A DUAL BLOCK COORDINATE PROXIMAL ALGORITHM WITH APPLICATION TO DECONVOLUTION OF INTERLACED VIDEO SEQUENCES

F. Abboud[1,2]*, E. Chouzenoux*[1]*, J.-C. Pesquet*[1]*, J.-H. Chenot*[2]*, and L. Laborelli*[2]

[1] Université Paris-Est, LIGM, UMR CNRS 8049, Champs sur Marne, France

[2] INA, Institut National de l'Audiovisuel. 94366 Bry sur Marne, France

ABSTRACT

Inverse problems encountered in video processing often require to minimize criteria involving a high number of variables. Among available optimization techniques, proximal methods have shown their efficiency in solving large scale possibly nonsmooth problems. When some of the proximity operators involved in these methods do not have closed form expressions, they may constitute a bottleneck in terms of computational complexity and memory requirements. In this paper, we address this problem and propose accelerated techniques for solving it. A new dual block-coordinate forward-backward algorithm computing the proximity operator of a sum of convex functions composed with linear operators is proposed and theoretically analyzed. The numerical performance of the approach is assessed through an application to deconvolution and super-resolution of interlaced video sequences.

Index Terms— Proximity operator, duality, forward-backward, convex optimization, block-coordinate approach, video processing, deconvolution, super-resolution, deinterlacing.

1. INTRODUCTION

A wide range of inverse problems in image processing requires to find a solution to a large scale optimization problem. In the context of video processing, one has to process massive datasets therefore it is of paramount importance to limit the memory and complexity requirements of optimization algorithms (see e.g. [1, 2]). In the last decades, there has been a growing interest in efficient convex optimization methods grounded on the use of the proximity operator [3, 4]. In these splitting methods, a minimizer of a sum of convex functions is obtained iteratively by evaluating, for each involved function, either its gradient (provided that the function is differentiable) or its proximity operator (especially when the function is nonsmooth). Among the class of algorithms of interest, one can mention the Forward-Backward algorithm (including iterative thresholding methods as special cases) [5], the Alternating Direction Method of Multipliers and its parallel extensions [6], the Douglas-Rachford algorithm and its parallel extensions [7, 8], the Generalized Forward-Backward algorithm [9], as well as more recent primal-dual strategies [10].

The proximity operator of a number of functions has a closed form expression [3, 11], but when the form of a function is more involved or, when the metric in which the proximity operator is computed is not the standard Euclidean one, one often needs to resort to iterative strategies in order to compute it. Attention must be paid to this problem since the overall computation cost of the optimization method becomes then strongly dependent on the efficiency of the subiterations implemented for computing the proximity operator.

In this paper, we focus on the computation of the proximity operator at a point $\widetilde{x} \in \mathbb{R}^N$ of a function $g = f + \sum_{j=1}^{J} h_j \circ A_j$, where f is a convex function from $\mathbb{R}^N$ to $]-\infty,+\infty]$ and, for every $j \in \{1,\ldots,J\}$, h_j is a convex function from $\mathbb{R}^{M_j}$ to $]-\infty,+\infty]$ and A_j is an $M_j \times N$ real-valued matrix. Although generic optimization algorithms can be used to compute the proximity operator of g, tailored algorithms have been proposed for this purpose. Among available techniques, the primal-dual splitting algorithm in [12] has the ability of handling a finite sum of functions composed with arbitrary linear operators, without requiring any matrix inversion step. This latter feature turns out to be a main advantage in video processing where these matrices often are of high dimensions. A special case of the algorithm in [12] is the Dykstra-like algorithm in [13], which constitutes an extension of a popular algorithm for finding the projection of a signal onto an intersection of convex constraint sets. An appealing strategy, initially proposed in the area of machine learning [14, 15], and recently applied to image restoration problems [16], is to combine these primal-dual techniques (sometimes called dual ascent methods) with a block-coordinate approach. In [14, 15], a stochastic dual block-coordinate Forward-Backward (FB) algorithm is proposed to compute the proximity operator of g in the particular case when, for every $j \in \{1,\ldots,J\}$, $M_j = 1$. A deterministic version of this algorithm, associated with a FISTA-like acceleration technique [17], is proposed in [16].

The research leading to these results has received funding from the EC Seventh Framework Programme (FP7/2007-2013) under Grant Agreement n^{o}600827

However, as we will see, few of the aforementioned block-coordinate algorithms fully exploit the flexibility offered by the dual formulation of the problem. Moreover, their convergence guarantees are somewhat limited, in the sense that only decay properties on the dual of the objective function are available. The main contribution of our paper is to bring this new class of algorithms in the arena of video processing. Another contribution is to propose a new form of the dual block FB algorithm for computing the proximity operator of function g for arbitrary matrices $(A_j)_{1 \leqslant j \leqslant J}$ and to establish the convergence of this algorithm for both the primal and dual iterates. It is worth noticing that the proposed algorithm can be practically accelerated thanks to the introduction of preconditioning matrices. In a similar manner, variable metric approaches have been found to be useful for accelerating the convergence of the standard FB algorithm in [18, 19, 20]. The numerical performance of the proposed method will be illustrated by means of simulations for deblurring and deinterlacing video sequences.

The rest of the paper is organized as follows: Section 2 introduces some useful optimization tools and the problem formulation. Section 3 describes the proposed algorithm and investigates its convergence properties. Section 4 addresses the application of this algorithm to deconvolution and super-resolution of interlaced video sequences.

2. PROBLEM STATEMENT

2.1. Proximity operator

Let us first introduce some notation which will be useful in the sequel. Let $B \in \mathbb{R}^{N \times N}$ be some positive definite matrix and let $\langle \cdot \mid \cdot \rangle$ denote the standard inner product of $\mathbb{R}^N$. We define the weighted norm $\| \cdot \|_B = \langle \cdot \mid B \cdot \rangle^{1/2}$, and its associated inner product $\langle \cdot \mid \cdot \rangle_B$. Let $\Gamma_0(\mathbb{R}^N)$ designate the class of lower-semicontinuous convex functions from $\mathbb{R}^N$ to $]-\infty, +\infty]$ with a nonempty domain. The computation of the proximity operator of $g \in \Gamma_0(\mathbb{R}^N)$ at $\widetilde{x} \in \mathbb{R}^N$ amounts to solving the following optimization problem:

$$\underset{x \in \mathbb{R}^N}{\text{minimize}} \;\; g(x) + \frac{1}{2}\|x - \widetilde{x}\|_B^2. \tag{1}$$

Problem (1) admits a unique solution denoted by $\text{prox}_{B,g}(\widetilde{x})$.

2.2. Dual formulation

Let us define the conjugate of a function $\psi \in \Gamma_0(\mathbb{R}^N)$ as

$$\psi^* : \mathbb{R}^N \to [-\infty, +\infty] : x \to \sup_{\nu \in \mathbb{R}^N} \left(\langle \nu | x \rangle - \psi(\nu) \right). \tag{2}$$

The dual problem associated to Problem (1) when B is equal to I_N (the identity matrix of $\mathbb{R}^N$) reads

$$\underset{(y^j)_{1 \leqslant j \leqslant J} \in \mathbb{R}^M}{\text{minimize}} \;\; \varphi\Big(-\sum_{j=1}^J A_j^\top y^j + \widetilde{x} \Big) + \sum_{j=1}^J h_j^*(y^j), \tag{3}$$

where $M = M_1 + \cdots + M_J$, φ is the Moreau envelope of parameter 1 of f^*, which has a nonexpansive (i.e. 1-Lipschitzian) gradient [21] and f^* (resp. h_j^*, $j \in \{1, \ldots, J\}$) is the conjugate function of f (resp. h_j, $j \in \{1, \ldots, J\}$). Under the following qualification condition, there always exists a solution to Problem (3) [21]:

Assumption 1 $\text{ri}\left(A(\text{dom}\, f)\right) \cap \text{ri}\,(\text{dom}\, h) \neq \varnothing$, *where* ri *denotes the relative interior of a set.*

3. PROPOSED ALGORITHM

We propose the following dual block preconditioned FB algorithm to solve the considered optimization problem.

Algorithm 1 Dual Block Preconditioned Forward-Backward

Initialization
- $B_j \in \mathbb{R}^{M_j \times M_j}$ with $B_j \succeq A_j A_j^\top, \quad \forall j \in \{1, \ldots, J\}$
- $\epsilon \in]0, 1]$, $(y_0^j)_{1 \leqslant j \leqslant J} \in \mathbb{R}^M$, $z_0 = -\sum_{j=1}^J A_j^\top y_0^j$.

For $n = 0, 1, \ldots$
- $\gamma_n \in [\epsilon, 2 - \epsilon]$
- $j_n \in \{1, \ldots, J\}$
- $x_n = \text{prox}_f(\widetilde{x} + z_n)$
- $\widetilde{y}_n^{j_n} = y_n^{j_n} + \gamma_n B_{j_n}^{-1} A_{j_n} x_n$
- $y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n B_{j_n}^{-1} \text{prox}_{\gamma_n B_{j_n}^{-1}, h_{j_n}} \left(\gamma_n^{-1} B_{j_n} \widetilde{y}_n^{j_n} \right)$
- $y_{n+1}^j = y_n^j, \quad \forall j \in \{1, \ldots, J\} \setminus \{j_n\}$
- $z_{n+1} = z_n - A_{j_n}^\top (y_{n+1}^{j_n} - y_n^{j_n})$.

The above algorithm performs, at each iteration $n \in \mathbb{N}$, the selection of a block index j_n defining the dual variable $y_n^{j_n}$ which is updated, while all the other dual variables with indices $j \neq j_n$ are kept unchanged. The update only involves matrix A_{j_n} and the proximity operator of function h_{j_n}, for which a proximal step is performed in a suitable metric defined by matrix B_{j_n}. This algorithm can be derived from the block-coordinate variable metric FB algorithm from [19] applied to the dual problem (3).

Although the preconditioning matrices $(B_j)_{1 \leqslant j \leqslant J}$ can be chosen in a flexible manner in Algorithm 1, a simpler (non-preconditioned) version of the algorithm is obtained by choosing, for every $j \in \{1, \ldots, J\}$, $B_j = \|A_j\|^2 I_{M_j}$. If, additionally, f is an ℓ_1 norm and, for every $j \in \{1, \ldots, J\}$ h_j is the indicator function of a singleton $\{b^j\}$ with $b^j \in \mathbb{R}^{M_j}$, an algorithm similar to the one studied in [22] is retrieved.

3.1. Particular case when $f = 0$

Let us discuss some connexions existing between Algorithm 1 and previous methods in the literature. First note that the existing works are restricted to the case when $f = 0$. We will make this assumption in this section. In this case, Algorithm 1 can be simplified as follows.

Algorithm 2 Dual Block Preconditioned FB when $f = 0$

Initialization
$B_j \in \mathbb{R}^{M_j \times M_j}$ with $B_j \succeq A_j A_j^\top, \quad \forall j \in \{1,\dots,J\}$
$\epsilon \in]0,1]$, $(y_0^j)_{1\leqslant j\leqslant J} \in \mathbb{R}^M$, $x_0 = \widetilde{x} - \sum_{j=1}^J A_j^\top y_0^j$.
For $n = 0, 1, \dots$
$\gamma_n \in [\epsilon, 2-\epsilon]$
$j_n \in \{1,\dots,J\}$
$\widetilde{y}_n^{j_n} = y_n^{j_n} + \gamma_n B_{j_n}^{-1} A_{j_n} x_n$
$y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n B_{j_n}^{-1} \text{prox}_{\gamma_n B_{j_n}^{-1}, h_{j_n}} \left(\gamma_n^{-1} B_{j_n} \widetilde{y}_n^{j_n}\right)$
$y_{n+1}^j = y_n^j, \quad \forall j \in \{1,\dots,J\} \setminus \{j_n\}$
$x_{n+1} = x_n - A_{j_n}^\top (y_{n+1}^{j_n} - y_n^{j_n})$.

It appears interesting to compare the previous algorithm with the one previously proposed in [12] which, in the absence of error terms and relaxation factor, reads:

Algorithm 3 Parallel Dual Block FB [12]

Initialization
$(\omega_j)_{1\leqslant j\leqslant J} \in]0,1]^J$ such that $\sum_{j=1}^J \omega_j = 1$,
$\beta \geqslant \max_{j\in\{1,\dots,J\}} \|A_j\|^2$,
$B_j = \beta \omega_j^{-1} I_{M_j}, \quad \forall j \in \{1,\dots,J\}$
$\epsilon \in]0,1]$, $(y_0^j)_{1\leqslant j\leqslant J} \in \mathbb{R}^M$, $x_0 = \widetilde{x} - \sum_{j=1}^J A_j^\top y_0^j$.
For $n = 0, 1, \dots$
$\gamma_n \in [\epsilon, 2-\epsilon]$
For $j = 1, \dots, J$
$\widetilde{y}_n^j = y_n^j + \gamma_n B_j^{-1} A_j x_n$
$y_{n+1}^j = \widetilde{y}_n^j - \gamma_n B_j^{-1} \text{prox}_{\gamma_n B_j^{-1}, h_j} \left(\gamma_n^{-1} B_j \widetilde{y}_n^j\right)$
$x_{n+1} = x_n - \sum_{j=1}^J A_j^\top (y_{n+1}^j - y_n^j)$.

Algorithms 2 and 3 exhibit some obvious similarities in their structure. It should be however pointed out that there exist two main differences between them. First, one can observe that the whole set of dual variables $(y_n^j)_{1\leqslant j\leqslant J}$ is updated in a parallel manner at iteration n of Algorithm 3, followed by an update of the primal variable x_n. Conversely, in Algorithm 2, the dual variables are updated one at a time, according to a coordinate ascent strategy, and after each update, the primal variable is also updated. The second difference is that the conditions required on matrices $(B_j)_{1\leqslant j\leqslant J}$ are less restrictive in the case of Algorithm 2. Indeed, the matrices involved in Algorithm 3 are such that, for every $j \in \{1,\dots,J\}$,

$$B_j \succeq \omega_j B_j = \beta I_{M_j} \succeq \|A_j\|^2 I_{M_j} \succeq A_j A_j^\top. \tag{4}$$

Since the preconditioning matrices $(B_j)_{1\leqslant j\leqslant J}$ usually play a critical role in the convergence speed of proximal algorithms [18, 19, 23], more freedom in their choice should be beneficial to the algorithm performance.

3.2. Convergence results

Convergence properties of iterates generated by the proposed Dual Block Preconditioned FB Algorithm 1 can be established under the additional assumption:

Assumption 2
(i) *The functions f and $(h_j)_{1\leqslant j\leqslant J}$ are semi-algebraic.*[1]
(ii) *For every $j \in \{1,\dots,J\}$, the restriction of h_j^* to its domain is continuous.*
(iii) *For every $j \in \{1,\dots,J\}$, matrix B_j is definite positive.*
(iv) *The sequence $(j_n)_{n\in\mathbb{N}}$ is chosen according to a quasi-cyclic rule, i.e. there exists $K \geqslant J$ such that, for every $n \in \mathbb{N}$, $\{1,\dots,J\} \subset \{j_n,\dots,j_{n+K-1}\}$.*

The following result can then be deduced from [19]:

Theorem 1 *Under Assumptions 1 and 2, if the sequence $(y_n)_{n\in\mathbb{N}} = \left((y_n^j)_{1\leqslant j\leqslant J}\right)_{n\in\mathbb{N}}$ is bounded, then this sequence converges to a solution to the dual problem* (3)*. In addition, the sequence $(x_n)_{n\in\mathbb{N}}$ converges to the proximity operator of g evaluated at $\widetilde{x}$.*

4. APPLICATION TO VIDEO RESTORATION

4.1. Observation model

Interlaced scan has been the main sampling pattern for TV recording, broadcasting, and displaying. This scan format was initially retained because of its efficiency with regard to the time response of the CRT screens and the persistence of the human visual system [24]. The recent development of HD flat LCD and plasma screens has raised the problem of video deinterlacing, i.e. computing the missing even or odd rows of each interlaced field to recover the initial progressive video sequences. With the development of higher quality displays, the difficulty of the problem increases since interlacing artefacts are more visible on large TV screens with high brightness and contrast [25]. In this part, we focus on the joint deblurring and deinterlacing problem where an interlaced frame sequence $(y_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^{TL}$ is related to the sought progressive sequence $(\overline{x}_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^{TN}$ through the model:

$$(\forall t \in \{1,\dots,T\}) \quad y_t = \mathrm{S}_t (h * \overline{x}_t) + w_t, \tag{5}$$

where T is the number of time frames, L (resp. $N = 2L$) is the number of pixels in each image of the interlaced (resp. progressive) sequence, $\mathrm{S}_t = \mathrm{S}_\mathrm{o}$ for odd values of t (resp. $\mathrm{S}_t = \mathrm{S}_\mathrm{e}$ for even values of t) is a row decimation operator, $h \in \mathbb{R}^P$ corresponds to a convolution kernel accounting for spatial blur, and $(w_t)_{1\leqslant t\leqslant T}$ is an unknown additive noise. An estimate of the original sequence can then be obtained by minimizing a penalized criterion $F = \Phi + \Psi$ where Φ is the least squares data fidelity term:

$$(\forall x \in \mathbb{R}^{TN}) \quad \Phi(x) = \frac{1}{2} \sum_{t=1}^T \|\mathrm{S}_t (h * x_t) - y_t\|^2, \tag{6}$$

[1] Semi-algebraicity is a property satisfied by a wide class of functions, which means that their graph is a finite union of sets defined by a finite number of polynomial inequalities.

and Ψ is a regularization term incorporating some a priori knowledge on the sought video sequence.

Here, we propose to define the regularization term as

$$(\forall x \in \mathbb{R}^{TN}) \quad \Psi(x) = \sum_{t=1}^{T} \Psi_t(x_t) + \mathrm{M}(x), \qquad (7)$$

where, for every $t \in \{1, \dots, T\}$, $\Psi_t = \eta \ \mathrm{sltv} + \iota_{[x_{\min}, x_{\max}]^N}$ with $\eta \in]0, +\infty[$, sltv is the semi-local total variation from [26], and ι_C denotes the indicator function of a set C. Moreover, M is a nonsmooth temporal regularization term which favors the similarity between successive video frames while taking into account motion estimation, similarly to the strategy proposed in [4] .

4.2. Optimization algorithm

A minimizer of the objective function F is computed by using the PALM algorithm recently proposed in [27]. In this method, each image x_t is updated in a sequential manner thanks to a FB iteration consisting of a gradient step on Φ with respect to x_t and a proximal step on the restriction to x_t of Ψ. Since the proximity operator of Ψ does not have an explicit form and involves discrete differences and motion related linear operators, we resort to a dual FB algorithm to compute it. More precisely, the performance of Algorithms 1, 2, and 3 are compared for performing this task. Note that, when implementing Algorithm 1, the function f is chosen equal to $\iota_{[x_{\min}, x_{\max}]^N}$.

4.3. Experimental results

To evaluate the performance of the methods, two video sequences are considered: the synthetic sequence **Claire** from http://www.cipr.rpi.edu/resource/sequences/cif.html, with $N = 256 \times 256$, from which $T = 14$ images are extracted and then blurred, interlaced, and corrupted with a white Gaussian noise resulting in an initial SNR value of 27.25 dB, and a real blurry interlaced video sequence extracted from a French broadcast archive programme **Au théâtre ce soir** provided by INA from which we read out $T = 10$ interlaced frames of size $L = 720 \times 288$. For both examples, the motion operators involved in M are computed over the observed sequences and then interpolated to reach the desired resolution. The SNR for the restored **Claire** sequence is equal to 30.90 dB. No ground truth is available for the INA recording, the good restoration quality on this example can be analysed through visual inspection of the images displayed in Figure 1.

Table 1 illustrates, for three different frames of the video sequences, the averaged time spent for the computation of prox_Ψ when running 100 iterations of PALM in association with either the proposed Algorithm 1, its variant in Algorithm 2, or Algorithm 3 from [12]. The three algorithms are stopped when $\|x_{n+1} - x_n\| \leqslant 5 \times 10^{-6} \|x_{n+1}\|$. The provided

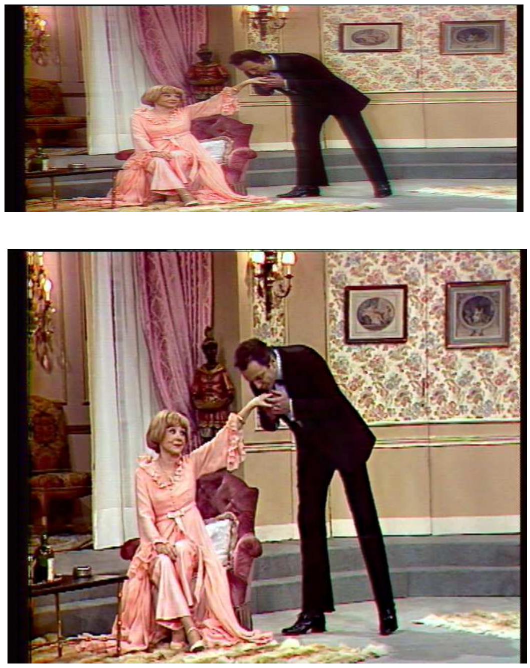

Fig. 1. 5th frame of the INA sequence: Noisy blurred interlaced field (top) and restored progressive image (bottom).

time values correspond to tests run on an Intel(R) Core(TM) i7-3770 CPU @ 3.4 GHz using a Matlab 7 implementation.

	Algorithm	1	2	3
Claire	Frame 1	2.50	**0.96**	5.71
	Frame 6	2.53	**0.82**	7.66
	Frame 13	2.51	**1.14**	7.08
Au théâtre ce soir	Frame 1	**1.98**	2.21	21.05
	Frame 5	2.08	**2.06**	28.14
	Frame 10	**1.89**	2.23	19.17

Table 1. Comparison between Algorithms 1, 2 and 3 in terms of computation time (in s.).

One can note that, depending on the example, the best performance in terms of computation time are obtained either with Algorithm 1 or Algorithm 2. On the opposite, the dual FB Algorithm 3 from [12] needs up to 10 more time to meet the stopping criterion.

5. CONCLUSION

This paper presents a new primal-dual splitting algorithm that handles efficiently the computation of the proximity operator for composite functions. The convergence of both its primal and dual iterates is guaranteed. The experimental results show the good performance of the proposed algorithm for deblurring and deinterlacing video sequences. Note that more sophisticated preconditioning strategies could be applied to further accelerate the convergence of the method.

6. REFERENCES

[1] Z. Xu, Gan Z., and Zhu X., "Compressed video super-resolution reconstruction based on regularized algorithm," in *6th IEEE International Conference of Signal Processing (ICSP 2006)*, Beijing, China, 16-20 Nov. 2006.

[2] X. Zhang, R. Xiong, S. Ma, and W. Gao, "A robust video super-resolution algorithm," in *28th IEEE Picture Coding Symposium (PCS 2010)*, Nagoya, Japan, 8-10 Dec. 2010, pp. 574–577.

[3] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds., pp. 185–212. Springer-Verlag, New York, 2010.

[4] F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot, and L. Laborelli, "A hybrid alternating proximal method for blind video restoration," in *22nd IEEE European Signal Processing Conference (EUSIPCO 2014)*, Lisbon, Portugal, 1-5 Sep. 2014, pp. 1811–1815.

[5] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Machine Learn.*, vol. 8, no. 1, pp. 1–122, 2011.

[7] R. I. Boţ and C. Hendrich, "A Douglas-Rachford type primal-dual method for solving inclusions with mixtures of composite and parallel-sum type monotone operators," vol. 23, no. 4, pp. 2541–2565, Dec. 2013.

[8] P. L. Combettes and J.-C. Pesquet, "Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping," 2014, http://www.optimization-online.org/DB_HTML/2014/04/4333.html.

[9] H. Raguet, M. J. Fadili, and G. Peyré, "A generalized forward-backward splitting," *SIAM J. Imaging Sci.*, vol. 6, no. 3, pp. 1199–1226, 2013.

[10] J.-C. Pesquet and A. Repetti, "A class of randomized primal-dual algorithms for distributed optimization," *J. Nonlinear Convex Anal.*, 2015.

[11] C. Chaux, J.-C. Pesquet, and N. Pustelnik, "Nested iterative algorithms for convex constrained image recovery problems," *SIAM J. Imaging Sci.*, vol. 2, no. 2, pp. 730–762, Jun. 2009.

[12] P. L. Combettes, D. Dũng, and B. C. Vũ, "Proximity for sums of composite functions," *J. Math. Anal. Appl.*, vol. 380, no. 2, pp. 680–688, Aug. 2011.

[13] H. H. Bauschke and P. L. Combettes, "A Dykstra-like algorithm for two monotone operators," *Pac. J. Optim*, vol. 4, no. 3, pp. 383–391, 2007.

[14] M. Jaggi, V. Smith, M. Takac, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, "Communication-efficient distributed dual coordinate ascent," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, Eds., pp. 3068–3076. Curran Associates, Inc., 2014.

[15] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 567–599, Feb. 2013.

[16] A. Chambolle and T. Pock, "A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions," Tech. Rep., 2015, http://www.optimization-online.org/DB_FILE/2015/01/4719.pdf.

[17] 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.

[18] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function," *J. Optim. Theory App.*, vol. 162, no. 1, pp. 107–132, Jul. 2014.

[19] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, "A block coordinate variable metric forward-backward algorithm," Tech. Rep., 2013, http://www.optimization-online.org/DB_HTML/2013/12/4178.html.

[20] S. Becker and J. Fadili, "A quasi-Newton proximal splitting method," in *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, Lake Tahoe, Nevada, 3-8 Dec. 2012, pp. 2627–2635.

[21] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, 2011.

[22] D. A. Lorenz, S. Wenger, F. Schöpfer, and M. A. Magnor, "A sparse Kaczmarz solver and a linearized Bregman method for online compressed sensing," Mar. 2014, http://arxiv.org/abs/1403.7543.

[23] A. Repetti, E. Chouzenoux, and J.-C. Pesquet, "A penalized weighted least squares approach for restoring data corrupted with signal-dependent noise," in *20th IEEE European Conference on Image and Signal Processing, (EUCIPCO 2012)*, Bucharest, Romania, 27-31 Aug. 2012, pp. 1553–1557.

[24] S. Mallat, "Super-resolution bandlet up-conversion for HDTV," Tech. Rep., 2006, http://www.di.ens.fr/∼mallat/papiers/whitepaper.pdf.

[25] S. H. Keller, *Video Upscaling Using Variational Methods*, Ph.D. thesis, The Image Group, Department of Computer Science Faculty of Science, University of Copenhagen, 2007.

[26] L. Condat, "Semi-local total variation for regularization of inverse problems," in *22nd IEEE European Conference on Image and Signal Processing, (EUSIPCO 2014)*, Lisbon, Portugal, 1-5 Sep. 2014, pp. 1806–1810.

[27] J. Bolte, S. Sabach, and M. Teboulle, "Proximal alternating linearized minimization for nonconvex and nonsmooth problems," *Math. Program.*, vol. 146, no. 1-2, pp. 459–494, Aug. 2014.