A Parallel Proximal Algorithm for Anisotropic Total Variation Minimization

Kamilov, U.

TR2017-002  February 2017

Abstract
Total variation (TV) is one of the most popular regularizers for stabilizing the solution of ill-posed inverse problems. This paper proposes a novel proximal-gradient algorithm for minimizing TV regularized least-squares cost functionals. Unlike traditional methods that require nested iterations for computing the proximal step of TV, our algorithm approximates the latter with several simple proximals that have closed form solutions. We theoretically prove that the proposed parallel proximal method achieves the TV solution with arbitrarily high precision at a global rate of converge that is equivalent to the fast proximal gradient methods. The results in this paper have the potential to enhance the applicability of TV for solving very large scale imaging inverse problems.

IEEE Transactions on Image Processing

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.

Copyright © Mitsubishi Electric Research Laboratories, Inc., 2017
201 Broadway, Cambridge, Massachusetts 02139
A Parallel Proximal Algorithm for Anisotropic Total Variation Minimization

Ulugbek S. Kamilov, Member, IEEE

Abstract—Total variation (TV) is a one of the most popular regularizers for stabilizing the solution of ill-posed inverse problems. This paper proposes a novel proximal-gradient algorithm for minimizing TV regularized least-squares cost functionals. Unlike traditional methods that require nested iterations for computing the proximal step of TV, our algorithm approximates the latter with several simple proximals that have closed form solutions. We theoretically prove that the proposed parallel proximal method achieves the TV solution with arbitrarily high precision at a global rate of converge that is equivalent to the fast proximal-gradient methods. The results in this paper have the potential to enhance the applicability of TV for solving very large scale imaging inverse problems.

Index Terms—Proximal gradient method, total variation regularization, inverse problems, convex optimization

I. INTRODUCTION

The problem of estimating an unknown signal from noisy linear observations is fundamental in signal processing. The estimation task is often formulated as the linear inverse problem that consists in minimizing a cost functional. The latter typically includes a quadratic data-fidelity term, as well as a regularizer that mitigates the ill-posedness of the problem by promoting solutions with desirable properties such as transform-domain sparsity or positivity.

One of the most widely used regularizers in the context of image reconstruction is total variation (TV). TV was originally introduced by Rudin et al. [1] as a regularization approach capable of reducing noise, while preserving image edges. It is often interpreted as a sparsity-promoting $\ell_1$-penalty on the image gradient and has proven to be successful in a wide range of applications in the context of sparse recovery of images from incomplete or corrupted measurements [2]–[8].

The minimization of TV regularized cost functionals is a nontrivial optimization task. The challenging aspects are the non-smooth nature of the regularization term and the large amount of data that needs to be processed in a typical application. Proximal gradient methods [9] such as iterative shrinkage/thresholding algorithm (ISTA) [10]–[12] and its accelerated variants [13], [14] are standard approaches to circumvent the non-smoothness of the TV regularizer and are among the methods of choice for solving practical linear inverse problems.

Nonetheless, ISTA–based optimization of TV is complicated by the fact that the corresponding proximal operator does not have a closed form solution. Practical implementations rely on computational solutions that require an additional nested optimization algorithm for evaluating the TV proximal [15], [16]. This typically leads to a prohibitively slow reconstruction when dealing with very large scale imaging problems such as the ones in 3D computational imaging.

In this paper, we propose a novel approach for solving TV–based imaging problems that requires no nested iterations. We consider anisotropic variant of TV and eliminate sub-iterations by approximating the exact proximal with an alternative that evaluates several simpler proximal operators that have closed form solutions. Conceptually, our method builds upon two distinct lines of prior research on inexact proximal-gradient algorithms [17]–[20] and cycle spinning [10], [21]–[23]. We believe that the results presented in this paper are useful to practitioners working with very large scale problems where the bottleneck is in the evaluation of the TV proximal.

Two key contributions of this paper are summarized as follows

- New parallel proximal-gradient method for solving anisotropic TV regularized reconstruction problems. The algorithm builds upon fast iterative shrinkage/thresholding algorithm (FISTA) [15], but avoids sub-iterations by exploiting a specific decomposition of TV as an average of several simple regularizers.
- Theoretical analysis of the method proving that it achieves the TV solution with arbitrarily high precision at a global convergence rate of $O(1/t^2)$, where $t$ denotes the iteration number. This makes the proposed algorithm ideal for solving very large-scale image reconstruction problems, where nested optimization is undesirable. In addition, we experimentally illustrate possible computational gains due to our approach on the problems of image deconvolution and super-resolution.

A. Related Work

The results in this paper are most closely related to the work on TV–based image reconstruction by Beck and Teboulle [15]. Their approach for solving TV requires an additional nested FISTA, implemented in the dual form, for evaluating the proximal. Our aim is to avoid sub-iterations by replacing the exact TV proximal with a specific approximation that can still guarantee convergence to the TV solution. While Beck and Teboulle’s approach considers both isotropic and anisotropic variants of the TV regularization, we restrict our attention to anisotropic TV.

In another related work, Condat [24] proposed a direct algorithm for 1D TV proximal, which can be used to accelerate the
resolution of 1D TV-regularized inverse problems. Approach taken in this paper is fundamentally different, where instead of finding an exact computational solution for evaluating the proximal, we find a suitable approximation. This, however, allows our method to generalize to inverse problems of arbitrary number of dimensions.

From the convex optimization perspective, our work is related to inexact proximal-gradient algorithms that were extensively studied for various applications. For example, in the context of online learning, Zinkevich [25] has proposed an incremental projected-gradient algorithm that minimizes a smooth cost functional by evaluating its partial gradients. He has proved that, with a proper adaptation of the step size, the algorithm reaches the minimizer at a global convergence rate of $O(1/\sqrt{T})$. The algorithm and its analysis were extended by Duchi and Singer [26] for optimizing cost functionals containing non-smooth regularizers. Bertsekas [18] further generalized those results to include algorithms that combine partial gradient, subgradient, and proximal iterations. D’Aspremont [17] showed that optimal $O(1/i^2)$ complexity of Nesterov’s algorithm [27] is preserved, when the gradient is computed only up to a small, uniformly bounded error. More recently, Schmidt [19] and Devolder et al. [20] have investigated the convergence rates of proximal-gradient algorithms where the goal is to compute the unknown signal $x \in \mathbb{R}^N$ from the noisy measurements $y \in \mathbb{R}^M$. Here, the matrix $H \in \mathbb{R}^{M \times N}$ models the response of the acquisition device and the vector $e \in \mathbb{R}^M$ represents the measurement noise, which is often assumed to be i.i.d. Gaussian. When the problem (1) is ill-posed, the standard approach is to formulate the estimation as the following minimization problem
\begin{equation}
\hat{x} = \arg \min_{x \in \mathbb{R}^N} \{C(x)\} \quad (2)
\end{equation}
\begin{equation}
= \arg \min_{x \in \mathbb{R}^N} \{D(x) + R(x)\} \quad (3)
\end{equation}
where
\begin{equation}
D(x) \triangleq \frac{1}{2} \|y - Hx\|_{l_2}^2 \quad (4)
\end{equation}
is the quadratic data fidelity term. Two common variants of TV are the anisotropic TV regularizer
\begin{equation}
R(x) \triangleq \lambda \sum_{n=1}^{N} \sum_{d=1}^{D} |D_d x_n|^2 \quad (5)
\end{equation}
and isotropic TV regularizer
\begin{equation}
R(x) \triangleq \lambda \sum_{n=1}^{N} \|D_x n\|_{l_2}^2 \triangleq \lambda \sum_{n=1}^{N} \sqrt{\sum_{d=1}^{D} (|D_d x_n|)^2}. \quad (6)
\end{equation}
Here, $D : \mathbb{R}^N \rightarrow \mathbb{R}^{N \times D}$ is the discrete gradient operator, $\lambda > 0$ is a parameter controlling amount of regularization, and $D$ is the number of dimensions in the signal. The matrix $D_d$ denotes the finite difference operation along the dimension $d$ with appropriate boundary conditions (periodization, Neumann boundary conditions, etc.).

### II. Background

#### A. Problem Formulation

We consider a linear inverse problem
\begin{equation}
y = Hx + e, \quad (1)
\end{equation}
where the goal is to compute the unknown signal $x \in \mathbb{R}^N$ from the noisy measurements $y \in \mathbb{R}^M$. Here, the matrix $H \in \mathbb{R}^{M \times N}$ models the response of the acquisition device and the vector $e \in \mathbb{R}^M$ represents the measurement noise, which is often assumed to be i.i.d. Gaussian. When the problem (1) is ill-posed, the standard approach is to formulate the estimation method for solving (2) is ISTA
\begin{equation}
x^t \leftarrow \text{prox}_{\gamma R}(x^{t-1} - \gamma t D(x^{t-1})), \quad (7)
\end{equation}
where the gradient of the quadratic term is given by
\begin{equation}
\nabla D(x) = H^T(Hx - y) \quad (8)
\end{equation}
and $\gamma_t > 0$ is a step-size that can be set to $\gamma_t = 1/L$ with $L \triangleq \lambda_{\text{min}}(H^T H)$ to ensure convergence [14]. Iteration (7) combines the gradient-descent step with a proximal operation defined as
\begin{equation}
\text{prox}_{\gamma R}(z) \triangleq \arg \min_{x \in \mathbb{R}^N} \left\{ \frac{1}{2} \|x - z\|^2 + \gamma R(x) \right\}. \quad (9)
\end{equation}
The proximal operator (9) corresponds to the regularized solution of the denoising problem where $H$ is an identity. Although elegant and simple, it is well known that ISTA only achieves a suboptimal convergence rate of $O(1/t)$. Its accelerated version FISTA can be described as follows
\begin{equation}
x^t \leftarrow \text{prox}_{\gamma R}(u^{t-1} - \gamma t \nabla D(u^{t-1})), \quad (10a)
\end{equation}
\begin{equation}
q_t \leftarrow \left(1 + \sqrt{1 + 4q_{t-1}^2} \right)/2 \quad (10b)
\end{equation}
\begin{equation}
u^t \leftarrow x^t + (q_t - 1)/q_t (x^t - x^{t-1}) \quad (10c)
\end{equation}
with $u^0 = x^0$ and $q_0 = 1$. Method (10) preserves the simplicity of ISTA (7), but provides a significantly better rate of convergence as summarized in the following theorem from [14].
Theorem 1. Assume a fixed step size \( \gamma_t = \gamma \in (0, 1/L] \) and let \( \{x^t\}_{t=1,2,...} \) be the sequence of estimates generated by FISTA. Then for any \( t \geq 1 \), we have that
\[
C(x^t) - C(x^*) \leq \frac{2}{\gamma(t+1)^2} \|x^0 - x^*\|_2^2,
\]
where \( x^* \) is a minimizer of \( C \).

The change in convergence rate from \( O(1/t) \) to \( O(1/t^2) \) becomes crucial when solving very large scale inverse problems, where one tries to reduce the amount of matrix-vector products with \( H \) and \( H^T \).

Application of ISTA and FISTA is straightforward for regularizers such as \( \ell_1 \)-penalty that admit closed form proximal operators \( 9 \). However, many other popular regularizers including TV do not have closed form proximals and require an additional iterative algorithm for solving \( 9 \). This adds significant computational overhead to the estimation process, which we shall eliminate for the anisotropic TV in the next section.

III. PROPOSED APPROACH

In this section, we present our main results. We start by introducing the proposed approach and then follow up by analyzing its convergence.

A. General formulation

We turn our attention to a more general optimization problem
\[
\hat{x} = \arg\min_{x \in \mathbb{R}^N} \{C(x)\},
\]
where the cost functional is of the following form
\[
C(x) = D(x) + R(x) = D(x) + \frac{1}{K} \sum_{k=1}^{K} R_k(x).
\]

The precise connection between \( 13 \) and TV-regularized cost functional will be discussed shortly. We assume that the data-fidelity term \( D \) is convex and differentiable with a Lipschitz continuous gradient. This means that there exists a constant \( L > 0 \) such that, for all \( x, z \in \mathbb{R}^N \),
\[
\|\nabla D(x) - \nabla D(z)\|_{\ell_2} \leq L\|x - z\|_{\ell_2}.
\]
We also assume that each \( R_k \) is a continuous, convex function that is possibly nondifferentiable and that the optimal value \( C^* \) is finite and attained at \( x^* \) (which is not necessarily unique).

We consider fast parallel proximal algorithms that have the following form
\[
x_t^{(1)} = \frac{1}{K} \sum_{k=1}^{K} \text{prox}_{\gamma_k R_k}(u^{(t-1)} - \gamma_t \nabla D(u^{(t-1)}))
\]
\[
q_t = \left(1 + \sqrt{1 + 4q_{t-1}^2}\right)/2
\]
\[
u_t = x^t + (q_t - 1)/q_t (x_t^{(1)} - x_t^{(0)}),
\]
with \( u^0 = x^0 \) and \( q_0 = 1 \). Here, \( \text{prox}_{\gamma_k R_k} \) is the proximal operator associated with \( \gamma_k R_k \). We are specifically interested in the case where the proximals \( \text{prox}_{\gamma_k R_k} \) have closed forms, in which case they are preferable to the computation of the full proximal \( \text{prox}_{\gamma R} \).

We now establish a connection between \( 13 \) and TV-regularized cost. Define a linear transform \( W : \mathbb{R}^N \rightarrow \mathbb{R}^{N \times D} \) that consists of two sub-operators: the averaging operator \( A : \mathbb{R}^N \rightarrow \mathbb{R}^{N \times D} \) and the discrete gradient \( D \) as in \( 5 \), both normalized by \( 1/\sqrt{2} \). The averaging operator consists of \( D \) matrices \( A_d \) that denote the pairwise averaging along the dimension \( d \). Accordingly, the operator \( W \) is a union of scaled and shifted discrete Haar wavelet and scaling functions along each dimension \( 28 \). Since we consider all possible shifts along each dimension the transform is redundant and can be interpreted as the union of \( K = 2D \), scaled, orthogonal transforms
\[
W = \begin{bmatrix} W_1 \end{bmatrix}, \quad \vdots \quad \begin{bmatrix} W_K \end{bmatrix}.
\]

Figure 1 illustrates the grouping of differences and averages into 4 wavelets for a \( 2D \) image. The transform \( W \) and its pseudo-inverse
\[
W^\dagger = \frac{1}{K}[W^\top_1 \ldots W^\top_K]
\]
satisfy the following two properties of Parseval frames \( 35 \)
\[
\text{arg min}_{x \in \mathbb{R}^N} \left\{ \frac{1}{2} \|z - Wx\|_2^2 \right\} = W^\top z \quad \text{(for all } z \in \mathbb{R}^{KN})
\]
and
\[
W^\top W = I.
\]

One can thus express the anisotropic TV regularizer as the following sum
\[
R(x) = \lambda \sqrt{2} \sum_{k=1}^{K} \sum_{n \in H_k} ||[W_k x]_n||,
\]
where \( H_k \subseteq [1 \ldots N] \) is the set of all detail coefficients of the transform \( W_k \). Then, the proposed parallel proximal algorithm for TV can be expressed as follows
\[
\begin{align*}
& z^t = x^{t-1} - \gamma_t H^\top(Hx^{t-1} - y) \\
& x^t = \frac{1}{K} \sum_{k=1}^{K} W^\top_k T(W_k z^t; \sqrt{2}K \gamma_t \lambda),
\end{align*}
\]
and fast parallel proximal algorithm can be expressed as
\[
\begin{align*}
z^t &\leftarrow u^{t-1} - \gamma_t H^T (H u^{t-1} - y) \quad (20a) \\
x^t &\leftarrow \frac{1}{K} \sum_{k=1}^{K} W_k^T T(W_k z^t; \sqrt{2} K \gamma_t \lambda) \quad (20b) \\
q_t &\leftarrow (1 + \sqrt{1 + 4q^2_{t-1}})/2 \quad (20c) \\
u^t &\leftarrow x^t + (q_{t-1} - 1)/q_t (x^t - x^{t-1}), \quad (20d)
\end{align*}
\]
with \(u^0 = x^0\) and \(q_0 = 1\). Here, \(T\) is the component-wise shrinkage function
\[
T(y; \tau) \triangleq \max(|y| - \tau, 0) \frac{y}{|y|}, \quad (21)
\]
which is applied only on scaled differences \(Dz^t\).

The algorithm in (19) is closely related to a technique called cycle spinning [21] that is commonly used for improving the performance of wavelet-domain denoising. In particular, when \(H = I\) and \(\gamma_t = 1\), for all \(t = 1, 2, \ldots\), the algorithm yields the solution
\[
\hat{x} \leftarrow W^T T(W y; \sqrt{2} K \lambda), \quad (22)
\]
which can be interpreted as the traditional cycle spinning algorithm restricted to the the Haar wavelet-transform.

B. Theoretical convergence

The convergence results in this section assume that the gradient of \(D\) and subgradients of \(R_k\) are bounded, i.e., there exists \(G > 0\) such that for all \(k\) and \(t\), \(\|\nabla D(x^t)\|_{\ell_2} \leq G\) and \(\|\nabla R_k(x^t)\|_{\ell_2} \leq G\). The following proposition that we prove in the appendix establishes the convergence of the fast parallel proximal algorithm.

Proposition 1. Assume a fixed step size \(\gamma_t = \gamma \in (0, 1/L]\). Then, we have that
\[
C(x^t) - C(x^*) \leq \frac{2}{\gamma (t + 1)^2} \|x^0 - x^*\|_{\ell_2}^2 + 4\gamma G^2. \quad (23)
\]

Proof: See Appendix.

Proposition 1 states that for a constant step-size, convergence can be established to the neighborhood of the optimum, which can be made arbitrarily close to 0 by letting \(\gamma \rightarrow 0\). Additionally, the global convergence rate of fast parallel proximal algorithm matches that of FISTA. Note that the result here extends the preliminary work [34] that established the convergence of the standard parallel proximal algorithm (19).

IV. NUMERICAL EXAMPLES

The main purpose of this section is to empirically demonstrate the convergence of our fast parallel proximal algorithm (FPPA) and validate our theoretical contribution in Proposition 1. We additionally present some comparisons of TV against some other state-of-the-art methods on the problem of single image super-resolution. All the simulations were performed with MATLAB on an Apple iMac with a 4 GHz Intel Core i7 processor and 32 GBs of memory.

A. Empirical Validation of Proposition 1

To empirically validate the convergence of the algorithm, we consider an image deblurring problem where the blur is a 5 \times 5 Gaussian of variance 2 and where the blurry image is contaminated with an additive white Gaussian noise (AWGN) of 30 dB SNR. We evaluate the performance on a standard image dataset Set14, used in the previous works [36]–[38]. Following these works, only the luminance component of color images was considered. Some examples from the dataset are illustrated in Figure 2.

The simulation results on all 14 images are summarized in Table I. There we compare FPPA against the exact TV-FISTA [15], which computes the TV proximal in the dual domain. For FPPA, we consider 3 different step-sizes \(\gamma = 1/L, \quad \gamma = 1/(4L),\) and \(\gamma = 1/(16L)\), where \(L = \lambda_{\text{max}}(H^T H)\) is the Lipschitz constant, and report three quantities: (a) the relative cost accuracy \(\|C(x^t) - C(x^*)\|/C(x^*)\), (b) the relative peak signal-to-noise ratio (PSNR) in dB with respect to the TV solution \(x^*\), and (c) the speedup factor. The images \(x^t\) and \(x^*\) are computed with FPPA and the exact TV-FISTA, respectively, and \(C\) is the TV-regularized least-squares cost. The regularization parameter \(\lambda\) was manually selected for the optimal PSNR performance of TV. To ensure the convergence, we deliberately select the same strict stopping rules for all the algorithms; they are run for a maximum of \(t_{\text{max}} = 10^4\) iterations with an additional stopping criterion based on measuring the relative change of the solution in two successive iterations
\[
\frac{\|x^t - x^{t-1}\|_{\ell_2}}{\|x^{t-1}\|_{\ell_2}} \leq 10^{-5}. \quad (24)
\]

The maximal number of inner iterations for the proximal of
Table 1: The relative cost, PSNR with respect to the TV solution, and the speedup factor for the images from Set14.

<table>
<thead>
<tr>
<th>Images</th>
<th>Set14</th>
<th>γ = 1/L</th>
<th>Cost Accuracy</th>
<th>PSNR rel. TV</th>
<th>Time</th>
<th>γ = 1/(4L)</th>
<th>Cost Accuracy</th>
<th>PSNR rel. TV</th>
<th>Time</th>
<th>γ = 1/(16L)</th>
<th>Cost Accuracy</th>
<th>PSNR rel. TV</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baboon</td>
<td>0.0035</td>
<td>48.40</td>
<td>x16</td>
<td>0.0009</td>
<td>58.45</td>
<td>x11</td>
<td>0.0002</td>
<td>63.52</td>
<td>x7</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Barbara</td>
<td>0.0062</td>
<td>48.01</td>
<td>x19</td>
<td>0.0016</td>
<td>57.24</td>
<td>x11</td>
<td>0.0005</td>
<td>65.23</td>
<td>x5</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Bridge</td>
<td>0.0089</td>
<td>45.66</td>
<td>x20</td>
<td>0.0023</td>
<td>55.14</td>
<td>x12</td>
<td>0.0006</td>
<td>64.89</td>
<td>x8</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Coastguard</td>
<td>0.0081</td>
<td>46.81</td>
<td>x14</td>
<td>0.0021</td>
<td>55.95</td>
<td>x8</td>
<td>0.0005</td>
<td>65.23</td>
<td>x5</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Comic</td>
<td>0.0087</td>
<td>43.32</td>
<td>x13</td>
<td>0.0023</td>
<td>52.65</td>
<td>x8</td>
<td>0.0006</td>
<td>62.13</td>
<td>x5</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Face</td>
<td>0.0290</td>
<td>43.84</td>
<td>x13</td>
<td>0.0083</td>
<td>51.70</td>
<td>x7</td>
<td>0.0022</td>
<td>61.21</td>
<td>x5</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Flowers</td>
<td>0.0097</td>
<td>47.13</td>
<td>x12</td>
<td>0.0026</td>
<td>56.32</td>
<td>x8</td>
<td>0.0007</td>
<td>65.42</td>
<td>x5</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Foreman</td>
<td>0.0130</td>
<td>43.03</td>
<td>x15</td>
<td>0.0037</td>
<td>51.23</td>
<td>x9</td>
<td>0.0010</td>
<td>60.38</td>
<td>x5</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Lenna</td>
<td>0.0260</td>
<td>41.06</td>
<td>x22</td>
<td>0.0080</td>
<td>48.67</td>
<td>x12</td>
<td>0.0021</td>
<td>57.70</td>
<td>x8</td>
<td>0.0006</td>
<td>66.22</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Man</td>
<td>0.0084</td>
<td>47.61</td>
<td>x17</td>
<td>0.0022</td>
<td>56.88</td>
<td>x10</td>
<td>0.0006</td>
<td>60.96</td>
<td>x8</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Monarch</td>
<td>0.0200</td>
<td>43.73</td>
<td>x19</td>
<td>0.0058</td>
<td>51.85</td>
<td>x12</td>
<td>0.0015</td>
<td>60.96</td>
<td>x8</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Pepper</td>
<td>0.0130</td>
<td>44.31</td>
<td>x19</td>
<td>0.0038</td>
<td>52.66</td>
<td>x12</td>
<td>0.0010</td>
<td>61.84</td>
<td>x8</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Ppt3</td>
<td>0.0160</td>
<td>41.69</td>
<td>x16</td>
<td>0.0047</td>
<td>49.50</td>
<td>x9</td>
<td>0.0012</td>
<td>58.03</td>
<td>x6</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Zebra</td>
<td>0.0160</td>
<td>41.49</td>
<td>x13</td>
<td>0.0044</td>
<td>50.29</td>
<td>x8</td>
<td>0.0011</td>
<td>59.55</td>
<td>x5</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>0.0133</td>
<td>44.72</td>
<td>x16</td>
<td>0.0038</td>
<td>53.47</td>
<td>x10</td>
<td>0.0009</td>
<td>62.53</td>
<td>x6</td>
<td>0.0004</td>
<td>65.89</td>
<td>x7</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. Recovery of test images from blurry and noisy measurements. The TV cost functional $C(x')$ is plotted over 100 iterations for 3 distinct step-sizes $\gamma$. The dotted line plots the cost of the exact TV-FISTA. Note the nearly perfect match between FPPA at $\gamma = 1/L$ and TV-FISTA, which is consistent with their identical global convergence rates.

Figure 5. Recovery of test images from blurry and noisy measurements. PSNR (dB) is plotted over 100 iterations for 3 distinct step-sizes $\gamma$. The dotted line plots the PSNR obtained by the exact TV-FISTA. The plot illustrates that the quality of the reconstructed images is approximately the same for both FPPA and TV-FISTA.

TV-FISTA was set to 100, also with a stopping criterion (24).

Figure 3 illustrates the evolution of the relative cost accuracy at every iteration of FPPA. It also reports the theoretical upper bound on the performance of FISTA in (11), as well as the actual evolution of the relative cost accuracy at every iteration of TV-FISTA. Figures 4 and 5 show the evolution of the cost $C$ and PSNR, respectively, in the first 100 iterations of the algorithms. These figures highlight the convergence of FFPA and TV-FISTA within those 100 iterations, which indicates that the stopping criterion selected above was sufficiently strict. Finally, Figures 6 and 7 offer visual and quantitative evaluation of the final estimated images for Man and Baboon.

Proposition 1 suggests that the gap $(C(x') - C(x^*))$ is proportional to the step-size and shrinks to 0 as the step-size is reduced. Such behavior is clearly observed both in Table I and Figure 3. This suggests that our theoretical result is also valid in practice. On average, the relative cost accuracy for Set14 is about 1.33% at $\gamma = 1/L$, and decreases further for $\gamma = 1/(4L)$ and $\gamma = 1/(16L)$. Additionally, we note that the solution of our algorithm is very close to that of TV-FISTA visually and quantitatively. This implies that, while requiring no nested iterations, FPPA can potentially approximate the solution of TV with arbitrarily accurate precision at $O(1/t^2)$ convergence rate of FISTA. Note also that FPPA is substantially faster than the standard approach that requires sub-iterations. For example in our simulations, FPPA achieved an average speed-up of 16 for $\gamma = 1/L$ on the Set14 images.

B. Discussion on TV-based Imaging

Minimization of TV regularized cost functionals is one of many approaches for reconstructing images from their linear measurements. A vast majority of these approaches rely on some form of prior information or constraints for regularizing the image formation process [39]. Depending on the type of prior information, algorithms can be loosely classified into several categories including traditional linear methods [40], Bayesian and statistical methods [35], [41]–[44], optimization based methods with pre-specified regularizers [45]–[48], patch based methods exploiting similarities in a given image [49]–[52], methods based on dictionary learning [53]–[55], supervised learning approaches based on deep convolutional networks (CNNs) [37], [38], [56]–[61].
It has been widely reported that powerful patch-based methods based on the BM3D algorithm outperform TV on certain image restoration problems such as deblurring and denoising [47], [52]. Similar improvements were observed by another class of powerful methods based on deep convolutional networks [37], [57]. Nonetheless, each reconstruction approach has a distinct set of advantages and drawbacks that influences its applicability to various imaging problems. For example, patch-based methods rely on the block-matching procedure for grouping similar image patches. This implies that these methods require a suitable initial estimate of the image for a reliable block-matching, which makes them ideal for image denoising or deblurring [52], but makes their generalization to arbitrary imaging problems difficult. On the other hand, CNN based methods have a simple structure as a succession of convolutions. These methods, therefore, enjoy lower computational complexity for reconstruction compared to the patch-based methods. However, they typically require a separate training procedure over a sufficiently large image dataset. For example, Dong et al. [56] report that it took about three days to train their SR-CNN model on 24800 sub-images of size $32 \times 32$, extracted from 91 training images. Similarly, Chen and Pock [38] report 20.8 hours of training on 400 images of size $180 \times 180$. Additionally, model parameters in such imaging methods are highly optimized for a given problem, which implies that a slight modification in the acquisition system requires complete retraining of the network.

Finally, while large training datasets are easy to generate for certain class of problems such as, for example, image super-resolution, they are harder to obtain for other applications such as bio-microscopy or medical imaging.

Compared to more advanced methods such as BM3D [52] or SR-CNN [37], TV based imaging does not rely on a suitable initialization for block-matching or require an additional training procedure. This makes it straightforward to apply to a larger set of imaging problems including image restoration [3], [15], depth imaging [62], [63], magnetic resonance imaging (MRI) [5], [64], computer tomography (CT) [33], phase-contrast tomography [65], optical microscopy [66]–[68], and inverse wave scattering [69]. In particular, TV imaging algorithms are particularly well-suited for very large-scale 3D imaging problems, where training and block-matching become prohibitively expensive. In such applications, it becomes crucial to have access to fast optimization algorithms for TV such as the one proposed here.

TV has been extensively compared in several prior works and a comprehensive comparison falls beyond the scope of this paper. Nonetheless, Table II summarizes its performance, in terms of PSNR (dB) and running time (sec), on image super-resolution over a dataset Set14. Specifically, we exactly reproduce the image upscaling problem that was also considered in previous works [37], [38], [54]. We report the results of both FPPA and TV-FISTA, as well as the results of a simple bicubic interpolation, Hessian Schatten-Norm Regularization ($HS_2$) [47], and SR-CNN [37]. FPPA, TV-FISTA, and $HS_2$
were run for 20 iterations with PSNR optimal regularization parameters. The computation of the proximals of TV-FISTA and $H_S^2$ was limited to 5 sub-iterations. We relied on the MATLAB implementations of $H_S^2$ and SR-CNN that was provided by the authors. Since none of the methods were optimized for speed, the running times are expected to further improve after a careful code optimization.

The very first observation is that FPPA closely approximates the TV-FISTA solution at the fraction of the running time (0.04 dB difference for about $\times 3$ reduction in time). Additionally, both TV methods yield images that are within 0.4 dB compared to the powerful SR-CNN. As TV does not require an extensive model training procedure, this indicates that it can be a simpler, but an effective, alternative to SR-CNN when training is not practical or possible.

V. CONCLUSION

The fast parallel proximal method, which was presented in this paper, is beneficial in the context of anisotropic TV regularized image reconstruction, especially when the computation of the TV proximal is costly. We presented a mixture of theoretical and empirical evidence demonstrating that the method can accurately approximate the TV solution at the competitive global convergence rates without resorting to expensive sub-iterations. Future work will aim at extending the theoretical analysis presented here to isotropic variant of TV and by applying the methods to practical large scale imaging problems. Additionally, it would be beneficial to study the method when $\gamma_t$ is decreased progressively, which could mitigate the stalling effect when $\gamma$ is fixed to a small value.

VI. APPENDIX

A. Review of Convex Analysis

Before embarking on the actual proof of Proposition 1, it is convenient to summarize a few facts that will be used next.

A subgradient of a convex function $C$ at $x$ is any vector $\tilde{\nabla}C(x)$ that satisfies the inequality

$$C(y) \geq C(x) + \langle \tilde{\nabla}C(x), y - x \rangle,$$  \hspace{1cm} (25)

for all $y$. When $C$ is differentiable, the only possible choice for $\tilde{\nabla}C$ is the gradient $\nabla C$. The set of subgradients of $C$ at $x$ is the subdifferential of $C$ at $x$, denoted $\partial C(x)$. The condition that $\tilde{\nabla}C$ be a subgradient of $C$ at $x$ can then be written $\tilde{\nabla}C(x) \in \partial C(x)$.

We also remind another fundamental property of a smooth and continuously differentiable function $D$ with a Lipschitz continuous gradient and Lipschitz constant $L$. For any $\gamma \in (0, 1/L]$, such functions satisfy

$$D(x) \leq D(y) + \langle \nabla D(y), x - y \rangle + \frac{1}{2\gamma} \|x - y\|_2^2,$$  \hspace{1cm} (26)

and all $x, y$. 

The proximal operator is defined as

$$x = \text{prox}_{\gamma R}(z)$$

arg\min_{x \in \mathbb{R}^N} \left\{ \frac{1}{2} \|x - z\|_2^2 + \gamma R(x) \right\} (27b)

where $\gamma > 0$ and $R$ is a convex continuous function. The proximal operator is characterized by the following inclusion, for all $x, z \in \mathbb{R}^N$

$$x = \text{prox}_{\gamma R}(z) \iff \frac{z - x}{\gamma} \in \partial R(x). \quad (28)$$

### B. Proof of Proposition 1

We consider the following algorithm, which is perfectly equivalent to the fast parallel proximal algorithm (14)

$$u^t \leftarrow (1 - 1/q_t)x^{t-1} + (1/q_t)v^{t-1} \quad (29a)$$

$$x^t \leftarrow \frac{1}{K} \sum_{k=1}^K \text{prox}_{\gamma R}(u^t - \gamma \nabla D(u^t)) \quad (29b)$$

$$v^t \leftarrow x^{t-1} + q_t(x^t - x^{t-1}), \quad (29c)$$

where $x^0 = v^0, q_0 = 1,$ and $q_t$ satisfies

$$q_t^2 - q_t q_{t-1} \leq 0. \quad (30)$$

for all $t = 1, 2, \ldots$ To see the equivalence of (29) to the fast parallel proximal algorithm (14), first set $q_t$ as in (14b) and then eliminate the auxiliary variables $\{v^t\}$ by plugging (29c) into (29a).

We start by using (26) to find an upper bound for $D$ at $x^t$

$$D(x^t) \leq D(u^t) + \langle \nabla D(u^t), x^t - u^t \rangle + \frac{1}{2\gamma} \|x^t - u^t\|_2^2. \quad (31)$$

We then define an intermediate quantity

$$x_k^t \triangleq \text{prox}_{\gamma R}(u^{t-1} - \gamma \nabla D(u^{t-1})).$$

The optimality conditions for (29b) imply that there must equate $K$ subgradient vectors $\tilde{\nabla} R_k(x_k^t) \in \partial R_k(x_k^t)$ such that

$$x_k^t = u^t - \gamma(\nabla D(u^t) + \tilde{\nabla} R_k(x_k^t)). \quad (32)$$

This implies that

$$x^t = u^t - \gamma(\nabla D(u^t) + g^t), \quad (33)$$

where

$$g^t \triangleq \frac{1}{K} \sum_{k=1}^K \tilde{\nabla} R_k(x_k^t). \quad (34)$$

The relationships (32) and (33) together with bounds on the subgradients implies that

$$\|x^t - x_k^t\|_2 \leq \|g^t - \tilde{\nabla} R_k(x_k^t)\|_2 \leq 2\gamma G. \quad (35)$$

We then bound $R_k$ at any $z \in \mathbb{R}^N$ as follows

$$R_k(z) \geq R_k(x_k^t) + (\tilde{\nabla} R_k(x_k^t), z - x_k^t) \quad (36a)$$

$$= R_k(x_k^t) + (\tilde{\nabla} R_k(x_k^t), z - x_k^t) \quad (36b)$$

$$\geq \langle \nabla R_k(x_k^t), x_k^t - x_k^t \rangle \quad (36c)$$

$$= \langle \nabla R_k(x_k^t), z - x_k^t \rangle + (\tilde{\nabla} R_k(x_k^t), x_k^t - x_k^t \rangle \quad (36d)$$

$$\leq \langle \nabla R_k(x_k^t), z - x_k^t \rangle + 4\gamma G^2, \quad (37b)$$

where in (a) and (b) we used the convexity of $R_k$. By rearranging (36), we obtain for any $z \in \mathbb{R}^N$

$$R_k(x^t) \leq R_k(z) - (\tilde{\nabla} R_k(x_k^t), z - x_k^t) \quad (37a)$$

$$= (\tilde{\nabla} R_k(x_k^t), x_k^t - x_k^t \rangle \quad (36b)$$

$$\geq \langle \nabla R_k(z), z - x_k \rangle + 4\gamma G^2, \quad (37b)$$

where in (a) we used Cauchy-Schwarz inequality and in (b) we used Cauchy-Schwarz inequality.

### Table II: UpScaling by factor x3 Performance in Terms of PSNR and RunTime for the Images from Set14.

<table>
<thead>
<tr>
<th>Images</th>
<th>Bicubic</th>
<th>FPPA</th>
<th>TV</th>
<th>H-S</th>
<th>SR-CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baboon</td>
<td>23.21</td>
<td>23.53</td>
<td>23.53</td>
<td>23.54</td>
<td>23.60</td>
</tr>
<tr>
<td>Bridge</td>
<td>24.40</td>
<td>24.94</td>
<td>24.94</td>
<td>24.96</td>
<td>25.07</td>
</tr>
<tr>
<td>Coastguard</td>
<td>26.55</td>
<td>27.03</td>
<td>27.03</td>
<td>27.04</td>
<td>27.20</td>
</tr>
<tr>
<td>Comic</td>
<td>23.12</td>
<td>23.92</td>
<td>23.91</td>
<td>23.90</td>
<td>24.39</td>
</tr>
<tr>
<td>Face</td>
<td>32.82</td>
<td>33.35</td>
<td>33.38</td>
<td>33.59</td>
<td>33.58</td>
</tr>
<tr>
<td>Flowers</td>
<td>27.23</td>
<td>28.53</td>
<td>28.37</td>
<td>28.36</td>
<td>28.97</td>
</tr>
<tr>
<td>Foreman</td>
<td>31.18</td>
<td>33.17</td>
<td>33.05</td>
<td>32.84</td>
<td>33.15</td>
</tr>
<tr>
<td>Lenna</td>
<td>31.68</td>
<td>32.73</td>
<td>32.69</td>
<td>32.94</td>
<td>33.39</td>
</tr>
<tr>
<td>Man</td>
<td>27.01</td>
<td>27.84</td>
<td>27.76</td>
<td>27.78</td>
<td>28.18</td>
</tr>
<tr>
<td>Monarch</td>
<td>29.43</td>
<td>31.71</td>
<td>31.54</td>
<td>30.75</td>
<td>32.39</td>
</tr>
<tr>
<td>Pepper</td>
<td>32.39</td>
<td>33.91</td>
<td>33.69</td>
<td>33.55</td>
<td>34.35</td>
</tr>
<tr>
<td>Pp</td>
<td>23.71</td>
<td>25.21</td>
<td>25.57</td>
<td>24.86</td>
<td>26.02</td>
</tr>
<tr>
<td>Zebra</td>
<td>26.63</td>
<td>28.36</td>
<td>28.22</td>
<td>27.95</td>
<td>28.87</td>
</tr>
<tr>
<td>Average</td>
<td>27.54</td>
<td>28.64</td>
<td>28.60</td>
<td>28.49</td>
<td>29.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Set14</th>
<th>Bicubic</th>
<th>FPPA</th>
<th>TV</th>
<th>H-S</th>
<th>SR-CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images</td>
<td>PSNR (dB)</td>
<td>Time (s)</td>
<td>PSNR (dB)</td>
<td>Time (s)</td>
<td>PSNR (dB)</td>
</tr>
<tr>
<td>Baboon</td>
<td>23.21</td>
<td>0.004</td>
<td>23.53</td>
<td>0.533</td>
<td>23.53</td>
</tr>
<tr>
<td>Barbara</td>
<td>26.25</td>
<td>0.004</td>
<td>26.77</td>
<td>0.918</td>
<td>26.77</td>
</tr>
<tr>
<td>Bridge</td>
<td>24.40</td>
<td>0.003</td>
<td>24.94</td>
<td>0.556</td>
<td>24.94</td>
</tr>
<tr>
<td>Coastguard</td>
<td>26.55</td>
<td>0.003</td>
<td>27.03</td>
<td>0.233</td>
<td>27.03</td>
</tr>
<tr>
<td>Comic</td>
<td>23.12</td>
<td>0.003</td>
<td>23.92</td>
<td>0.233</td>
<td>23.91</td>
</tr>
<tr>
<td>Face</td>
<td>32.82</td>
<td>0.003</td>
<td>33.35</td>
<td>0.201</td>
<td>33.38</td>
</tr>
<tr>
<td>Flowers</td>
<td>27.23</td>
<td>0.004</td>
<td>28.53</td>
<td>0.370</td>
<td>28.37</td>
</tr>
<tr>
<td>Foreman</td>
<td>31.18</td>
<td>0.003</td>
<td>33.17</td>
<td>0.222</td>
<td>33.05</td>
</tr>
<tr>
<td>Lenna</td>
<td>31.68</td>
<td>0.004</td>
<td>32.73</td>
<td>0.564</td>
<td>32.69</td>
</tr>
<tr>
<td>Man</td>
<td>27.01</td>
<td>0.003</td>
<td>27.84</td>
<td>0.714</td>
<td>27.76</td>
</tr>
<tr>
<td>Monarch</td>
<td>29.43</td>
<td>0.005</td>
<td>31.71</td>
<td>0.881</td>
<td>31.54</td>
</tr>
<tr>
<td>Pepper</td>
<td>32.39</td>
<td>0.005</td>
<td>33.91</td>
<td>0.532</td>
<td>33.69</td>
</tr>
<tr>
<td>Pp3</td>
<td>23.71</td>
<td>0.004</td>
<td>25.21</td>
<td>0.703</td>
<td>25.57</td>
</tr>
<tr>
<td>Zebra</td>
<td>26.63</td>
<td>0.003</td>
<td>28.36</td>
<td>0.485</td>
<td>28.22</td>
</tr>
<tr>
<td>Average</td>
<td>27.54</td>
<td>0.004</td>
<td>28.64</td>
<td>0.510</td>
<td>28.60</td>
</tr>
</tbody>
</table>
for any $z \in \mathbb{R}^N$. We next add bounds (31) and (38) and use the convexity of $D$ to obtain

$$C(x') \leq C(z) + \frac{1}{\gamma} (x' - u^t, z - x')$$

$$+ \frac{1}{2\gamma} \|x' - u^t\|_2^2 + 4\gamma G^2,$$

for all $z \in \mathbb{R}^N$. By evaluating (39) at $z = x^{t-1}$ and $z = x^*$ and taking the convex combination of the bounds, we obtain

$$C(x') - (1 - 1/q_t)C(x^{t-1}) - (1 - 1/q_t)C(x^*)$$

$$\leq \frac{1}{\gamma} \|x' - u^t, x^* - x'\|_2 + \frac{1}{2} \|x' - x^*\|_2^2$$

$$+ \frac{1}{2\gamma q_t}(\|v^{t-1} - x^*\|_2^2 - \|v' - x^*\|_2^2) + 4\gamma G^2,$$

where in the last step we completed the squares and used the definition of the auxiliary variables $\{v^t\}$ in (29a) and (29c). We thus get the following recursive relationship

$$\gamma q_t^2 (C(x') - C(x^*)) + \frac{1}{\gamma} \|v' - x^{*}\|_2^2$$

$$\leq \gamma (q_t^2 - q_t) (C(x^{t-1}) - C(x^*)) + \frac{1}{\gamma} \|v^{t-1} - x^*\|_2^2$$

$$+ 4q_t^2 \gamma G^2.$$

By using the bound (30), using a particular $q_t = (t + 1)/2$, and iterating over $t$, we get

$$C(x') - C(x^*) \leq \frac{2}{\gamma (t + 1)^2} \|x^0 - x^*\|_2^2 + 4\gamma G^2.$$  

This completes the proof.

Acknowledgements

The author would like to thank H. Mansour and A. Vetro for their help in the preparation of this manuscript.

References


