Accelerating Overrelaxed and Monotone Fast Iterative Shrinkage-Thresholding Algorithms With Line Search for Sparse Reconstructions

Marcelo V. W. Zibetti, *Member, IEEE*, Elias S. Helou, and Daniel. R. Pipa, *Member, IEEE*

***Abstract*— Recently, specially crafted unidimensional optimization has been successfully used as line search to accelerate the overrelaxed and monotone fast iterative shrinkage-threshold algorithm (OMFISTA) for computed tomography. In this paper, we extend the use of fast line search to the monotone fast iterative shrinkage-threshold algorithm (MFISTA) and some of its variants. Line search can accelerate the FISTA family considering typical synthesis priors, such as the ℓ_1-norm of wavelet coefficients, as well as analysis priors, such as anisotropic total variation. This paper describes these new MFISTA and OMFISTA with line search, and also shows through numerical results that line search improves their performance for tomographic high-resolution image reconstruction.**

***Index Terms*— Tomographic image reconstruction, iterative shrinkage-thresholding, line search.**

I. Introduction

SPARSE reconstructions with ℓ_1-norm penalty, as used by Compressed Sensing (CS) [1], [2], have a wide range of applications, such as Computed Tomography (CT) [3], [4], Magnetic Resonance Imaging [5], and Reflection Seismology [6]. In all these applications, the reconstruction problem can be posed as an ℓ_1-norm regularized least squares problem of the form:

$$\hat{\mathbf{f}} = \arg\min_{\mathbf{f}} \Psi(\mathbf{f}), \tag{1}$$

with

$$\Psi(\mathbf{f}) = \frac{1}{2}\|\mathbf{g} - \mathbf{H}\mathbf{f}\|_2^2 + \lambda\|\mathbf{L}\mathbf{f}\|_1, \tag{2}$$

also known as LASSO when $\mathbf{L} = \mathbf{I}$ [7], but henceforth referred to as an ℓ_2-ℓ_1 problem. In (2), $\mathbf{f}$ is a $P \times 1$ vector which represents the image to be reconstructed, the $M \times 1$ vector $\mathbf{g}$ represents the captured data and the matrix $\mathbf{H}$ represents the system model. Also, the $N \times P$ matrix $\mathbf{L}$ represents a transformation in the prior model, normally a sparsifying operator, such as wavelet transform or finite difference operator.

There are many ways of solving ℓ_2-ℓ_1 problems, in this paper we are interested in methods based on the proximal operator, such as iterative shrinkage-thresholding algorithm (ISTA) [8] and its fast versions, the fast ISTA (FISTA) [9] and rapidly accelerated proximal gradient (RAPID) [10]. Methods such as FISTA, and its monotone (MFISTA) variant [11], provide accurate numerical approximations to the solution of (2), with low computational cost per iteration and high speed of convergence.

Besides being relatively simple to implement, FISTA and MFISTA have convergence of order $1/k^2$, being k the iteration index. A recent improvement on these methods appeared in [12], where over-relaxation and variable step size were included in order to further improve convergence speed. The variable step size proposed in [12] allows the inclusion of a fast line search in the method. The line search can compute the optimal step size, and if it can be done at very little increase in computational cost, then it can improve the overall convergence speed.

Fast and optimal line search methods for least squares problems mixed with ℓ_1-norm penalty have not been well exploited so far. Recently, [13] studied optimal unidimensional minimization of ℓ_1-penalized denoising problems and presented a fast algorithm that can be adapted to line search in this particular case. Also, [14] introduces a more general optimal line search, however with a high computational cost, so using it in all iterations is not worthwhile. In [15] it was proposed the use of the exact and fast line search from [13] as a method to compute the optimal step size of the OMFISTA. This new method is named OMFISTA with line search (OMFISTA-LS), and it is also revisited in detail in this paper. The optimization from [13] was improved in terms of speed in [16] and utilized as a line search to accelerate ISTA, IRLS, and NLCG with success.

Here, we extend the ideas of line search for OMFISTA [15] and propose its use in MFISTA. The use of line search in OMFISTA is revisited, detailed, and the underlying theory extended accordingly. Also, the line search is applied to both the synthesis prior and the analysis prior versions of MFISTA and OMFISTA. For the analysis prior, the fast gradient projection (FGP) algorithm is utilized as the proximal operator.

Manuscript received July 8, 2016; revised March 31, 2017 and April 25, 2017; accepted April 25, 2017. Date of publication April 28, 2017; date of current version May 26, 2017. The work of M. V. W. Zibetti was supported by CNPq under Grant 475553/2013-6. The work of E. S. Helou was supported in part by CNPq under Grant 311476/2014-7 and in part by FAPESP under Grant 2013/07375-0. The work of D. R. Pipa was supported by CNPq under Grant 312023/2015-4. This paper was presented in part at the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, September 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jan Sijbers. *(Corresponding author: Marcelo V. W. Zibetti.)*

M. V. W. Zibetti and D. R. Pipa are with the Federal University of Technology-Paraná, Curitiba 80230-901, Brazil (e-mail: marcelozibetti@utfpr.edu.br; danielpipa@utfpr.edu.br).

E. S. Helou is with the State University of Sao Paulo in São Carlos, São Carlos 13566-590, Brazil (e-mail: elias@icmc.usp.br).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2017.2699483

1057-7149 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

The main application of these algorithms in this paper is computed X-ray tomography, which is a non-destructive 3D imaging technique used to investigate the internal structure of objects that are transparent to X-ray. There are several applications demanding the identification of structural properties of bones, rocks, seeds, among others. The high-resolution tomographic imaging beamline of the Brazilian Synchrotron Light Source (LNLS) can capture lots of data of scanned objects in each angle, but in order to have an acceptable acquisition time, only few angles are captured. Under this circumstances, fast Radon operators [17], [18], and ℓ_1-norm regularization are required in order to, respectively, enable reasonable computation time for each iteration and provide accurate reconstruction with sparse angular sampling.

The mathematical problem underlying tomography is to recover a function f from its line integrals, i.e., to find f that solves

$$\int_{-\infty}^{\infty} f\left(t\begin{pmatrix}\cos\theta\\ \sin\theta\end{pmatrix} + s\begin{pmatrix}-\sin\theta\\ \cos\theta\end{pmatrix}\right)\mathrm{d}s = g(\theta, t), \tag{3}$$

where g is given. Because data is finite, we try to recover discrete images and, therefore, given the linear nature of the data model, we are actually led to a linear system of equations

$$\mathbf{g} = \mathbf{Hf} + \mathbf{e}, \tag{4}$$

which composes the data-discrepancy term in (2), i.e. $\|\mathbf{g} - \mathbf{Hf}\|_2^2$, and $\mathbf{e}$ represents noise and model errors.

In Section II, we review the FISTA, MFISTA and OMFISTA methods, as well as the FGP method for analysis prior, while in Section III the fast line search for ℓ_2-ℓ_1 problems is discussed. The proposed MFISTA with line search (MFISTA-LS) and the revisited OMFISTA-LS are detailed in section IV, while the experimental comparison and discussion is presented in section V.

II. Fast Iterative Shrinkage-Thresholding Algorithms

FISTA [9], as defined by its acronym, is a fast iterative algorithm, with convergence rate of $O(1/k^2)$ [9], that utilizes the shrinkage-thresholding operator. It was initially designed to solve ℓ_2-ℓ_1 problems with synthesis priors [19], mainly when $\mathbf{L}$, from (2), is an invertible matrix, or when one defines a specific dictionary in $\mathbf{D}$. Then, the minimization problem solved by FISTA can be written as:

$$\Psi(\mathbf{u}) = \frac{1}{2}\|\mathbf{g} - \mathbf{Au}\|_2^2 + \lambda\|\mathbf{u}\|_1, \tag{5}$$

where $\mathbf{A} = \mathbf{HL}^{-1}$, when $\mathbf{L}$ is an invertible matrix, or $\mathbf{A} = \mathbf{HD}$, with $\mathbf{D}$ as a dictionary [20]. The coefficient vector $\mathbf{u}$ is expected to be a sparse representation of the image, e.g., a sparse wavelet representation.

The shrinkage-thresholding operator, or soft-thresholding operator [21], arises from the proximal operator of the ℓ_1-norm [22], and is given by:

$$S_\lambda(y_i) = \begin{cases} 0 & , |y_i| \leq \lambda \\ y_i - \lambda \mathrm{sign}(y_i) & , \text{otherwise.} \end{cases} \tag{6}$$

When utilized to solve (5) with FISTA, it results in the following iteration:

$$\mathbf{u}_k = \mathbf{S}_{\lambda/c}\left(\frac{1}{c}\mathbf{A}^T(\mathbf{g} - \mathbf{A}\mathbf{y}_k) + \mathbf{y}_k\right), \tag{7}$$

where $\mathbf{S}_\lambda(\mathbf{y}) = [S_\lambda(y_1), \ldots, S_\lambda(y_N)]^T$. Note that convergence is only guaranteed [9] if $c \geq ||\mathbf{A}^T\mathbf{A}||_2$. Also, the higher the c value the lower the convergence speed of FISTA.

The point $\mathbf{y}_{k+1}$ in (7) is a special combination of the two previous approximations:

$$\mathbf{y}_{k+1} = \mathbf{u}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{u}_k - \mathbf{u}_{k-1}), \tag{8}$$

where $t_{k+1} = \left(1 + \sqrt{1 + 4t_k^2}\right)/2$, for $t_1 = 1$ and $\mathbf{u}_0 = \mathbf{y}_1$.

FISTA is not a monotone algorithm [11]. Due to this non-monotonicity the authors of [9] proposed the monotone FISTA (MFISTA) [11]. Here, the step in equation (7) is changed to:

$$\mathbf{z}_k = \mathbf{S}_{\lambda/c}\left(\frac{1}{c}\mathbf{A}^T(\mathbf{g} - \mathbf{A}\mathbf{y}_k) + \mathbf{y}_k\right), \tag{9}$$

and the actual approximate solution at iteration k is given by:

$$\mathbf{u}_k = \arg\min_{\mathbf{z}}\{\Psi(\mathbf{z}) | \mathbf{z} \in \{\mathbf{z}_k, \mathbf{u}_{k-1}\}\}. \tag{10}$$

This means $\mathbf{z}_k$, from (9), is the new approximation only if it reduces the value of Ψ compared to previous iteration $\mathbf{u}_{k-1}$. In MFISTA, the computation of $\mathbf{y}_{k+1}$ is modified to:

$$\mathbf{y}_{k+1} = \mathbf{u}_k + \frac{t_k}{t_{k+1}}(\mathbf{z}_k - \mathbf{u}_k) + \frac{t_k - 1}{t_{k+1}}(\mathbf{u}_k - \mathbf{u}_{k-1}). \tag{11}$$

In [12] the over-relaxed and monotone FISTA (OMFISTA) was proposed. This modified member of the FISTA family allows the use of over-relaxation, with two variable steps. The equation (9) is modified to include relaxation in OMFISTA:

$$\mathbf{z}_k = \mathbf{S}_{\lambda\beta_k/c}\left(\frac{\beta_k}{c}\mathbf{A}^T(\mathbf{g} - \mathbf{A}\mathbf{y}_k) + \mathbf{y}_k\right), \tag{12}$$

and also (10) changed to:

$$\mathbf{u}_k = \arg\min\{\Psi(\mathbf{z}) | \mathbf{z} \in \{\mathbf{u}_{k-1} + \alpha_k(\mathbf{z}_k - \mathbf{u}_{k-1}), \mathbf{u}_{k-1}\}\}, \tag{13}$$

where the two new step parameters β_k and α_k may be different from $\beta_k = \alpha_k = 1$, as in MFISTA. The computation of $\mathbf{y}_{k+1}$ is also modified from (11) to:

$$\begin{aligned}\mathbf{y}_{k+1} &= \mathbf{u}_k + \frac{t_k}{t_{k+1}}(\mathbf{z}_k - \mathbf{u}_k) + \frac{t_k - \alpha_1}{t_{k+1}}(\mathbf{u}_k - \mathbf{u}_{k-1}) \\ &\quad + \frac{t_k}{t_{k+1}}\left(1 - \eta_k\alpha_k\beta_k^{-1}\right)(\mathbf{y}_k - \mathbf{z}_k), \end{aligned} \tag{14}$$

where, according to [12], the following conditions need to be satisfied for convergence:

$$\begin{aligned} &\eta_k \leq \eta_{k-1}, \\ &\beta_k(\beta_k - 2) + \eta_k\alpha_k \leq 0, \\ &0 \leq \alpha_k \leq 1, \\ &\beta_k \geq 0 \end{aligned} \tag{15}$$

Also $t_1 = \alpha_1$ and:

$$t_{k+1} = \frac{\alpha_1 \alpha_{k+1} + \sqrt{\alpha_1^2 \alpha_{k+1}^2 + 4t_k^2}}{2}. \tag{16}$$

These new parameters make OMFISTA more complex, but also give it flexibility and possibly better performance. Conditions (15) are sufficient, in practice we can extrapolate them a bit and get even more speed, as done in [15].

The most relevant point of this algorithm is that $\mathbf{u}_{k-1} + \alpha_k(\mathbf{z}_k - \mathbf{u}_{k-1})$ may be the next solution instead of $\mathbf{z}_k$. This allows the user to choose a step α_k different from a unitary one. This modification opens the possibility of using line search to choose a better α_k. It is also possible to decouple the α_k used in (16) and the step used (13), which we call hereafter μ_k, being the next solution defined as $\mathbf{u}_{k-1} + \mu_k(\mathbf{z}_k - \mathbf{u}_{k-1})$. Even though μ_k and α_k are originally the same value, according to the convergence analysis in [12], and our extension in the Appendix for $\alpha_k > 1$, they need not be. This gives freedom to use line search on OMFISTA without extrapolating conditions (15), or up to the new limit proposed in (33) of this paper.

While line search acceleration has already been employed for iterative shrinking-threshold algorithms (ISTA), as reported in [23], only recently it was applied to OMFISTA[1] [15] and, as far as we know, it has not been applied to MFISTA until now.

A. FISTA for Analysis Prior and Fast Gradient Projection

While the shrinkage-thresholding operator is of fast computation, it only provides the exact proximal operator for synthesis problems. For analysis priors [19], the proximal operator may not have a closed form solution. Due to this, the authors of FISTA extended it in [11] for analysis priors such as total variation, and also included positivity constraints.[2] This was done through the use of the dual formulation of the analysis prior from [24]. Furthermore, in [11] the speed of convergence the gradient projection (GP) method was improved including the same momentum acceleration used in FISTA, a technique denominated there as fast GP (FGP).

The FISTA-FGP algorithm, for minimizing the problem (2) can be written as:

$$\mathbf{f}_k = \mathrm{FGP}_{\lambda/\tau,\mathbf{L}}\left(\frac{1}{\tau}\mathbf{H}^T(\mathbf{g} - \mathbf{H}\mathbf{y}_k) + \mathbf{y}_k\right). \tag{17}$$

Here, $\mathrm{FGP}_{\lambda,\mathbf{L}}(\mathbf{b})$ is an algorithm to compute the proximal of $||\mathbf{Lf}||_1$ given by:

$$\mathrm{FGP}_{\lambda,\mathbf{L}}(\mathbf{b}) = \arg\min_{\mathbf{f}} \frac{1}{2}||\mathbf{b} - \mathbf{f}||_2^2 + \lambda||\mathbf{Lf}||_1 = \mathrm{prox}_{\lambda||\mathbf{Lf}||_1}(\mathbf{b}). \tag{18}$$

In (17), the point $\mathbf{y}_{k+1}$ is computed as:

$$\mathbf{y}_{k+1} = \mathbf{f}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{f}_k - \mathbf{f}_{k-1}). \tag{19}$$

The most important difference between FISTA and FISTA-FGP is that the soft-thresholding operator, which is the proximal operator for ℓ_1 norm of the coefficients, i.e $||\mathbf{u}||_1$, is replaced by the proximal operator of the analysis prior, i.e $||\mathbf{Lf}||_1$, solved using FGP.

The main iteration of the FGP is given by:

$$\mathbf{p}_k = \mathcal{P}\left(\frac{1}{\lambda s}\mathbf{L}(\mathbf{b} - \lambda\mathbf{L}^T\mathbf{r}_k) + \mathbf{r}_k\right), \tag{20}$$

where the next iteration point $\mathbf{r}_{k+1}$ is given by:

$$\mathbf{r}_{k+1} = \mathbf{p}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{p}_k - \mathbf{p}_{k-1}), \tag{21}$$

and the output of the algorithm $\mathbf{f} = \mathrm{FGP}_{\lambda,\mathbf{L}}(\mathbf{b})$, after N FGP iterations of (20) is given by:

$$\mathbf{f} = \mathbf{b} - \lambda\mathbf{L}^T\mathbf{p}_N. \tag{22}$$

In (20), t_k is the same as in FISTA, but $s \geq ||\mathbf{L}^T\mathbf{L}||_2$, and $\mathcal{P}(\mathbf{r})$ is an element-wise operator:

$$\mathcal{P}(r_i) = \frac{r_i}{\max\{1, |r_i|\}}. \tag{23}$$

Note that the soft-thresholding operator could be implemented using FGP algorithm too, assuming $\mathbf{L}$ as an identity matrix. However, the standard form in (6) is much more efficient. In [11], FGP was combined with FISTA and MFISTA. In this paper, we use FGP in OMFISTA as well, and most important, line search can be applied to these methods to accelerate them.

III. Line Search for ℓ_1-Norm Regularized Least Squares Problems

The line search for the problem (2) can be written as:

$$\mu^* = \arg\min_{\mu} \Psi(\mu), \tag{24}$$

with

$$\Psi(\mu) = \frac{1}{2}\|\mathbf{g} - \mathbf{H}(\mathbf{f} + \mu\mathbf{d})\|_2^2 + \lambda\|\mathbf{L}(\mathbf{f} + \mu\mathbf{d})\|_1, \tag{25}$$

where $\mathbf{d}$ is the search direction and μ the step. The k index was omitted for visual convenience. Equation (25) can be written as:

$$\Psi(\mu) = \sum_{j=1}^{M} \frac{q_j}{2}|\mu - x_j|^2 + \sum_{i=1}^{N} \omega_i|\mu - \mu_i|, \tag{26}$$

with:

$$x_j = \frac{g_j - \mathbf{h}_j^T\mathbf{f}}{\mathbf{h}_j^T\mathbf{d}}, \quad q_j = |\mathbf{h}_j^T\mathbf{d}|^2 = (\mathbf{h}_j^T\mathbf{d})^2,$$
$$\mu_i = -\frac{\mathbf{l}_i^T\mathbf{f}}{\mathbf{l}_i^T\mathbf{d}}, \quad \omega_i = \lambda|\mathbf{l}_i^T\mathbf{d}| = \lambda\sqrt{(\mathbf{l}_i^T\mathbf{d})^2}, \tag{27}$$

where $\mathbf{h}_j^T$ is the j-th row of $\mathbf{H}$, and $\mathbf{l}_i^T$ is the i-th row of $\mathbf{L}$, and g_j is the j-th element of the vector $\mathbf{g}$. Equation (26) is valid to represent (25) or (5), where $\mathbf{f}$, $\mathbf{H}$ and $\mathbf{L}$ in (27) have to be replaced by $\mathbf{u}$, $\mathbf{A}$ and $\mathbf{I}$ respectively.

Arranging the line search problem as (26) allows us to visualize it as a simpler one-dimensional optimization problem [16]. In this case, standard iterative one-dimensional optimization methods such as the method of false position (MFP) [25] and specific fast methods such as [13] and [16] can be used to solve (26).

[1] Example 2(1) of the paper [12].

[2] Paper [11] considers general convex constrains.

The MFP [25] is an iterative line search that starts with two extreme points μ_a and μ_b, such that $\Psi'(\mu_a) \leq 0 \leq \Psi'(\mu_b)$. The next point is given by:

$$\mu_{new} = \mu_a - \Psi'(\mu_a)\left[\frac{\mu_b - \mu_a}{\Psi'(\mu_b) - \Psi'(\mu_a)}\right]. \tag{28}$$

Here, $\Psi'(\mu)$ is defined as one element of the subdifferential $\partial\Psi(\mu)$ [16]. Then μ_{new} replaces μ_b if $\Psi'(\mu_{new}) > 0$, or it replaces μ_a if $\Psi'(\mu_{new}) < 0$.

While this line search provides a good approximate solution at low cost usually with few iterations, it may take much longer to find an exact minimum solution, specially if μ^* is one of the μ_i points. Also, care should be taken with a possible singularity. Therefore, if there exists a method that finds the exact minimum within a relatively short time, then, finding this exact minimum could be worthwhile.

In [13], a method to find the minimum of a generalized version of (26) was developed for denoising problems. According to [13], the exact minimum of (26) is given by:

$$\mu^* = \text{median}(\mu_1, \ldots, \mu_N, p_0, \ldots, p_N) \tag{29}$$

where p_i is computed as:

$$p_i = x_q - \frac{\Omega_i}{Q}, \tag{30}$$

with $x_q = \frac{1}{Q}\sum_{j=1}^{M} q_j x_j$ and $Q = \sum_{j=1}^{M} q_j$. Considering an increasing ordering of μ_i, $\mu_1 \leq \mu_2 \leq \ldots \leq \mu_N$ and using the same indexes, the $\mathbf{\Omega}_i$'s are defined as:

$$\mathbf{\Omega}_i = \sum_{k=1}^{i} \omega_k - \sum_{k=i+1}^{N} \omega_k. \tag{31}$$

Starting with the lowest level $\mathbf{\Omega}_0 = -\sum_{i=1}^{N} \omega_i$, the $\mathbf{\Omega}_i$'s values form an increasing sequence. Note the p_i's are decreasingly ordered, $p_N \leq \ldots \leq p_1 \leq p_0$. Li and Osher [13] proved that the minimum of (26) is the median of a set composed by the N values of $\{\mu_i\}$ and the $N+1$ values of $\{p_i\}$.

This method is relatively fast, the most computationally expensive parts are (a) sorting $\{\mu_i\}$, requiring on average[3] $O(N \log N)$ operations, (b) computing the $\mathbf{\Omega}_i$ values, as $\mathbf{\Omega}_i = \mathbf{\Omega}_{i-1} + 2\omega_i$, using $O(N)$ operations, and (c) computing the median in (29), using on average $O(2N \log 2N)$ operations.

In [16], an alternative implementation, with lower computational cost than [13] for problem (24) was proposed. The algorithm in [16] basically computes sequentially the extreme values[4] of $\partial\Psi(\mu_i)$, looking for the root of $\partial\Psi(\mu)$, which may be an μ_i or a p_i. On average, the method presented in [16] is 1.5 times faster than the line search technique proposed in [13].

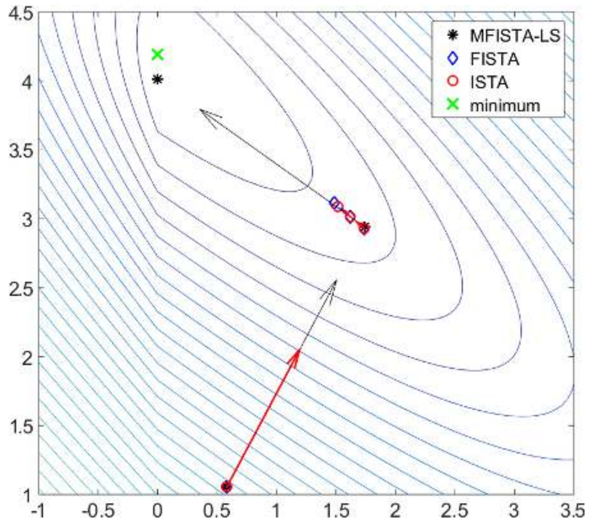


Fig. 1. Evolution of the algorithms according to the iterations. FISTA-LS can have bigger steps than ISTA or FISTA, getting close to the minimum faster.

IV. Accelerating MFISTA and OMFISTA With Line Search

While fast and exact line search, such as those from [13] and [16], were shown to accelerate algorithms as ISTA and non-linear conjugated gradient for ℓ_2-ℓ_1 problems, only in [15] it was tested for FISTA family of methods, more precisely, in one version of the OMFISTA presented in [12]. However, here we show that line search can be applied to other members of this family as well.

A. Line Search on MFISTA

In FISTA and MFISTA, the inner step parameter c controls the speed of convergence, but it is limited due to $c \geq ||\mathbf{A}^T\mathbf{A}||_2$ in the synthesis form, and due to $\tau \geq ||\mathbf{H}^T\mathbf{H}||_2$ in the analysis form. These limits are important to ensure convergence with fixed step.

One can use line search in MFISTA, allowing convergence to be faster, basically finding a better point than the one specified by (10). Even though not explicit in [11], any $\mathbf{f}_k$ in which $\Psi(\mathbf{f}_k) \leq \min(\Psi(\mathbf{z}_k), \Psi(\mathbf{f}_{k-1}))$ can be used to replace the $\mathbf{z}_k$ or $\mathbf{f}_{k-1}$. In this sense, one can apply line search using the iteration (9), and optimizing $\Psi(\mathbf{f}_{k-1}+\mu_k(\mathbf{z}_k-\mathbf{f}_{k-1}))$, replacing the originally possible steps $\mu_k = 1$ or $\mu_k = 0$. This choice of the new solution can increase the speed of the MFISTA algorithm.

In Figure 1, we illustrate these accelerated steps comparing the first three iterations of ISTA, FISTA and MFISTA-LS in a two-dimensional example. One can clearly see that, in the second iteration, linear search places MFISTA-LS in a point much closer to the final optimizing limit point than FISTA or ISTA in the third iteration.

The MFISTA-FGP algorithm is described in Table I. Note that the synthesis version is nearly the same as in Table I, except that $\mathbf{f}_k$ must be replaced by $\mathbf{u}_k$, $\mathbf{H}$ replaced by $\mathbf{A}$, and $\mathbf{L}$ is replaced by the identity, while the FGP algorithm is simply changed to the shrinkage-thresholding operator.

The most computationally expensive parts of this algorithm are expected to be the left multiplications by matrix $\mathbf{H}$ and

[3] The quicksort algorithm requires at most $O(N^2)$ operations, but on average it uses only $O(N \log N)$ operations.

[4] Since $\partial\Psi(\mu_i)$ is a set, extreme values of the set, i.e the maximum value, is computed.

TABLE I

MFISTA-FGP-LS

```
$\mathbf{y}_1 = \mathbf{f}_0$
$\bar{\mathbf{f}}_0 = \mathbf{H}\mathbf{f}_0$
$\bar{\mathbf{y}}_1 = \bar{\mathbf{f}}_0$
$t_1 = 1$
$k = 1$
while(stopping condition=false)
    $\mathbf{z}_k = \mathrm{FGP}_{\lambda/\tau,\mathbf{L}}\left(\mathbf{y}_k + \frac{1}{\tau}\mathbf{H}^T(\mathbf{g} - \bar{\mathbf{y}}_k)\right)$
    $\mathbf{d}_k = \mathbf{z}_k - \mathbf{f}_{k-1}$
    $\bar{\mathbf{d}}_k = \mathbf{H}\mathbf{d}_k$
    $\bar{\mathbf{z}}_k = \bar{\mathbf{f}}_{k-1} + \bar{\mathbf{d}}_k$
    line search specific computations from Table II
    $\mathbf{f}_k = \mathbf{f}_{k-1} + \mu_k \mathbf{d}_k$
    $\bar{\mathbf{f}}_k = \bar{\mathbf{f}}_{k-1} + \mu_k \bar{\mathbf{d}}_k$
    $t_{k+1} = (1 + \sqrt{1 + 4t_k^2})/2$
    $\mathbf{y}_{k+1} = \mathbf{f}_k + \frac{t_k - 1}{t_{k+1}}(\mathbf{f}_k - \mathbf{f}_{k-1}) + \frac{t_k}{t_{k+1}}(\mathbf{z}_k - \mathbf{f}_k)$
    $\bar{\mathbf{y}}_{k+1} = \bar{\mathbf{f}}_k + \frac{t_k - 1}{t_{k+1}}(\bar{\mathbf{f}}_k - \bar{\mathbf{f}}_{k-1}) + \frac{t_k}{t_{k+1}}(\bar{\mathbf{z}}_k - \bar{\mathbf{f}}_k)$
    $k = k + 1$
end for $k$
end while
```

TABLE II

LINE SEARCH SPECIFIC COMPUTATIONS

```
        $Q = \bar{\mathbf{d}}_k^T \bar{\mathbf{d}}_k$
        $x_q = (\mathbf{g} - \bar{\mathbf{f}}_{k-1})^T \bar{\mathbf{d}}_k / Q$
        for $i = 1$ to $N$
            $l_i = \mathbf{l}_i^T \mathbf{d}_k$ (also $= [\mathbf{L}\mathbf{d}_k]_i$)
            $\mu_i = -\mathbf{l}_i^T \mathbf{f}_{k-1} / l_i$
            $\omega_i = \lambda |l_i|$
        end for $i$
    $\mu_k = \text{line search}(x_q, Q, \{\mu_i, \omega_i\}_{i=1}^N)$
```

its adjoint. The computations using matrix $\mathbf{L}$ should be cheap, otherwise the FGP algorithm will be very costly.

One important remark here is that if c or τ are not known, then methods like backtracking can be utilized to check if the current c_k or τ_k values satisfy convergence conditions of FISTA, MFISTA or OMFISTA [9], [12], or if they need to be increased. Here we assume $c \geq ||\mathbf{A}^T\mathbf{A}||_2$ or $\tau \geq ||\mathbf{H}^T\mathbf{H}||_2$ can be computed.

B. Revisiting OMFISTA-LS

In [15], line search was applied to OMFISTA with successful results, but only on the synthesis version and with $\beta_k = 1$. Also, not all convergence conditions stated in (15) were strictly respected. In this paper, the general version of the OMFISTA is considered. The algorithm is detailed in Table III. Again, only the OMFISTA-FGP with line search is shown, but the non-FGP form is straightforward.

Even though good performance can be obtained respecting convergence conditions in (15), here we show that these theoretical limits can be increased. In [12], it is shown that:

$$\begin{aligned}\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}^*) &\\ \leq \frac{4\eta_1}{\eta_k(1 + \sum_{i=1}^k \alpha_k)^2}&[\Psi(\mathbf{f}_1) - \Psi(\mathbf{f}^*)\\ &+ \frac{1}{2\eta_1}\|\eta_1\alpha_1\beta_1^{-1}\mathbf{z}_1 + (1 - \eta_1\alpha_1\beta_1^{-1})\mathbf{y}_1 - \mathbf{f}^*\|^2]. \qquad (32)\end{aligned}$$

TABLE III

OMFISTA-FGP-LS

```
$\mathbf{y}_1 = \mathbf{f}_0$
$\bar{\mathbf{f}}_0 = \mathbf{H}\mathbf{f}_0$
$\bar{\mathbf{y}}_1 = \bar{\mathbf{f}}_0$
$k = 1$
$k_0 = k$
$\eta_0 = 1$
while(stopping condition = false)
    $\mathbf{z}_k = \mathrm{FGP}_{\beta_k\lambda/\tau,\mathbf{L}}\left(\mathbf{y}_k + \frac{\beta_k}{\tau}\mathbf{H}^T(\mathbf{g} - \bar{\mathbf{y}}_k)\right)$
    $\mathbf{d}_k = \mathbf{z}_k - \mathbf{f}_{k-1}$
    $\bar{\mathbf{d}}_k = \mathbf{H}\mathbf{d}_k$
    $\bar{\mathbf{z}}_k = \bar{\mathbf{f}}_k + \bar{\mathbf{d}}_k$
    line search specific computations from Table II
    $\mathbf{f}_k = \mathbf{f}_{k-1} + \mu_k \mathbf{d}_k$
    $\bar{\mathbf{f}}_k = \bar{\mathbf{f}}_{k-1} + \mu_k \bar{\mathbf{d}}_k$
    if $(k_0 = k)$ first iteration
        $t_k = \alpha_k$
    end if
    $\eta_k = \min(\eta_{k-1}, \beta_k(2 - \beta_k)/\alpha_k)$
    $t_{k+1} = (\alpha_{k_0}\alpha_{k+1} + \sqrt{\alpha_{k_0}^2\alpha_{k+1}^2 + 4t_k^2})/2$
    $\mathbf{y}_{k+1} = \mathbf{f}_k + \frac{t_k - \alpha_{k_0}}{t_{k+1}}(\mathbf{f}_k - \mathbf{f}_{k-1}) + \frac{t_k}{t_{k+1}}(\mathbf{z}_k - \mathbf{f}_k)$
        $+ \frac{t_k}{t_{k+1}}(1 - \eta_k\alpha_k\beta_k^{-1})(\mathbf{y}_k - \mathbf{z}_k)$
    $\bar{\mathbf{y}}_{k+1} = \bar{\mathbf{f}}_k + \frac{t_k - \alpha_{k_0}}{t_{k+1}}(\bar{\mathbf{f}}_k - \bar{\mathbf{f}}_{k-1}) + \frac{t_k}{t_{k+1}}(\bar{\mathbf{z}}_k - \bar{\mathbf{f}}_k)$
        $+ \frac{t_k}{t_{k+1}}(1 - \eta_k\alpha_k\beta_k^{-1})(\bar{\mathbf{y}}_k - \bar{\mathbf{z}}_k)$
    $k = k + 1$
end while
```

This result from [12] indicates that large α_k steps increase OMFISTA convergence speed. In the present work we note that we can use any α_k sequence that satisfies:

$$\alpha_k \leq \max\left(1, \frac{\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}_{k-1})}{\Psi(\mathbf{z}_k) - \Psi(\mathbf{f}_{k-1})}\right), \qquad (33)$$

This choice of α_k is required so the following relation holds:

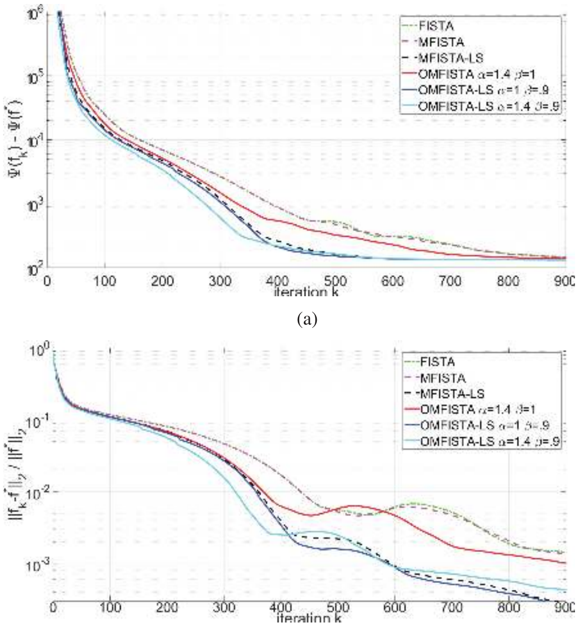
$$\Psi(\mathbf{f}_k) \leq (1 - \alpha_k)\Psi(\mathbf{f}_{k-1}) + \alpha_k\Psi(\mathbf{z}_k), \qquad (34)$$

This condition is always satisfied for $\alpha_k \leq 1$ because of (13) and convexity of $\Psi(\mathbf{f})$. However, we notice that larger steps α_k can increase the speed of convergence of OMFISTA. Convergence analysis of OMFISTA is extended in the Appendix to support (33), keeping most the analysis already provided in [12]. Note the improvement of OMFISTA for $\alpha_k > 1$ is independent of the line search. However, in the same way as MFISTA-LS, line search can give an extra boost on the convergence speed.

V. EXPERIMENTS

In this section, the MFISTA and the OMFISTA are compared with its line search-equipped counterparts, MFISTA-LS and OMFISTA-LS respectively. Unless otherwise stated, the line search procedure used was the one proposed in [16]. We focus here on tomographic image reconstruction. The purpose of this experiment is to illustrate the speed up of MFISTA and OMFISTA obtained with the use of line search.

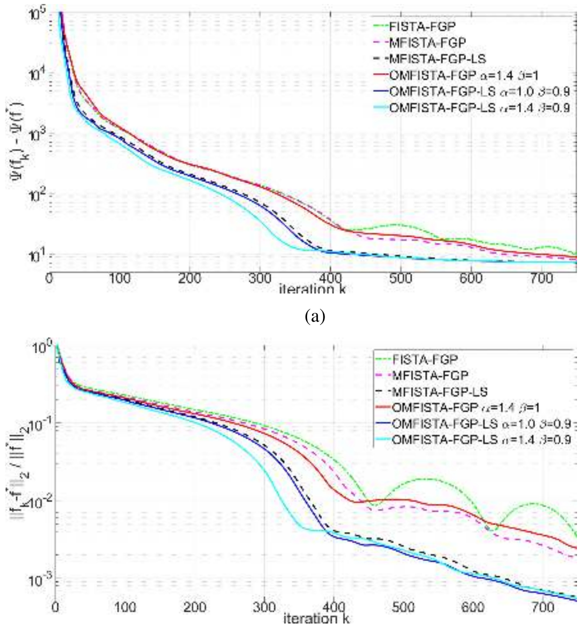
Two tomographic setups were used: For the wavelet (synthesis) reconstructions, a sinogram with 176 angles and



(a)

(b)

Fig. 2. Algorithms with wavelet regularization according to the iterations. (a) Cost function convergence. (b) Proximity with the optimal solution.



(a)

(b)

Fig. 3. Algorithms with finite difference regularization according to the iterations. (a) Cost function convergence. (b) Proximity with the optimal solution.

1024 values per angle from real data provided by the LNLS and a synthetic generated sinogram from Shepp-Logan phantom and Brain phantom[5] [26] with 256 angles and 1024 values per angle. The reconstructed images have 1024×1024 pixels. For finite difference (analysis) reconstructions, the same real data was utilized, but the phantom sinograms with 128 angles and 512 values per angle was utilized. The reconstructed images have 512×512 pixels. The NUFFT from [18] with $2\times$ oversampling and 6×6 interpolating filters was used to compute $\mathbf{Hf}$ and $\mathbf{H}^T\mathbf{g}$. The values of the cost function are computed at each iteration as well as the error between the solution and the minimum solution. The computing time is also recorded.

For this experiments, the λ was chosen by visually inspecting the reconstructed images, the same λ is used for all methods with synthesis prior, and a different one was used for all methods with analysis prior. The τ parameters were computed using $\|\mathbf{H}^T\mathbf{Hf}\|_2/\|\mathbf{f}\|_2$, where $\mathbf{f}$ is a constant intensity image, the computation of c followed the same idea using also the wavelet transform, and the same c and τ parameter values were used in all methods. In the OMFISTA tests, different parameters for β_k were tested, but they were kept constant during the iterative process. For synthesis prior, the sparsity is enforced in the wavelet domain, where Daubechies 4 wavelet was empirically chosen. For analysis prior, vertical and horizontal finite difference where utilized.

This experiment evaluates the evolution of the algorithms with respect to minimizing the cost function (2) plotted as $\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}^*)$ and the proximity with the minimum solution $\|\mathbf{f}_k - \mathbf{f}^*\|_2/\|\mathbf{f}^*\|_2$. The minimum value of the cost function $\Psi(\mathbf{f}^*)$ and the minimum point $\mathbf{f}^*$ are determined by choosing the point with minimum cost of all methods running 7 times more iterations than those plotted.

A. Overview of the Methods

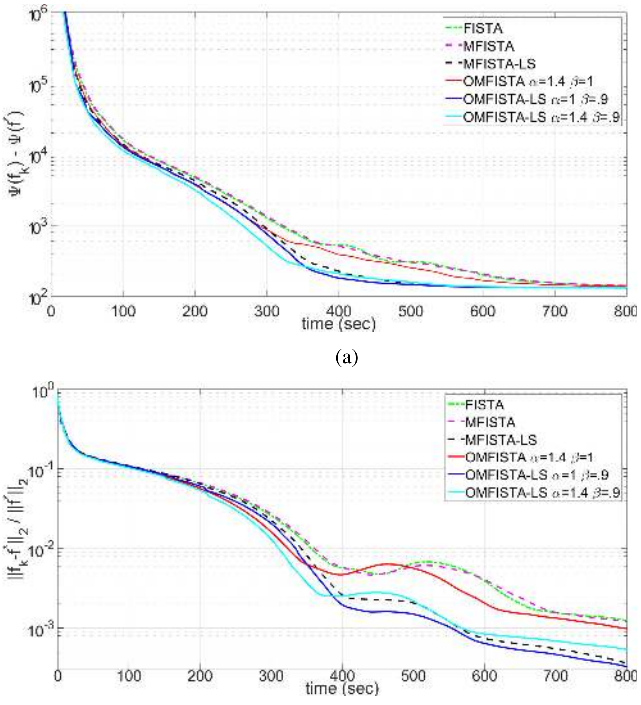
First, a direct comparison between FISTA, MFISTA, OMFISTA and the line search-equipped MFISTA-LS and OMFISTA-LS is shown in Figures 2 and 4 for wavelet (synthesis) prior, and in Figures 3 and 5 for finite difference (analysis) prior. Figures 2 and 3 show the convergence versus iteration count, where one can see the improvement per iteration achieved by the line search. In Figures 4 and 5 the convergence over processing time is shown, thus one can observe also a better performance when line search is utilized. These curves are averaged with the results for all phantoms.

In all eight plots, from Figure 2 to Figure 5, one can note some similar behavior with respect to acceleration. First, line search improved the method it was applied to, i.e. MFISTA-LS and MFISTA-FGP-LS were faster than MFISTA and MFISTA-FGP, as well as OMFISTA-LS and OMFISTA-FGP-LS were faster than OMFISTA and OMFISTA-FGP respectively. In all plots, OMFISTA-LS and OMFISTA-FGP-LS were the fastest methods, specially with $\alpha > 1$, and FISTA and FISTA-FGP were the slowest. Considering the convergence over iteration, on Figures 2 and 3, one can clearly see that line search improved speed in all methods. Table IV shows the average time[6] taken by each iteration of each method in the tested problems.

In Figure 9, some reconstructed images from synthetic data are shown. The unregularized image, reconstructed using filtered backprojection is shown in Figure 9(a) with

[5] Phantom based on a transaxial slice of the human head. The phantom mimics an actual medical image; for details, see sections 4.3 and 4.4 of [26]. The phantom has a random assignment of local inhomogeneities and a random introduction of small "tumors"; see section 5.2 of [26]. The 1024×1024 digitization of the phantom was produced by the software SNARK14 [27].

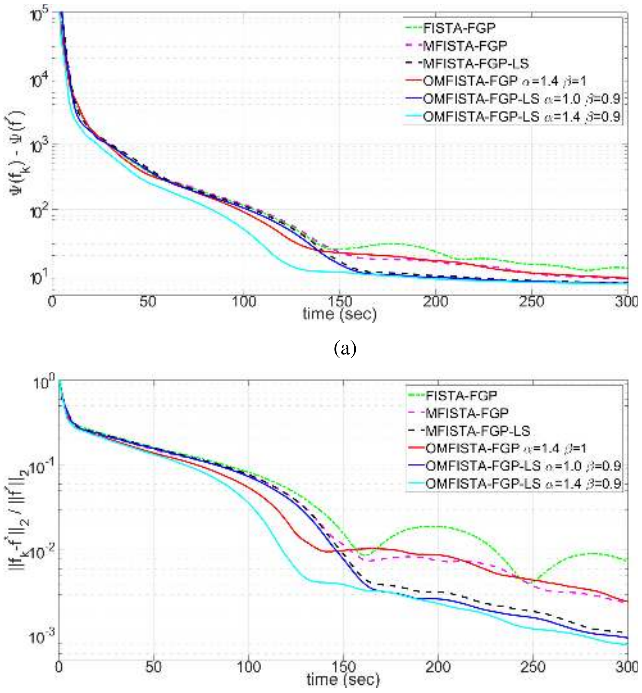
[6] Computational setup for numerical experiments: MATLAB R2015a, Windows 7, Intel Core i5-4570s @2.9GHz CPU, 32GB of DDR3-1333 Memory.



(a)

(b)

Fig. 4. Wavelet algorithms according to the processing time. (a) Cost function convergence with time. (b) Proximity with the optimal solution with time.



(a)

(b)

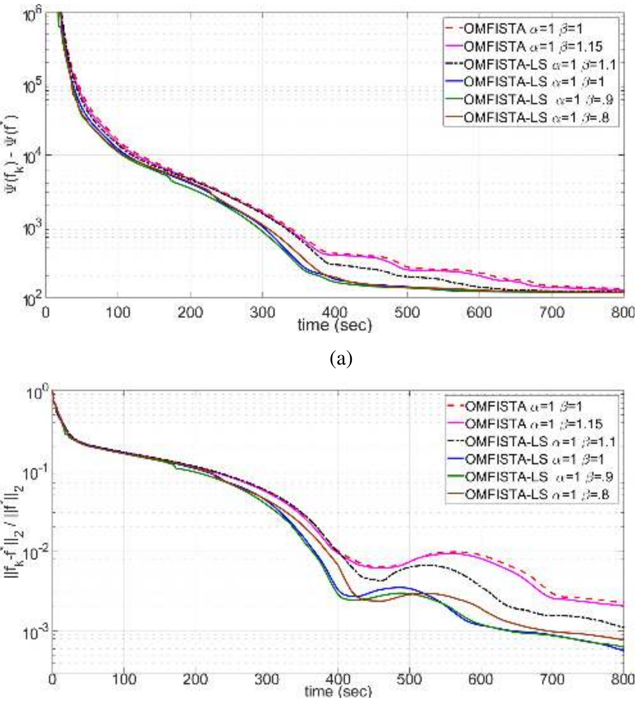
Fig. 5. Finite difference regularization methods according to the processing time. (a) Cost function convergence with time. (b) Proximity with the optimal solution with time.

PSNR = 7.7 dB [3]. The ℓ_1-norm regularized image, using wavelet transform (Daubechies 4), is shown in Figure 9(b) with PSNR = 26.2 dB, while regularized image with finite difference is shown in Figure 9(c) with PSNR = 31.0 dB. In Figure 10 some reconstructed images from real data from the LNLS are shown. There are no PSNR for these images since we have no ground truth for them.

TABLE IV

AVERAGE TIME PER ITERATION

Method	Wavelet (1024x1024)	Fin. Diff / FGP (512x512)
FISTA	0.82 sec	0.41 sec
MFISTA	0.83 sec	0.42 sec
OMFISTA	0.83 sec	0.42 sec
MFISTA-LS	0.96 sec	0.49 sec
OMFISTA-LS	0.96 sec	0.49 sec



(a)

(b)

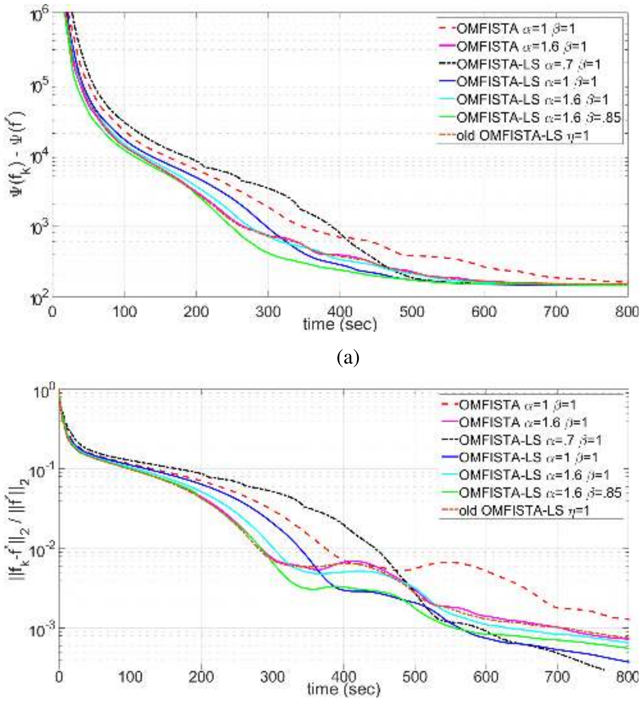
Fig. 6. Performance of OMFISTA with different β_k according to the processing time. (a) Cost function convergence with time. (b) Proximity with the optimal solution with time.

B. Exploiting OMFISTA Parameters

Many questions regarding the choice of OMFISTA parameters β_k and α_k can be raised. By now, there is no answer to which are the best parameters for α_k and β_k. However, in this section, we try to illustrate the performance of OMFISTA with different parameters so one can have an idea of how this choice affects the algorithm.

Regarding β_k, considering that $\beta_k = 1$ is the "neutral" parameter, we tested values closed to it. The results are shown in Figure 6 for Brain phantom only. Note that $\beta_k = 0.9$ was better than the others for OMFISTA-LS, $\beta_k = 1.15$ improved OMFISTA a bit.

Recall that theory from [12] requires $\alpha_k \leq 1$ for convergence guarantees. However, experimental practice here and in previous work demonstrates that this limit can be extended with improvements in convergence speed. In [15] we used $\alpha_k = \mu_k$ with some success, even though there is no convergence guarantee. Such results led us to take the effort to prove convergence with $\alpha_k > 1$, resulting in this OMFISTA-LS algorithm. Figure 7 illustrates its performance with different α_k choices, also, this OMFISTA-LS algorithm is compared with the old OMFISTA-LS from [15] with $\eta = 1$. The results seen in Figure 7 match well the theoretical predictions in which large α_k values should increase the convergence speed.



(a)

(b)

Fig. 7. Performance of OMFISTA with different α_k according to the processing time. (a) Cost function convergence with time. (b) Proximity with the optimal solution with time.

C. Performance Regarding the Line Search Algorithm

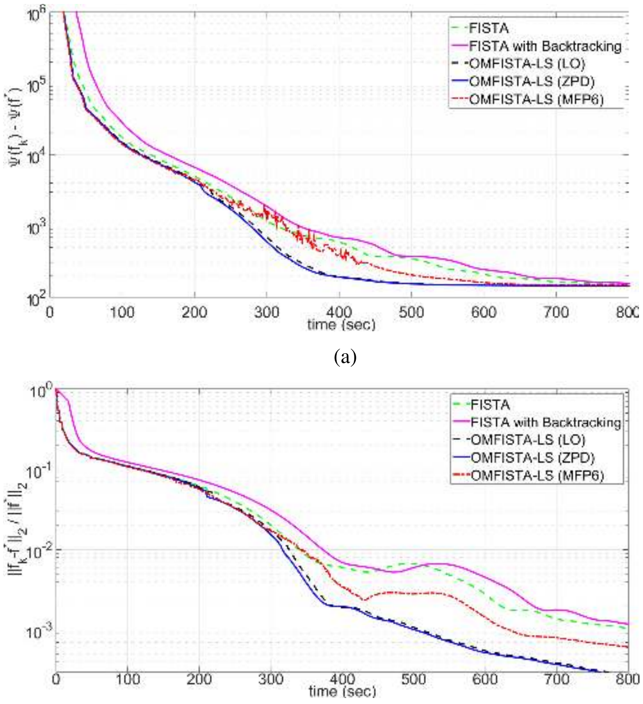
In this section, we compare the three fast line search methods discussed in section III. For this experiment, we denote as LO the method of Li and Osher, proposed in [13] for denoising, and adapted in [15] as a line search. This is a fast and exact one-dimensional optimization for ℓ_2-ℓ_1 problems. Another fast and exact line search tested here is denoted by ZPD, which is the method proposed in [16]. Also, we denote MFP6 as the method of false position [25] with six iterations. This is an iterative method for line search, and it did not provide the exact optimal point as the other two methods, but it provides a reasonably accurate solution. We also include a comparison of FISTA and FISTA with backtracking. Backtracking in FISTA is utilized to find a c that satisfy convergence conditions of the algorithm [9] when $c \geq ||\mathbf{A}^T\mathbf{A}||_2$ cannot be computed.

Figure 8 compares different line search methods for OMFISTA-LS with $\alpha = 1$ and $\beta_k = 0.9$. The main purpose of this test is to see if the line search method can make significant difference here as it did in [16] for ISTA, NLCG and IRLS. The conclusion is: very little difference between LO and ZPD, but MFP6 may be unreliable.

Note that, in Figure 8 the OMFISTA-LS with MFP6 got an instable behavior after 200 seconds, due to the precision of the estimated step. This is an example of why computing the optimal step, with LO or ZPD, is important. Also, backtracking usually makes FISTA a little slow since convergence conditions needs to be tested at each update of c_k or τ_k.

VI. Conclusions

This paper proposed an accelerating process through the use of fast line search for MFISTA and OMFISTA. Also,



(a)

(b)

Fig. 8. Comparison of the line search algorithms utilized in OMFISTA-LS. (a) Cost function convergence with time. (b) Proximity with the optimal solution with time.

the synthesis prior, using wavelet transform, and analysis prior, using anisotropic total variation, were considered in this paper. The analysis version is based on FGP. These algorithms were applied to tomographic image reconstruction using synthetic data and data provided by the LNLS. The experiments confirmed that line search made OMFISTA-LS and MFISTA-LS, as well as their FGP counterparts, faster than their original versions for CT problems. It is important to restate that OMFISTA-LS is more flexible than MFISTA-LS, and a proper combination of parameters may provide significant acceleration over MFISTA-LS.

Appendix
Convergence of OMFISTA

We need to further develop the theory from [12] in order to accept rule (33), which may take values larger than 1. We will start proving variations of [12, Lemmas 1(3) and 1(4)], but we will first introduce some notations. The problem we want to solve is a general form of (1) with:

$$\Psi(\mathbf{f}) = \psi(\mathbf{f}) + \rho(\mathbf{f})$$

where ψ is a smooth convex function with Lipschitz-continuous gradient $\nabla\psi$ and ρ is a proper lower semicontinuous convex function. We denote the Lipschitz constant of $\nabla\psi$ as $\mathcal{L}(\psi) \leq L$. Furthermore, let

$$Q_L(\mathbf{f}, \mathbf{g}) := \psi(\mathbf{g}) + \langle \nabla\psi(\mathbf{g}), \mathbf{f} - \mathbf{g}\rangle + \frac{L}{2}\|\mathbf{f} - \mathbf{g}\|^2 + \rho(\mathbf{f}),$$

$$\mathrm{prox}_{\beta\rho}(\mathbf{f}) := \operatorname*{argmin}_{\mathbf{g}} \left\{\rho(\mathbf{g}) + \frac{1}{2\beta}\|\mathbf{g} - \mathbf{f}\|^2\right\},$$

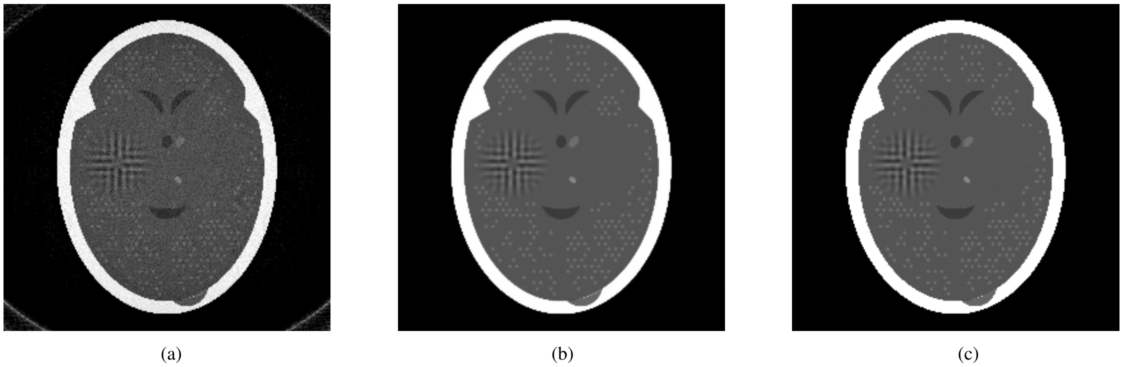


Fig. 9. Reconstructed images with synthetic data. The unregularized image has PSNR = 7.7 dB, the reconstructed ℓ_2-ℓ_1 images has PSNR = 26.2 dB for wavelet and PSNR = 31.0 dB for finite difference. (a) Unregularized image. (b) Reconstructed ℓ_2-ℓ_1 image with wavelet regularization. (c) Reconstructed ℓ_2-ℓ_1 image with finite difference regularization.

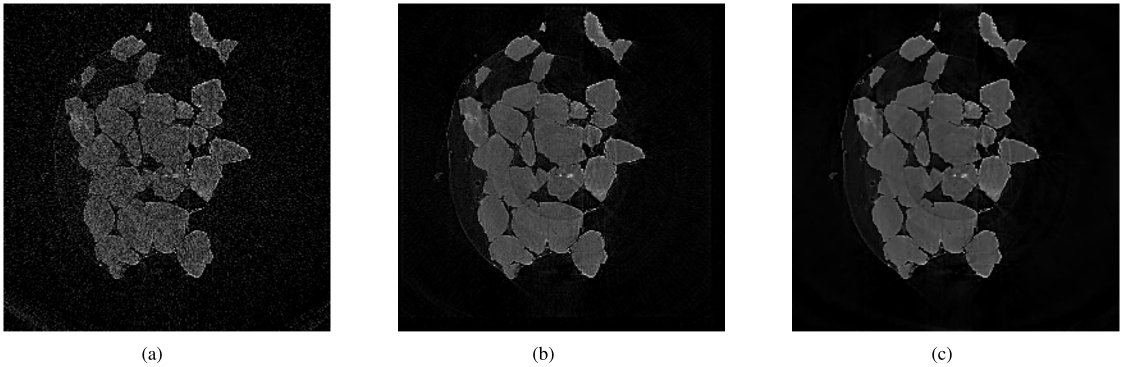


Fig. 10. Reconstructed images with real data from the LNLS. (a) Unregularized image. (b) Reconstructed ℓ_2-ℓ_1 image with wavelet regularization. (c) Reconstructed ℓ_2-ℓ_1 image with finite difference regularization.

and

$$\theta_L(\mathbf{f}) := \operatorname{prox}_{\frac{1}{L}\rho}\left(\mathbf{f} - \frac{1}{L}\nabla\psi(\mathbf{f})\right).$$

We now can proceed to the proofs.[7]

Lemma 1: Let $\mathbf{y}$, $\mathbf{w}$, $\mathbf{u} \in \mathbb{R}^P$, L, β, *and* $\alpha > 0$ *such that*

$$\Psi\left(\theta_{\frac{1}{\beta}}(\mathbf{y})\right) \le Q_L\left(\theta_{\frac{1}{\beta}}(\mathbf{y}), \mathbf{y}\right)$$

and

$$\Psi(\mathbf{u}) \le (1-\alpha)\Psi(\mathbf{w}) + \alpha\Psi\left(\theta_{\frac{1}{\beta}}(\mathbf{y})\right). \tag{35}$$

Then for any $\mathbf{f} \in \mathbb{R}^P$ *we have*

$$\Psi(\mathbf{u}) \le (1-\alpha)\Psi(\mathbf{w}) + \alpha\Psi(\mathbf{f}) - \frac{\alpha}{\beta}\left\langle\theta_{\frac{1}{\beta}}(\mathbf{y}) - \mathbf{y}, \mathbf{y} - \mathbf{f}\right\rangle + \alpha\left(\frac{L}{2} - \frac{1}{\beta}\right)\left\|\theta_{\frac{1}{\beta}}(\mathbf{y}) - \mathbf{y}\right\|^2.$$

Proof: We notice that following the proof of [12, Lemma 1(3)] we can arrive at[8]

$$\Psi\left(\theta_{\frac{1}{\beta}}(\mathbf{y})\right) \le \Psi(\mathbf{f}) - \frac{1}{\beta}\left\langle\theta_{\frac{1}{\beta}}(\mathbf{y}) - \mathbf{y}, \mathbf{y} - \mathbf{f}\right\rangle + \left(\frac{L}{2} - \frac{1}{\beta}\right)\left\|\theta_{\frac{1}{\beta}}(\mathbf{y}) - \mathbf{y}\right\|^2.$$

Now use this at the right-hand side of (35). □

Lemma 2: Let the sequence $\{t_k\}$ *be given as in* (16) *with* $\alpha_k \ge 0$*, then the following hold*

$$\begin{aligned} t_k^2 &= t_{k+1}(t_{k+1} - \alpha_1\alpha_{k+1}), && (36)\\ t_k &\le t_{k+1}, \quad \textit{and} && (37)\\ t_k &\ge \frac{\alpha_1}{2}\left(2 + \sum_{i=2}^{k}\alpha_i\right). && (38) \end{aligned}$$

Proof: The first two statements can be verified directly from (16), while (38) can be verified by induction. In fact, it holds as an equality for $t_1 = \alpha_1$ and assuming that $t_k \ge \alpha_k/2\left(2 + \sum_{i=2}^{k}\alpha_i\right)$ we get

$$\begin{aligned} t_{k+1} &\ge \frac{\alpha_1}{2}\left(\alpha_{k+1} + \sqrt{\alpha_{k+1}^2 + \left(2 + \sum_{i=2}^{k}\alpha_i\right)^2}\right)\\ &\ge \frac{\alpha_1}{2}\left(\alpha_{k+1} + 2 + \sum_{i=2}^{k}\alpha_i\right). \end{aligned}$$

□

Theorem 1: Let $\mathbf{f}_0 \in \mathbb{R}^P$ *be given and define* $\mathbf{y}_1 := \mathbf{f}_0$*. Furthermore, suppose* $\{\alpha_k\} \subset [0, \infty)$*,* $\{\eta_k\} \subset (0, \infty)$*,* $\{L_k\} \subset (0, \infty)$*,* $\{\beta_k\} \subset (0, \infty)$*, and* $\{\mathbf{f}_k\} \subset \mathbb{R}^P$ *satisfy*

$$\begin{aligned} &\eta_{k+1} \le \eta_k, && (39)\\ &L_{k+1}\beta_{k+1}^2 - 2\beta_{k+1} + \eta_{k+1}\alpha_{k+1} \le 0, && (40)\\ &\Psi(\mathbf{z}_{k+1}) \le Q_{L_{k+1}}(\mathbf{z}_{k+1}, \mathbf{y}_{k+1}), \quad \textit{and} && (41) \end{aligned}$$

[7] In order to simplify the proofs and the theory already presented, what is β or β_k here corresponds to β_k/c or β_k/τ in sections II and IV, as well as η_k corresponds to η_k/c or η_k/τ. Also, for $\psi(\mathbf{f})$ such as $\frac{1}{2}\|\mathbf{g} - \mathbf{Hf}\|_2^2$, the L corresponds to c.

[8] We have the following correspondence of symbols: $N = P$, $F = \Psi$, $f = \psi$, $g = \rho$, $p = \theta$, $\boldsymbol{x} = \mathbf{f}$, $\boldsymbol{y} = \mathbf{y}$, $\boldsymbol{\omega} = \mathbf{w}$, $\boldsymbol{\xi} = \mathbf{u}$, $z = \mathbf{z}$, and $\lambda = \alpha$.

$$\Psi(\mathbf{f}_{k+1}) \leq (1-\alpha_{k+1})\Psi(\mathbf{f}_k) + \alpha_{k+1}\Psi(\mathbf{z}_{k+1}), \qquad (42)$$

where $\mathbf{z}_{k+1} := \theta_{\frac{1}{\beta_{k+1}}}(\mathbf{y}_{k+1})$ *and* $\{\mathbf{y}_{k+1}\}$ *is given by:*

$$\mathbf{y}_{k+1} := \mathbf{f}_k + \frac{t_k}{t_{k+1}}(\mathbf{z}_k - \mathbf{f}_k) + \frac{t_k - \alpha_1}{t_{k+1}}(\mathbf{f}_k - \mathbf{f}_{k-1}) + \frac{t_k}{t_{k+1}}(1 - \eta_k \alpha_k \beta_k^{-1})(\mathbf{y}_k - \mathbf{z}_k),$$

where $t_1 := \alpha_1$ *and the remaining* t_k *follow* (16). *Then we have:*

$$\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}^*) \leq \frac{4}{(2+\sum_{i=2}^{k}\alpha_i)^2}\frac{\eta_1}{\eta_k} \times \left[\Psi(\mathbf{f}_1) - \Psi(\mathbf{f}^*) + \frac{1}{2\eta_1}\left\| \frac{\eta_1\alpha_1}{\beta_1}\theta_{\frac{1}{\beta_1}}(\mathbf{y}_1) + \left(1 - \frac{\eta_1\alpha_1}{\beta_1}\right)\mathbf{y}_1 - \mathbf{f}^*\right\|\right].$$

Proof: Notice that condition (41) allows us to use Lemma 1 instead of [12, Lemma 1(3)] and proceed exactly like in the proof of [12, Th. 1] (using Lemma 2 when necessary) until we get:

$$\Psi(\mathbf{f}_k) - \Psi(\mathbf{f}^*) \leq \frac{\alpha_1^2}{t_k^2}\frac{\eta_1}{\eta_k} \times \left[\Psi(\mathbf{f}_1) - \Psi(\mathbf{f}^*) + \frac{1}{2\eta_1}\left\| \frac{\eta_1\alpha_1}{\beta_1}\theta_{\frac{1}{\beta_1}}(\mathbf{y}_1) + \left(1 - \frac{\eta_1\alpha_1}{\beta_1}\right)\mathbf{y}_1 - \mathbf{f}^*\right\|\right].$$

Then, using (38) we prove the claim. □

Acknowledgment

The tomographic data presented here was kindly provided by Dr. Eduardo X. Miqueles, from the IMX imaging beamline at the Brazilian Synchrotron Light Source (LNLS). The authors would also like to show their gratitude to Prof. Alvaro R. De Pierro and, again, Dr. Miqueles for sharing their pearls of wisdom with us during the course of this research, and for their cooperation in our early results in [15]. They thank Prof. Jeffrey Fessler and his team for making NUFFT codes freely available. Also, the authors are grateful for the technical cooperation and deep discussions provided by the Lineup for Imaging Inverse Problems, at UTFPR, as well as Prof. Gabor T. Herman of City University of New York. They also thank Prof. Herman and his team for making SNARK14 freely available.

References

[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[2] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[3] A. C. Bovik, *Handbook of Image and Video Processing*, 1st ed. San Diego, CA, USA: Academic, 2000.

[4] M. Persson, D. Bone, and H. Elmqvist, "Total variation norm for three-dimensional iterative reconstruction in limited view angle tomography," *Phys. Med. Biol.*, vol. 46, no. 3, pp. 853–866, 2001.

[5] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing MRI," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 72–82, Mar. 2008.

[6] H. L. Taylor, S. C. Banks, and J. F. McCoy, "Deconvolution with the ℓ_1 norm," *Geophysics*, vol. 44, no. 1, pp. 39–52, 1979.

[7] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[8] M. Elad, *Sparse and Redundant Representations From Theory to Applications in Signal and Image Processing*, Springer, 2010.

[9] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[10] Z. Zhang and V. Saligrama, "Rapid: Rapidly accelerated proximal gradient algorithms for convex minimization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 3796–3800.

[11] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[12] M. Yamagishi and I. Yamada, "Over-relaxation of the fast iterative shrinkage-thresholding algorithm with variable stepsize," *Inverse Problems*, vol. 27, no. 10, p. 105008, Oct. 2011.

[13] Y. Li and S. Osher, "A new median formula with applications to PDE based denoising," *Commun. Math. Sci*, vol. 7, no. 3, pp. 741–753, 2009.

[14] Z. Wen, W. Yin, D. Goldfarb, and Y. Zhang, "A fast algorithm for sparse reconstruction based on shrinkage, subspace optimization, and continuation," *SIAM J. Sci. Comput.*, vol. 32, no. 4, pp. 1832–1857, 2010.

[15] M. V. W. Zibetti, E. S. Helou, E. X. Miqueles, and A. R. De Pierro, "Accelerating the over-relaxed iterative shrinkage-thresholding algorithms with fast and exact line search for high resolution tomographic image reconstruction," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 2305–2308.

[16] M. V. W. Zibetti, D. R. Pipa, and A. R. De Pierro, "Fast and exact unidimensional L2–L1 optimization as an accelerator for iterative reconstruction algorithms," *Digit. Signal Process.*, vol. 48, pp. 178–187, Jan. 2016.

[17] A. Averbuch, R. R. Coifman, D. L. Donoho, M. Israeli, Y. Shkolnisky, and I. Sedelnikov, "A framework for discrete integral transformations II—The 2D discrete radon transform," *SIAM J. Sci. Comput.*, vol. 30, no. 2, pp. 785–803, Jan. 2008.

[18] S. Matej, J. A. Fessler, and I. G. Kazantsev, "Iterative tomographic image reconstruction using Fourier-based forward and back-projectors," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 401–412, Apr. 2004.

[19] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, Apr. 2007.

[20] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[21] M. Elad, M. A. T. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proc. IEEE*, vol. 98, no. 6, pp. 972–982, Jun. 2010.

[22] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 123–131, 2013.

[23] M. Zibulevsky and M. Elad, "L1-L2 optimization in signal and image processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 76–88, May 2010.

[24] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, no. 1, pp. 89–97, 2004.

[25] G. D. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd ed. Springer, 2008.

[26] T. G. Herman, *Advances in Pattern Recognition Fundamentals of Computerized Tomography: Image Reconstruction From Projections*, 2nd ed. London, U.K.: Springer, 2009.

[27] R. Davidi, G. T. Herman, O. Langthaler, S. Sardana, and Z. Ye. (2017). *SNARK14: A Programming System for the Reconstruction of 2D Images From 1D Projections*. [Online]. Available: http://dig.cs.gc.cuny.edu/software/snark14/SNARK14.pdf

Marcelo V. W. Zibetti received the Ph.D. degree in electrical engineering from the Federal University of Santa Catarina, Brazil, in 2007. He has been an Associate Professor with the Federal University of Technology-Paraná, Brazil, since 2008. He is currently a Researcher with the Center for Advanced Imaging Innovation and Research, New York University, USA.

Elias S. Helou received the Ph.D. degree in applied mathematics from the University of Campinas, Brazil, in 2009. He has been a Faculty Member of the Institute of Mathematics and Computer Sciences with the University of São Paulo at São Carlos, Brazil, since 2010.

Daniel. R. Pipa received the D.Sc. degree in electrical engineering from COPPE, Federal University of Rio de Janeiro, in 2012. He has been an Assistant Professor with the Federal University of Technology-Paraná, Brazil, since 2012.