbb{R}^{m\times(n-1)}$ satisfying

$$\begin{aligned} |p_{i,j}| &\leq 1, i = 1,\ldots,m-1, j = 1,\ldots,n, \\ |q_{i,j}| &\leq 1, i = 1,\ldots,m, j = 1,\ldots,n-1. \end{aligned}$$

To be able to employ gradient type methods on the dual problem (4.2), it is important to note that the objective function of (4.2) is continuously differentiable, a property which thanks to Lemma 3.1 is established in the following result.

Lemma 4.1. *The objective function h of (4.2) is continuously differentiable and its gradient is given by*

$$\nabla h(\mathbf{p},\mathbf{q}) = -2\lambda\mathcal{L}^T P_C(\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q})). \tag{4.7}$$

Proof: Consider the function $s : \mathbb{R}^{m\times n} \to \mathbb{R}$ defined by

$$s(\mathbf{x}) = \|H_C(\mathbf{x})\|_F^2,$$

with H_C being defined in (4.3). Then, the dual function in (4.2) reads:

$$h(\mathbf{p},\mathbf{q}) = -s(\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q})) + \|\mathbf{b} - \lambda\mathcal{L}(\mathbf{p},\mathbf{q})\|_F^2.$$

Invoking Lemma 3.1(c), one has that $s(\cdot)$ continuously differentiable and its gradient is given by

$$\nabla s(\mathbf{x}) = 2(\mathbf{x} - P_C(\mathbf{x})).$$

Therefore,

$$\begin{aligned}\nabla h(\mathbf{p},\mathbf{q}) &= \nabla\left(-s(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q}))+\|\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q})\|_F^2\right)\\ &= \lambda\mathcal{L}^T\nabla s(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q}))-2\lambda\mathcal{L}^T(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q}))\\ &= 2\lambda\mathcal{L}^T(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q}))-2\lambda\mathcal{L}^TP_C(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q}))-2\lambda\mathcal{L}^T(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q}))\\ &= -2\lambda\mathcal{L}^TP_C(\mathbf{b}-\lambda\mathcal{L}(\mathbf{p},\mathbf{q})).\end{aligned}$$

□

Combining Proposition 4.1 and Lemma 4.1, we conclude that the dual problem (4.2) is a continuously differentiable convex minimization problem with a very simple constraint set. It allows us to consider simple first order gradient-based methods discussed earlier. In the next section we consider the two gradient-based methods GP and FGP when applied to the dual problem (4.2).

4.2 A Fast Denoising Method

Our objective is to solve the dual problem (4.2) whose gradient is given in Lemma 4.1. We first describe a gradient projection method for the denoising problem. We then apply the corresponding faster method FGP which has been discussed in Section 3.4.

Recall that the norm of a matrix $\mathbf{x}\in\mathbb{R}^{m\times n}$ is the Frobenius norm and for the pair $(\mathbf{p},\mathbf{q})\in\mathbb{R}^{(m-1)\times n}\times\mathbb{R}^{m\times(n-1)}$, the norm is

$$\|(\mathbf{p},\mathbf{q})\|\equiv\sqrt{\|\mathbf{p}\|_F^2+\|\mathbf{q}\|_F^2}.$$

To invoke any of the two methods (ISTA,FISTA) discussed in the previous section, we need to compute an upper bound on the Lipschitz constant of the gradient objective function of (4.2). This is done in the next lemma.

Lemma 4.2. *Let $L(h)$ be the Lipschitz constant of the gradient of the objective function h given in (4.2). Then,*

$$L(h)\le 16\lambda^2. \tag{4.8}$$

Proof: For every two pairs of matrices $(\mathbf{p}_1,\mathbf{q}_1),(\mathbf{p}_2,\mathbf{q}_2)$ where $\mathbf{p}_i\in\mathbb{R}^{(m-1)\times n}$ and $\mathbf{q}_i\in\mathbb{R}^{m\times(n-1)}$ for $i=1,2$, we have

$$\begin{aligned}\|\nabla h(\mathbf{p}_1,\mathbf{q}_1)-\nabla h(\mathbf{p}_2,\mathbf{q}_2)\| &\overset{(4.7)}{=} 2\lambda\|\mathcal{L}^T\left\{P_C[\mathbf{b}-\lambda\mathcal{L}(\mathbf{p}_1,\mathbf{q}_1)]\right\}-\mathcal{L}^T\left\{P_C[\mathbf{b}-\lambda\mathcal{L}(\mathbf{p}_2,\mathbf{q}_2)]\right\}\|\\ &\le 2\lambda\|\mathcal{L}^T\|\|P_C[\mathbf{b}-\lambda\mathcal{L}(\mathbf{p}_1,\mathbf{q}_1)]-P_C[\mathbf{b}-\lambda\mathcal{L}(\mathbf{p}_2,\mathbf{q}_2)]\|\\ &\overset{(*)}{\le} 2\lambda^2\|\mathcal{L}^T\|\cdot\|\mathcal{L}(\mathbf{p}_1,\mathbf{q}_1)-\mathcal{L}(\mathbf{p}_2,\mathbf{q}_2)\|\\ &\le 2\lambda^2\|\mathcal{L}^T\|\cdot\|\mathcal{L}\|\cdot\|(\mathbf{p}_1,\mathbf{q}_1)-(\mathbf{p}_2,\mathbf{q}_2)\|\\ &= 2\lambda^2\|\mathcal{L}^T\|^2\cdot\|(\mathbf{p}_1,\mathbf{q}_1)-(\mathbf{p}_2,\mathbf{q}_2)\|,\end{aligned}$$

where the transition (*) follows from the non-expensiveness property of the orthogonal projection operator ([12]). Now,

$$\begin{aligned}\|\mathcal{L}^T(\mathbf{x})\|^2 &= \sum_{i=1}^{m-1}\sum_{j=1}^{n}(x_{i,j}-x_{i+1,j})^2+\sum_{i=1}^{m}\sum_{j=1}^{n-1}(x_{i,j}-x_{i,j+1})^2\\ &\leq 2\sum_{i=1}^{m-1}\sum_{j=1}^{n}(x_{i,j}^2+x_{i+1,j}^2)+2\sum_{i=1}^{m}\sum_{j=1}^{n-1}(x_{i,j}^2+x_{i,j+1}^2)\\ &\leq 8\sum_{i=1}^{n}\sum_{j=1}^{m}x_{i,j}^2,\end{aligned}$$

where the last transition follows from the fact that for every $i=1,\ldots,m$ and $j=1,\ldots,n$ the term $x_{i,j}^2$ appears at most 4 times at the summation. Therefore,

$$\|\mathcal{L}^T(\mathbf{x})\|\leq\sqrt{8}\|\mathbf{x}\|,$$

implying that $\|\mathcal{L}^T\|\leq\sqrt{8}$ and hence $L(h)\leq 2\lambda^2\|\mathcal{L}^T\|^2\leq 16\lambda^2$.□

Plugging the expressions for the objective function and gradient along with the upper bound on the Lipschitz constant (4.8) into the gradient projection method described in Section 3, we obtain the following algorithm for the constrained denoising problem:

Algorithm GP($\mathbf{b},\lambda,N$)
Input:
$\mathbf{b}$ - observed image.
λ - regularization parameter.
N - Number of iterations.
Output:
$\mathbf{x}^*$ - An optimal solution of (4.1) (up to a tolerance).

Step 0. Take $(\mathbf{p}_0,\mathbf{q}_0)=(\mathbf{0}_{(m-1)\times n},\mathbf{0}_{m\times(n-1)})$.
Step k. $(k=1,2,\ldots,N)$ Compute

$$(\mathbf{p}_k,\mathbf{q}_k)=P_{\mathcal{P}}\left[(\mathbf{p}_{k-1},\mathbf{q}_{k-1})+\frac{1}{8\lambda}\mathcal{L}^T\left(P_C[\mathbf{b}-\lambda\mathcal{L}(\mathbf{p}_{k-1},\mathbf{q}_{k-1})]\right)\right],$$

Set $\mathbf{x}^*=P_C[\mathbf{b}-\lambda\mathcal{L}(\mathbf{p}_N,\mathbf{q}_N)]$

The projection onto the set $\mathcal{P}$ can be simply computed. For a pair $(\mathbf{p},\mathbf{q})$, the projection $P_{\mathcal{P}}(\mathbf{p},\mathbf{q})$ is given by $P_{\mathcal{P}}(\mathbf{p},\mathbf{q})=(\mathbf{r},\mathbf{s})$ where $\mathbf{r}\in\mathbb{R}^{(m-1)\times n}$ and $\mathbf{s}\in\mathbb{R}^{m\times(n-1)}$ are given by

$$r_{ij}=\begin{cases}\frac{p_{ij}}{\max\{1,\sqrt{p_{ij}^2+q_{ij}^2}\}} & i=1,\ldots,m-1,j=1,\ldots,n-1,\\ \frac{p_{in}}{\max\{1,|p_{in}|\}} & i=1,\ldots,m-1\end{cases}$$

and

$$s_{ij}=\begin{cases}\frac{q_{ij}}{\max\{1,\sqrt{p_{ij}^2+q_{ij}^2}\}} & i=1,\ldots,m-1,j=1,\ldots,n-1,\\ \frac{q_{mj}}{\max\{1,|q_{mj}|\}}. & j=1,\ldots,n-1\end{cases}$$

In the unconstrained case ($C = \mathbb{R}^{m\times n}$), the algorithm is the same as the method proposed by Chambolle [6] which, as explained in [6], seems to practically perform better than a slightly different gradient-based method originally proposed by the same author in [5].

Remark 4.2. When TV = TV_{l_1} the dual problem is defined over the set $\mathcal{P}_1$ as given in Remark 4.1. Therefore, the only difference in the gradient projection algorithm will be in the projection step. For a pair $(\mathbf{p}, \mathbf{q})$, the projection $P_{\mathcal{P}_1}(\mathbf{p}, \mathbf{q})$ is given by $P_{\mathcal{P}_1}(\mathbf{p}, \mathbf{q}) = (\mathbf{r}, \mathbf{s})$ where the components of $\mathbf{r} \in \mathbb{R}^{(m-1)\times n}$ and $\mathbf{s} \in \mathbb{R}^{m\times(n-1)}$ are given by

$$r_{ij} = \frac{p_{ij}}{\max\{1, |p_{i,j}|\}} \quad \text{and} \quad s_{ij} = \frac{q_{ij}}{\max\{1, |q_{i,j}|\}}.$$

Now, as shown in Section 3.4, we can use the fast gradient projection (FGP) on the dual problem, which warrant the better theoretical rate of convergence $O(1/k^2)$, as opposed to the GP method which is only of the order $O(1/k)$. When applied to the dual problem (4.2), the algorithm reads explicitly as follows.

Algorithm FGP$(\mathbf{b}, \lambda, N)$
Input:
$\mathbf{b}$ - observed image.
λ - regularization parameter.
N - Number of iterations.
Output:
$\mathbf{x}^*$ - An optimal solution of (4.1) (up to a tolerance).

Step 0. Take $(\mathbf{r}_1, \mathbf{s}_1) = (\mathbf{p}_0, \mathbf{q}_0) = (\mathbf{0}_{(m-1)\times n}, \mathbf{0}_{m\times(n-1)}), t_1 = 1$.
Step k. $(k = 1, \ldots, N)$ Compute

$$(\mathbf{p}_k, \mathbf{q}_k) = P_{\mathcal{P}}\left[(\mathbf{r}_k, \mathbf{s}_k) + \frac{1}{8\lambda}\mathcal{L}^T\left(P_C[\mathbf{b} - \lambda\mathcal{L}(\mathbf{r}_k, \mathbf{s}_k)]\right)\right], \tag{4.9}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \tag{4.10}$$

$$(\mathbf{r}_{k+1}, \mathbf{s}_{k+1}) = (\mathbf{p}_k, \mathbf{q}_k) + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{p}_k - \mathbf{p}_{k-1}, \mathbf{q}_k - \mathbf{q}_{k-1}). \tag{4.11}$$

Set $\mathbf{x}^* = P_C[\mathbf{b} - \lambda\mathcal{L}(\mathbf{p}_N, \mathbf{q}_N)]$

4.3 Numerical Examples I

We have tested the two methods GP and FGP on numerous examples and in all examples we noticed that FGP is much faster than GP. Here we give two examples. In the first example we took the 256×256 cameraman test image whose pixels were scaled to be between to be between 0 and 1. We then added to each pixel noise which is normally distributed with zero mean and standard deviation 0.1. The main problem (4.1) was then solved with $\lambda = 0.1$ with both GP and FGP methods. The first 30 function values of the two methods can be seen in Figure 1. Clearly FGP outperforms GP in this example.

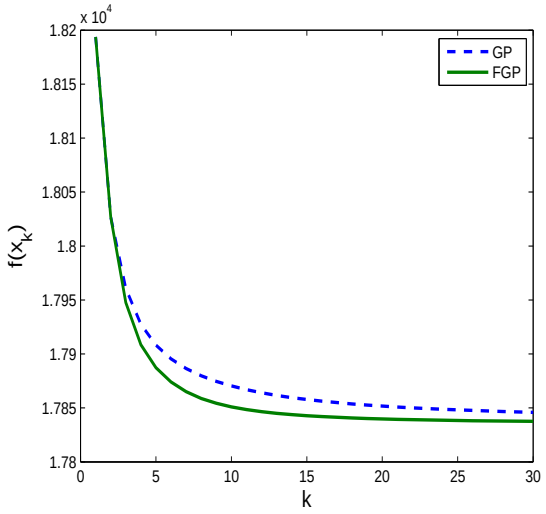


Figure 1: Function values at the first 30 iterations of GP and FGP employed on a 256×256 denoising problem

To better understand the advantage of FGP over GP, we also explored the accuracy obtained by the two methods. To do so, we have taken a small 10×10 image (which was actually the left up corner of the cameraman test image) and like in the previous example we added to the image normally distributed white noise with standard deviation 0.1. The parameter λ was again chosen as 0.1. Since the problem is small we were able to find an exact solution using SeDuMi [25]. Figure 2 shows the the difference $F(\mathbf{x}_k) - F_*$ (in log scale) for $k = 1, \ldots, 100$.

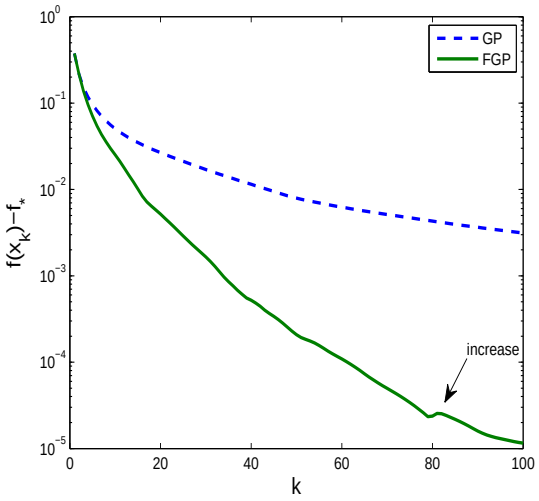


Figure 2: Accuracy of FGP compared with GP

Clearly, FGP reaches greater accuracies than those obtain by GP. After 100 iterations FGP reached an accuracy of 10^{-5} while GP reached an accuracy of only $10^{-2.5}$. Moreover, the function value reached by GP at iteration 100 was already obtained by GP after 25 iterations. Later on, in Section 5.1 we will see further evidence for the fast rate of convergence of FGP.

Figure 3: Left: the original moon image. Right: the noisy moon image

Another interesting phenomena can be seen at iterations 81 and 82 and is marked on the figure. As opposed to the GP method, FGP is not a monotone method. This does not have an influence on the convergence of the sequence and we see that in most iterations there is a decrease in the function value. However, later on we will see that this phenomena can have a severe impact on the convergence of a related two-steps method for the image deblurring problem.

We end this section with a visual and representative demonstration of the capabilities of the FGP algorithm. Consider the 537×358 moon test image[2] whose pixels are scaled to be between zero and one. A white Gaussian noise with standard deviation 0.08 was added to the image. The original and noisy images are given in Figure 3.

The PSNR[3] of the noisy image is 17.24 dB. We employed 20 iterations of the FGP method with regularization parameter $\lambda = 0.07$. The resulting denoised image is given in Figure 4. The PSNR of the denoised image is 29.93dB.

5 Total Variation-Based Deblurring

Consider now the constrained TV-based deblurring optimization model introduced in Section 2:

$$\min_{\mathbf{x}\in C} \|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|_F^2 + 2\lambda \mathrm{TV}(\mathbf{x}), \tag{5.1}$$

where $\mathbf{x} \in \mathbb{R}^{m\times n}$ is the original image to be restored, $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ is a linear transformation representing some blurring operator, $\mathbf{b}$ is the noisy and blurred image, and $\lambda > 0$ is a regularization parameter. Obviously problem (5.1) is within the setting of the general

[2]Part of the image processing toolbox of MATLAB.

[3]peak signal-to-noise ratio.

Figure 4: Denoised moon image with FGP.

model (3.1) with

$$f(\mathbf{x}) \equiv \|\mathcal{A}(\mathbf{x}) - \mathbf{b}\|^2, \quad g(\mathbf{x}) \equiv 2\lambda \mathrm{TV}(\mathbf{x}) + \delta_C(\mathbf{x}), \text{ and } \mathbb{E} = \mathbb{R}^{m\times n}.$$

Deblurring is of course more challenging than denoising. Indeed, to construct an equivalent smooth optimization problem for (5.1) via its dual along the approach of Section 4.1, it is easy to realize that one would need to invert the operator $\mathcal{A}^T\mathcal{A}$, which is clearly an ill-posed problem, i.e., such an approach is not viable. This is in sharp contrast to the denoising problem, where a smooth dual problem was constructed, and was the basis of efficient solution methods. To avoid this difficulty, we treat the TV deblurring problem (5.1) in two steps through the denoising problem. More precisely, first denote the optimal solution of the constrained *denoising* problem (4.1) with observed image $\mathbf{b}$, regularization parameter λ and feasible set C by $D_C(\mathbf{b}, \lambda)$. Then, with this notation $p_L(\mathbf{Y})$ (cf. 3.13), can be simply written as:

$$p_L(\mathbf{Y}) = D_C\left(\mathbf{Y} - \frac{2}{L}\mathcal{A}^T(\mathcal{A}(\mathbf{Y}) - \mathbf{b}), \frac{2\lambda}{L}\right).$$

We re-emphasize the fact that the TV-based regularization problem for deblurring requires at each iteration the solution of a denoising problem of the form (4.1) in addition to the gradient step. Thus, each iteration involves the solution of a subproblem that should be solved using an iterative method such as GP or FGP as described in Section 3. Note that this is in contrast to the situation with the simpler l_1 regularization problem where ISTA or FISTA require only the computation of a gradient step and a shrinkage, which in that case is an *explicit* operation, see (3.11) and (3.12). Finally, it interesting to note that for anisotropic TV the denoising problem can also be solved by using parametric maximal flow algorithm, see [16] for details.

5.1 FISTA Revisited: A Monotone Version

As opposed to ISTA, FISTA is not a monotone algorithm, that is, the function values are not guaranteed to be nonincreasing. Monotonicity seems to be a desirable property of minimization algorithms, but it is not required in the original proof of convergence of FISTA [1]. Moreover, we observed in the numerical simulations in [1] that the algorithm is "almost monotone", that is, except for very few iterations the algorithm exhibits a monotonicity property.

In our case, where the denoising subproblems are not solved exactly, monotonicity becomes an important issue. It might happen that due to the inexact computations of the denoising subproblems, the algorithm might become extremely non-monotone and in fact can even diverge! This is illustrated in the following numerical example.

Example 5.1. Consider a 64×64 image that was cut from the cameraman test image (whose pixels are scaled to be between 0 and 1). The image goes through a Gaussian blur of size 9×9 and standard deviation 4 (applied by the MATLAB functions imfilter and fspecial) followed by a an additive zero-mean white Gaussian noise with standard deviation 10^{-2}. We first chose the regularization parameter λ to be 10^{-3} and ran the FISTA method applied to (5.1) in which the denoising subproblems are solved using FGP with N (number of FGP iterations) taking the values $5, 10, 20$. We noticed that the the three runs result with virtually the same function values, leading to the conclusion that in this example the FGP method reaches a "good enough" solution after only 5 iterations so that there is no need to make additional inner iterations.

On the other hand, if we consider the same deblurring problem with a larger λ, specifically, $\lambda = 0.01$, the situation is completely different. The three graphs corresponding to $N = 5, 10, 20$ are presented in Figure 5. In the left image the denoising subproblems are solved using FGP and in the right image the denoising subproblems are solved using GP. Clearly FISTA in combination with either GP or FGP diverges when $N = 5$, although it seems that the combination FISTA/GP is worse than FISTA/FGP. For $N = 10$ FISTA/FGP seems to converge to a value which is a bit higher than the one obtained by the same method with $N = 20$ and FISTA/GP with $N = 10$ is still very much erratic and does not seem to converge.

From the previous example we can conclude that (1) FISTA can diverge when the subproblems are not solved exactly and (2) the combination FISTA/FGP seems to be better than FISTA/GP. The latter conclusion is another numerical evidence (in addition to the results of Section 4.3) to the superiority of FGP over GP. The first conclusion motivates us to explore and introduce a monotone version of FISTA. The monotone version of FISTA we proposed is as follows, and the algorithm is termed MFISTA (for monotone FISTA).

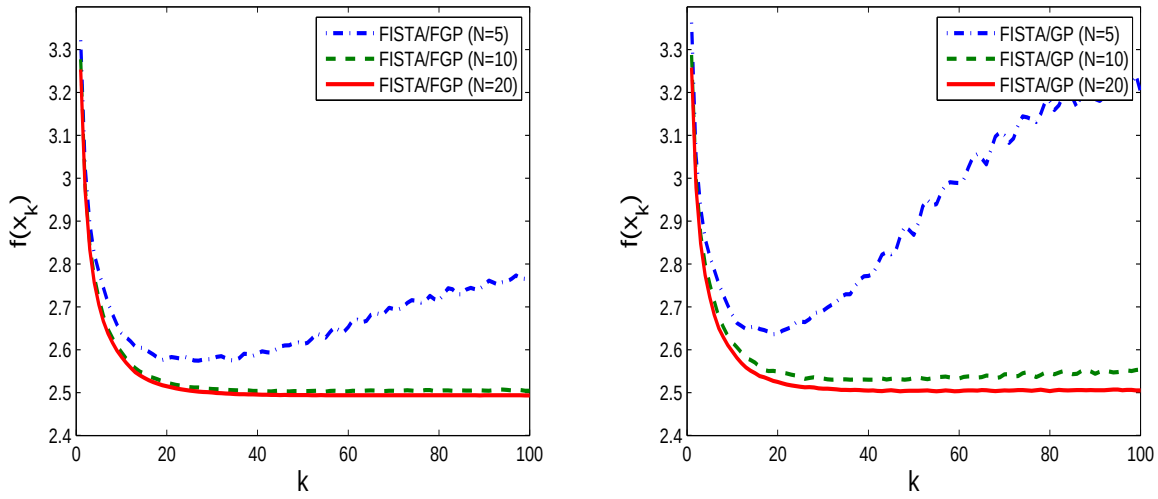


Figure 5: Function values of the first 100 iterations of FISTA. The denoising subproblems are solved using FGP (left image) or GP (right image) with $N = 5, 10, 20$.

MFISTA

Input: $L \geq L(f)$ - An upper bound on the Lipschitz constant of ∇f.

Step 0. Take $\mathbf{y}_1 = \mathbf{x}_0 \in \mathbb{E}$, $t_1 = 1$.

Step k. $(k \geq 1)$ Compute

$$\begin{aligned} \mathbf{z}_k &= p_L(\mathbf{y}_k), \\ t_{k+1} &= \frac{1+\sqrt{1+4t_k^2}}{2}, \qquad (5.2) \\ \mathbf{x}_k &= \operatorname{argmin}\{F(\mathbf{x}) : \ \mathbf{x} = \mathbf{z}_k, \mathbf{x}_{k-1}\} \qquad (5.3) \\ \mathbf{y}_{k+1} &= \mathbf{x}_k + \left(\frac{t_k}{t_{k+1}}\right)(\mathbf{z}_k - \mathbf{x}_k) + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}_k - \mathbf{x}_{k-1}). \qquad (5.4) \end{aligned}$$

MFISTA requires in addition to the computation of the proximal map, the computation of a single function evaluation: $F(\mathbf{z}_k)$ (the second function value $F(\mathbf{x}_{k-1})$ was already computed in the previous iterate). FISTA, on the other hand, does not require this additional function evaluation although a typical implementation of FISTA will comprise a function evaluation $F(\mathbf{x}_k)$ in order to monitor the progress of the algorithm (although, again, this is not mandatory).

It turns out that this modification does not affect the theoretical rate of convergence. Indeed, the convergence rate result for MFISTA will remain the same as the convergence rate result for FISTA:

Theorem 5.1. *Let $\{\mathbf{x}_k\}$ be generated by MFISTA. Then for any $k \geq 1$*

$$F(\mathbf{x}_k) - F(\mathbf{x}^*) \leq \frac{2L(f)\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{(k+1)^2}, \quad \forall \mathbf{x}^* \in X_*.$$

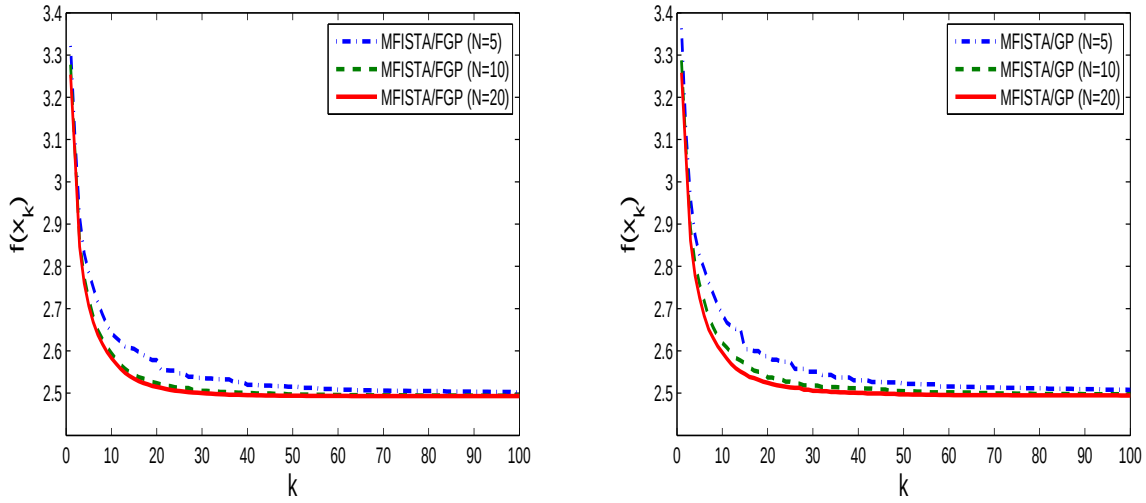


Figure 6: Function values of the first 100 iterations of MFISTA. The denoising subproblems are solved using either FGP (left image) or GP (right image) with $N = 5, 10, 20$.

The proof of this theorem requires modifications in the proof of FISTA given in [1]. This is developed in the Appendix.

Coming back to Example 5.1, we ran MFISTA on the exact same problem and the results are shown in Figure 6. Clearly the monotone version of FISTA seems much more robust and stable. Therefore, we suggest using MFISTA instead of FISTA.

6 Numerical Examples II

The purpose of this section is to demonstrate the effectiveness, simplicity and potential of MFISTA on TV-based deblurring problems. We compare it to the classical ISTA and to the recent algorithm TwIST (and its monotone version MTwIST) of [3] which, like MFISTA, is an algorithm that is also based on information of the two previous iterations. Clearly, there exists many other deblurring algorithms which are not discussed or/and compared here. A more thorough computational study and comparison is beyond the scope of this work and is left for future research. To facilitate such comparison, and as indicated in the introduction, a MATLAB code is available at

`http://iew3.technion.ac.il/~becka/papers/tv_fista.zip`

Here, our purpose is on showing that MFISTA can often reach fast rate of convergence even beyond the theoretical rate we have established, and as such MFISTA appears as a promising algorithmic framework for image restoration problems. Both ISTA and MFISTA were implemented with a fixed step size $\frac{1}{L}$, where L is the Lipschitz constant.

All the images in this section were scaled so that their pixels will be in the range between 0 and 1.

6.1 Comparison of ISTA, MFISTA and TwIST

Consider the 256×256 Lena test image that goes through a 9×9 gaussian blur with standard deviation 4 followed by an additive normally distributed noise with mean zero and standard deviation 10^{-3}. The regularization parameter was chosen to be $\lambda = 10^{-4}$. This regularization parameter was hand tuned to give the best SNR improvement. We emphasize that although the quality of the obtained image (e.g., in terms of SNR improvement) depends on the choice λ, the rate of convergence of the method does not depend on the choice of regularization parameter. We use ISTA, MFISTA and MTWIST of [3] with 100 iterations on the blurred and noisy image, and we obtain the three reconstructions given in Figure 7. The reconstruction of TWIST is better (that is, sharper) than the one provided by ISTA, and the reconstruction of MFISTA is better than the outcomes of both ISTA and MTWIST (sharper and with less ringing affects). This is also reflected in the objective function values after which are 0.606 (ISTA), 0.502 (MTWIST) and 0.466(MFISTA) and in the PSNR values given in the following table:

Method	PSNR (dB)	CPU times (sec.)
ISTA	26.73	21.32
MTWIST	27.82	21.40
MFISTA	29.13	21.37

The table above also summarizes the CPU times of each of the three methods implemented in MATLAB on a Pentuim 4, 1.8 Ghz. Clearly, the computational time is more or less the same for all of these methods. The reason for this is that the dominant computation at each iteration of the three methods is the evaluation of one function value[4], one gradient and one denoising step.

It is also interesting to note that MTWIST which is the monotone version of TWIST was employed here since for deblurring problems which are highly ill-conditioned, the original TWIST converges very slowly. The parameters for the MTWIST method were chosen as suggested in Section 6 of [3] for extremely ill-conditioned problems.

The function values of the first 100 iterations are presented in Figure 8. MTWIST was better in the first 12 iterations, however MFISTA reaches lower functions values starting from iteration number 13.

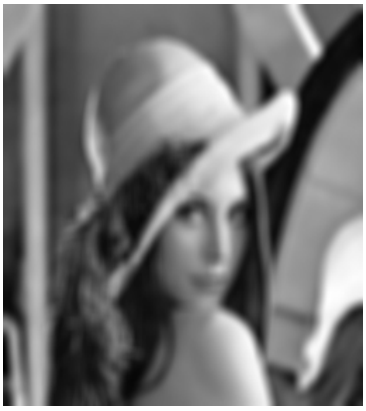
Figure 9 shows the evolution of the SNR improvement (ISNR) by MFISTA, MTWIST and ISTA. Clearly, MFISTA converges faster than ISTA and and MTWIST.

We have also compared the methods on a simple test image extracted from the function blur from the "regularization toolbox" [14]. The image goes through the same blur operator as in the previous example followed by an additive normally distributed noise with standard deviation $2 \cdot 10^{-2}$ which is 20 times larger than the standard deviation used in the "Lena" image. This results with the blurred and noisy image shown in Figure 10 (PSNR value of 23.43 dB).

We use ISTA, MFISTA and MTWIST of [3] with 100 iterations on the blurred and noisy image, and we obtain the three reconstructions given in Figure 11. The function value

[4]Although ISTA does not explicitly require the computation of the function value at each iteration, in our implementation this function value is evaluated at each step in order to monitor the progress of the algorithm.

Blurred and Noisy



ISTA($F_{100} = 0.606$)

MTWIST($F_{100} = 0.502$)

MFISTA($F_{100} = 0.466$)

Figure 7: The blurred and noisy Lena (top left image) along with the three reconstructions of ISTA (top right), TWIST (bottom left) and MFISTA (bottom right)

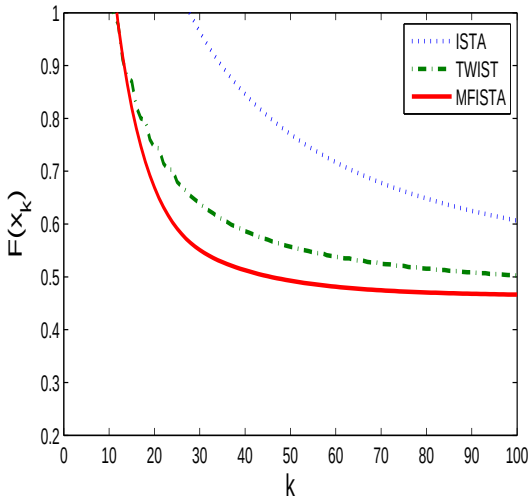


Figure 8: Function values of ISTA, MTWIST and MFISTA for the first 100 iterations

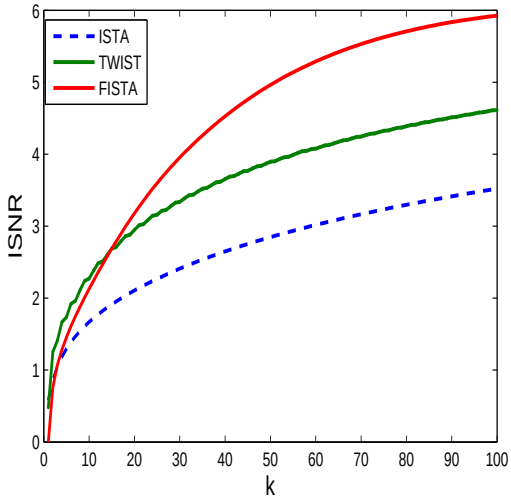


Figure 9: Evolution of the SNR improvement (ISNR) produced by ISTA, MTWIST and MFISTA for the first 100 iterations

Original Blurred and Noisy

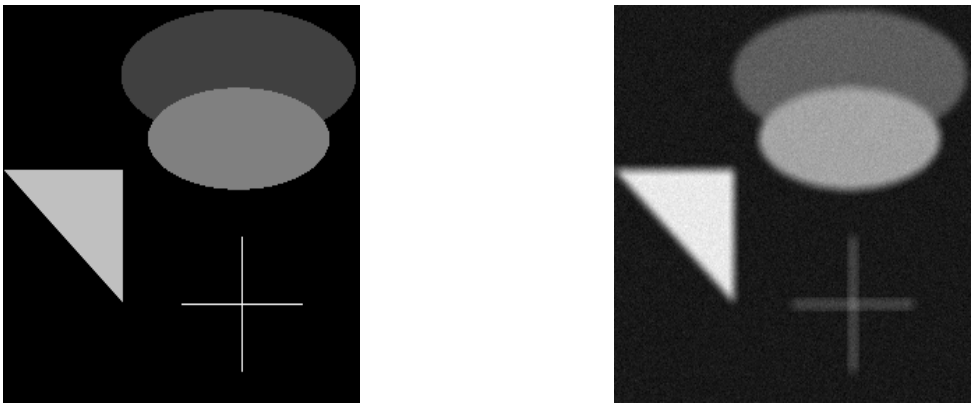


Figure 10: Left: the original image. Right: the blured and noisy image

MFISTA reaches is 14.89 which is lower than the function values obtained by ISTA and MTWIST (15.107 and 14.943 respectively). The PSNR value of the MFISTA reconstruction is 29.73 dB which is higher than the PSNR values of ISTA and MTWIST (26.26 dB and 27.67 dB respectively). Figure 11 also demonstrates the relatively rapid evolution of the SNR improvement made MFISTA in comparison to MTWIST and ISTA.

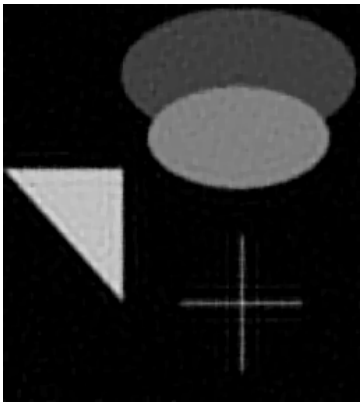
6.2 The Importance of Bound Constraints

In this example we demonstrate the importance of incorporating constraints in the main deblurring problem (5.1). Of course, bound constraints can be imposed only if such information on the original image is available. We will show that bound constraints on the pixels can have a strong effect on the quality of the reconstruction. The constraints have a visible effect if the image has relatively a lot of extreme pixels (that is, black and white). To illustrate the importance of constraints, we consider the 256×256 image "text" from the image processing toolbox of MATLAB. This image is very extreme in the sense that it contains only black and white pixels. The image was blurred using the same blurring operation as in the previous example followed by a normally distributed noise with mean zero and standard deviation 0.02. The regularization parameter was chosen to be $4 \cdot 10^{-4}$. The original and the blurred and noisy images are the top images in Figure 12. The bottom left image is the output of MFISTA without constraints ($C = \mathbb{R}^{256\times256}$) while the bottom right image is the output of MFISTA after 100 iterations with $0, 1$ bound constraints. That is,

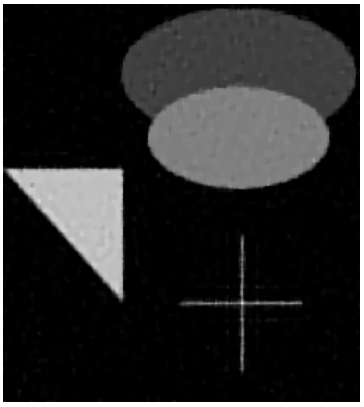
$$C = B_{0,1} = \left\{\mathbf{x} \in \mathbb{R}^{256\times256} : 0 \leq x_{ij} \leq 1, i, j = 1, \ldots, 256\right\}.$$

Obviously the problem with bound constraints gives rise to a better quality deblurred image. In particular, the black zone in the left image is not as smooth as the black zone in

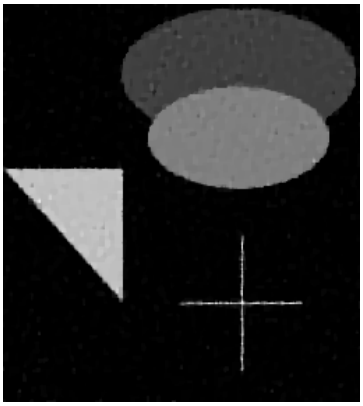
ISTA($F_{100} = 15.107$)



MTWIST($F_{100} = 14.943$)



FISTA($F_{100} = 14.890$)



ISNR

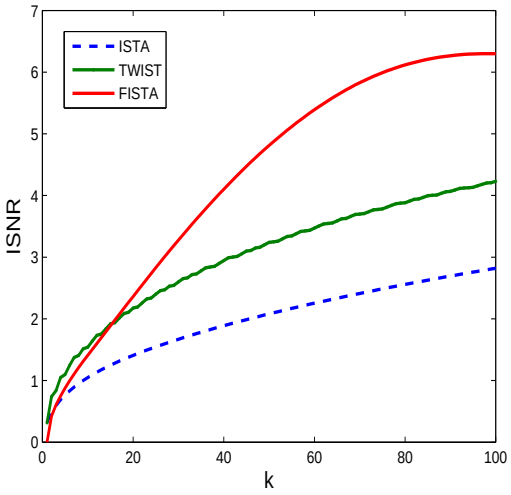


Figure 11: The reconstructions of ISTA (top left), TWIST (top right) and MFISTA (bottom left) along with the ISNR improvement of each method (bottom right)

the right image. Moreover, the PSNR of the left image is 18.06 dB while the PSNR of the right image is 20.27 dB.

7 Summary

We introduced new gradient-based schemes for the constrained total variation based image denoising and deblurring problems. Our framework is general enough to cover other types of nonsmooth regularizers. The proposed schemes keep the simplicity of first order methods, and within a dedicated analysis which relies on combining a dual approach with a fast gradient projection scheme, we have introduced monotone schemes that can be applied to TV-based deblurring problems and are proven to exhibit fast theoretical rate of convergence. Our initial numerical results confirm the predicted underlying theoretical results and demonstrate that within our framework one can devise algorithms that can perform significantly faster than some of currently known state of art gradient projected based methods. To facilitate such comparison, we have contributed a MATLAB code of the developed schemes. Further research within the proposed class of algorithms including other classes of regularizers and a thorough experimental/computational study remain to be investigated in the future.

A Appendix – Rate of Convergence of MFISTA

The problem setting is as defined in Section 3.1. That is, $F = f + g$ where f is a continuously differentiable function whose gradient satisfies a Lipschitz condition with constant L and g is a proper closed function. To establish the rate of convergence of MFISTA, we follow the analysis developed in [1]. For that, we need the following notation. For any $L > 0$, consider the quadratic approximation model for $F(\cdot)$ defined by,

$$Q_L(\mathbf{x}, \mathbf{y}) := f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \frac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2 + g(\mathbf{x}). \tag{A.1}$$

We now recall the following result from [1, Lemma 2.3].

Lemma A.1. *Let $\mathbf{y} \in \mathbb{E}$ and $L > 0$ be such that*

$$F(p_L(\mathbf{y})) \leq Q(p_L(\mathbf{y}), \mathbf{y}). \tag{A.2}$$

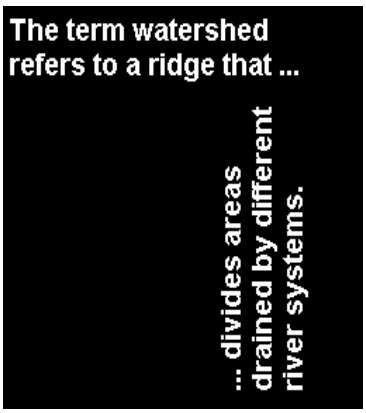
Then for any $\mathbf{x} \in \operatorname{dom} g$,

$$F(\mathbf{x}) - F(p_L(\mathbf{y})) \geq \frac{L}{2}\|p_L(\mathbf{y}) - \mathbf{y}\|^2 + L\langle \mathbf{y} - \mathbf{x}, p_L(\mathbf{y}) - \mathbf{y} \rangle.$$

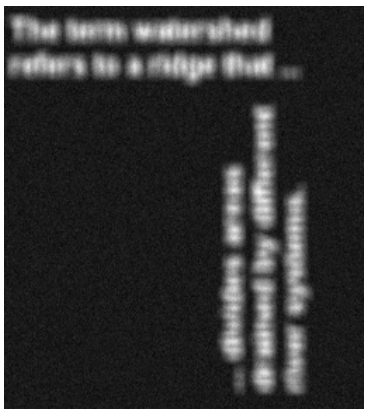
Note that by for $L \geq L(f)$ the condition (A.2) is always satisfied.

Now, the rate of convergence of MFISTA claimed in Theorem 5.1 will be established if we can prove a recursion inequality like the one derived for FISTA in [1, Lemma 4.1]. Indeed, with such a recursion at hand, the rate of convergence of MFISTA will follow mutatis-mutandis as derived for FISTA in [1, Theorem 4.1]. The next result established the required recursion.

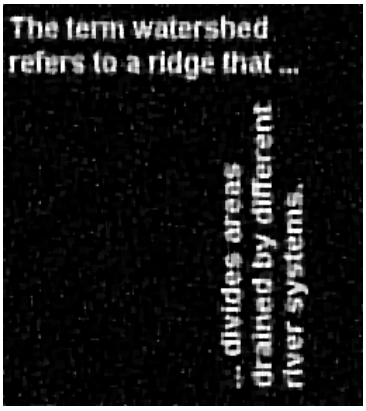
Original



Blurred and Noisy



MFISTA, $l = -\infty, u = \infty$



MFISTA, $l = 0, u = 1,$

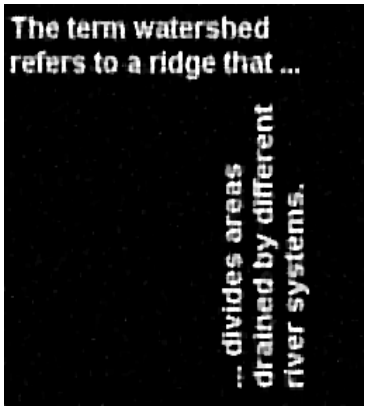


Figure 12: Top: original (left) and blurred (right) text image. Bottom: the left image is the output of MFISTA with 100 iterations without constraints. The right image is the output of MFISTA after 100 iteration with 0,1 bound constraints

Lemma A.2. *The sequences* $\{\mathbf{x}_k, \mathbf{y}_k, \mathbf{z}_k\}$ *generated by MFISTA satisfy:*

$$2L^{-1}\left[(t_k^2 v_k - t_{k+1}^2 v_{k+1}\right] \geq \|\mathbf{u}_{k+1}\|^2 - \|\mathbf{u}_k\|^2,$$

where $v_k := F(\mathbf{x}_k) - F(\mathbf{x}^*)$ *and* $\mathbf{u}_k := t_k \mathbf{z}_k - (t_k - 1)\mathbf{x}_{k-1} - \mathbf{x}^*$.

Proof. First, we apply Lemma A.1 at the points $(\mathbf{x} := \mathbf{x}_k \in \operatorname{dom} g, \mathbf{y} := \mathbf{y}_{k+1})$ and likewise at the points $(\mathbf{x} := \mathbf{x}^* \in \operatorname{dom} g, \mathbf{y} := \mathbf{y}_{k+1})$ to get:

$$\begin{aligned} 2L^{-1}(F(\mathbf{x}_k) - F(\mathbf{z}_{k+1})) &\geq \|\mathbf{z}_{k+1} - \mathbf{y}_{k+1}\|^2 + 2\langle \mathbf{z}_{k+1} - \mathbf{y}_{k+1}, \mathbf{y}_{k+1} - \mathbf{x}_k\rangle \\ 2L^{-1}(F(\mathbf{x}^*) - F(\mathbf{z}_{k+1})) &\geq \|\mathbf{z}_{k+1} - \mathbf{y}_{k+1}\|^2 + 2\langle \mathbf{z}_{k+1} - \mathbf{y}_{k+1}, \mathbf{y}_{k+1} - \mathbf{x}^*\rangle, \end{aligned}$$

where by the algorithm's definition we used $\mathbf{z}_{k+1} = p_L(\mathbf{y}_{k+1}) \in \operatorname{dom} g$. Multiplying the first inequality above by $(t_{k+1} - 1)$ and adding it to the second inequality we then get:

$$2L^{-1}((t_{k+1}-1)v_k - t_{k+1}(F(\mathbf{z}_{k+1}) - F(\mathbf{x}^*)) \geq t_{k+1}\|\mathbf{z}_{k+1} - \mathbf{y}_{k+1}\|^2 + 2\langle \mathbf{z}_{k+1} - \mathbf{y}_{k+1}, t_{k+1}\mathbf{y}_{k+1} - (t_{k+1}-1)\mathbf{x}_k - \mathbf{x}^*\rangle.$$

Now, by step (5.3) of MFISTA, we have $F(\mathbf{x}_{k+1}) \leq F(\mathbf{z}_{k+1})$. Therefore,

$$-t_{k+1}v_{k+1} = -t_{k+1}(F(\mathbf{x}_{k+1}) - F(\mathbf{x}^*)) \geq -t_{k+1}(F(\mathbf{z}_{k+1}) - F(\mathbf{x}^*))$$

and hence together with the last inequality we obtain:

$$2L^{-1}((t_{k+1}-1)v_k - t_{k+1}v_{k+1}) \geq t_{k+1}\|\mathbf{z}_{k+1} - \mathbf{y}_{k+1}\|^2 + 2\langle \mathbf{z}_{k+1} - \mathbf{y}_{k+1}, t_{k+1}\mathbf{y}_{k+1} - (t_{k+1}-1)\mathbf{x}_k - \mathbf{x}^*\rangle.$$

Multiplying the last inequality by t_{k+1}, and using the relation $t_k^2 = t_{k+1}^2 - t_{k+1}$ which holds thanks to (5.2) we obtain,

$$2L^{-1}(t_k^2 v_k - t_{k+1}^2 v_{k+1}) \geq \|t_{k+1}(\mathbf{z}_{k+1} - \mathbf{y}_{k+1})\|^2 + 2t_{k+1}\langle \mathbf{z}_{k+1} - \mathbf{y}_{k+1}, t_{k+1}\mathbf{y}_{k+1} - (t_{k+1}-1)\mathbf{x}_k - \mathbf{x}^*\rangle.$$

Now applying the usual Pythagoras relation: $\|b-a\|^2 + 2\langle b-a, a-c\rangle = \|b-c\|^2 - \|a-c\|^2$ to the right-handside of the last inequality with

$$a := t_{k+1}\mathbf{y}_{k+1}, \quad b := t_{k+1}\mathbf{z}_{k+1}, \quad c := (t_{k+1}-1)\mathbf{x}_k + \mathbf{x}^*,$$

we thus obtain:

$$2L^{-1}(t_k^2 v_k - t_{k+1}^2 v_{k+1}) \geq \|t_{k+1}\mathbf{z}_{k+1} - (t_{k+1}-1)\mathbf{x}_k - \mathbf{x}^*\|^2 - \|t_{k+1}\mathbf{y}_{k+1} - (t_{k+1}-1)\mathbf{x}_k - \mathbf{x}^*\|^2. \quad \text{(A.3)}$$

Now, recalling the definition of $\mathbf{y}_{k+1}$ given by (5.4) in MFISTA,

$$\mathbf{y}_{k+1} = \mathbf{x}_k + \left(\frac{t_k}{t_{k+1}}\right)(\mathbf{z}_k - \mathbf{x}_k) + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{x}_k - \mathbf{x}_{k-1})$$

one get $t_{k+1}\mathbf{y}_{k+1} = (t_{k+1}-1)\mathbf{x}_k + t_k\mathbf{z}_k - (t_k - 1)\mathbf{x}_{k-1}$, and hence with $\mathbf{u}_k := t_k\mathbf{z}_k - (t_k - 1)\mathbf{x}_{k-1} - \mathbf{x}^*$, the desired result follows from (A.3). □

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. 2008. Accepted for publication in *SIAM Journal on Imaging Sciences* (2008).

[2] D. P. Bertsekas. *Nonlinear Programming*. Belmont MA: Athena Scientific, second edition, 1999.

[3] J. Bioucas-Dias and M. Figueiredo. A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Trans. on Image Processing*, 16:2992–3004, 2007.

[4] A. Björck. *Numerical Methods for Least-Squares Problems*. Philadelphia, PA: SIAM, 1996.

[5] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20(1-2):89–97, 2004. Special issue on mathematics and image analysis.

[6] A. Chambolle. Total variation minimization and a class of binary MRF models. In *Lecture Notes in Computer Sciences*, volume 3757, pages 136–152, 2005.

[7] A. Chambolle and P.L. Lions. Image recovery via total variation minimization and related problmes. *Numerische Mathematik*, 76:167–188, 1997.

[8] T. F Chan, G.H. Golub, and P. Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM J. Sci. Comput.*, 20(6):1964–1977 (electronic), 1999.

[9] P. Combettes and V. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4:1168–1200, 2005.

[10] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*, 57(11):1413–1457, 2004.

[11] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1996.

[12] F. Facchinei and J. S. Pang. *Finite-dimensional variational inequalities and complementarity problems, Vol. II*. Springer Series in Operations Research. Springer-Verlag, New York, 2003.

[13] D. Goldfarb and W. Yin. Second-order cone programming methods for total variation-based image restoration. *SIAM Journal on Scientific Computing*, pages 622–645, 2005.

[14] P. C. Hansen. Regularization tools, a matlab package for analysis of discrete regularization problems. *Numerical Algorithms*, 6:1–35, 1994.

[15] M. Hintermuller and G. Stadler. An infeasible primal-dual algorithm for tv-based infconvolution-type image restoration. *SIAM Journal on Scientific Computing*, 28:1–23, 2006.

[16] D. S. Hochbaum. An efficient algorithm for image segmentation, Markov random fields and related problems. *J. ACM*, 48(4):686–701 (electronic), 2001.

[17] B. Martinet. Regularisation d'inéquations variationnelles par approximations successives. *Revue Francaise d'Automatique et Informatique Recherche Opérationnelle*, 4:154–159, 1970.

[18] J. J. Moreau. Proximitéet dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965.

[19] Y. E. Nesterov. Gradient methods for minimizing composite objective function. Technical report, CORE, September 2007. http://www.ecore.beDPs/dp1191313936.pdf.

[20] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269(3):543–547, 1983.

[21] G. B. Passty. Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *J. Math. Anal. Appl.*, 72(2):383–390, 1979.

[22] B. T. Polyak. *Introduction to optimization*. Translations Series in Mathematics and Engineering. Optimization Software Inc. Publications Division, New York, 1987. Translated from the Russian, With a foreword by Dimitri P. Bertsekas.

[23] R. T. Rockafellar. *Convex Analysis*. Princeton NJ: Princeton Univ. Press, 1970.

[24] L. I. Rudin, S. J. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[25] J. F. Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.

[26] A. N. Tikhonov and V. Y. Arsenin. *Solution of Ill-Posed Problems*. Washington, DC: V.H. Winston, 1977.

[27] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal of Scientific Computing*, 17:227–238, 1996.