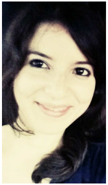
Proximity operator computation for video restoration.

Émilie Chouzenoux

Center for Visual Computing - INRIA Saclay, CentraleSupélec, France
Laboratoire d'Informatique Gaspard Monge - CNRS, Univ. Paris-Est Marne-la-Vallée, France

LCS Seminar
8 December 2017

In collaboration with



F. Abboud

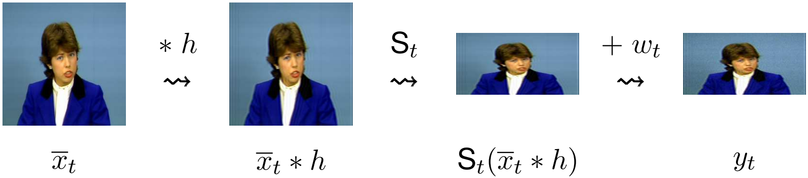
J.-C. Pesquet

J.-H. Chenot

L. Laborelli

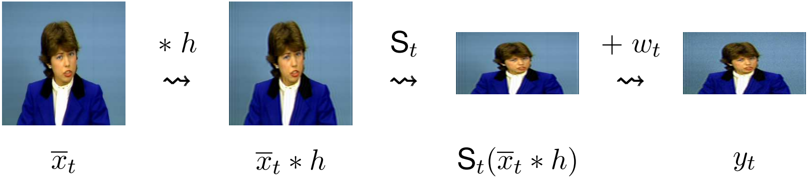
Institut National de l'Audiovisuel (INA)

Video restoration problem



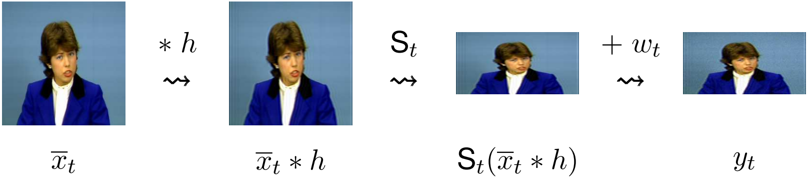
- $(\overline{x}_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$: original video sequence.

Video restoration problem



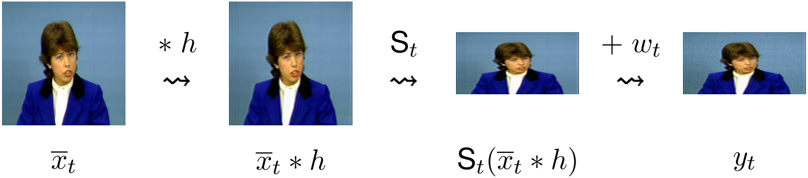
- $(\overline{x}_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$: original video sequence.
- $h \in \mathbb{R}^P$: convolution kernel.

Video restoration problem



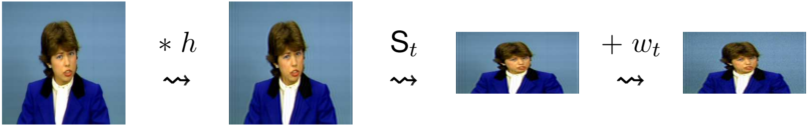
- $(\overline{x}_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$: original video sequence.
- $h \in \mathbb{R}^P$: convolution kernel.
- $\mathsf{S}_t \in \mathbb{R}^{L \times N}$: row decimation operator with $\mathsf{S}_t = \mathsf{S}_\mathrm{o}$ for odd values of t and $\mathsf{S}_t = \mathsf{S}_\mathrm{e}$ for even values of t.

Video restoration problem



- $(\overline{x}_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$: original video sequence.
- $h \in \mathbb{R}^P$: convolution kernel.
- $\mathsf{S}_t \in \mathbb{R}^{L \times N}$: row decimation operator with $\mathsf{S}_t = \mathsf{S}_\mathrm{o}$ for odd values of t and $\mathsf{S}_t = \mathsf{S}_\mathrm{e}$ for even values of t.
- $(w_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TL}$: unknown additive noise.

Video restoration problem



$\overline{x}_t$ $\quad$ $\overline{x}_t * h$ $\quad$ $\mathsf{S}_t(\overline{x}_t * h)$ $\quad$ y_t

- $(\overline{x}_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$: original video sequence.
- $h \in \mathbb{R}^P$: convolution kernel.
- $\mathsf{S}_t \in \mathbb{R}^{L \times N}$: row decimation operator with $\mathsf{S}_t = \mathsf{S}_\mathrm{o}$ for odd values of t and $\mathsf{S}_t = \mathsf{S}_\mathrm{e}$ for even values of t.
- $(w_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TL}$: unknown additive noise.
- $(y_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TL}$: interlaced blurred video sequence $(N = 2L)$.

Objective function

$$\underset{x\in\mathbb{R}^{TN}}{\text{minimize}} \quad F(x) = \Phi(x) + \Psi(x)$$

Φ ⇝ least squares data fidelity term:

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

Ψ ⇝ regularization term:

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

where $(\forall t \in \{1,\dots,T\})$ Ψ_t encourages spatial regularity and domain constraints on video frame x_t, and M is a temporal regularization term between neighboring frames.

Minimization strategy

$*$ Minimization of $F = \Phi + \Psi$ using the PALM approach: ([Bolte *et al.*, 2013])

Each image x_t is updated sequentially thanks to a forward-backward iteration combining:

1. a gradient step on Φ with respect to x_t,
2. a proximal step on the restriction to x_t of Ψ.

CONTENT OF THIS TALK:

How to compute the proximity operator of a convex function within a general metric, with limited memory and low computation time?
$\Rightarrow$ Exploit duality and preconditioned block alternation schemes!

Outline of the talk

1. Proximal operator
2. Dual forward-backward algorithms
   - Dual forward-backward algorithm
   - Block preconditioned DFB algorithm
   - Convergence results
3. Experimental results
4. Distributed strategy

Proximal operator

Notation and definitions

The set of symmetric definite positive matrices of $\mathbb{R}^{N\times N}$ will be denoted $\mathcal{S}^+(\mathbb{R}^N)$.

Let $U \in \mathcal{S}^+(\mathbb{R}^N)$. The weighted norm induced by U is

$$\|\cdot\|_U = \sqrt{\langle \cdot \mid U \cdot \rangle},$$

with the convention $\|\cdot\| = \|\cdot\|_{\mathrm{Id}}$.

The conjugate of a function $f\colon \mathbb{R}^N \to]-\infty,+\infty]$ is $f^*\colon \mathbb{R}^N \to [-\infty,+\infty]$ such that

$$(\forall u \in \mathbb{R}^N) \qquad f^*(u) = \sup_{x\in\mathbb{R}^N} \big(\langle x \mid u \rangle - f(x) \big).$$

Proximal operator http://proximity-operator.net

- Let $\Gamma_0(\mathbb{R}^N)$ denote the set of proper lsc convex functions from $\mathbb{R}^N$ to $]-\infty,+\infty]$.

The proximal operator $\mathrm{prox}_{U,f}(x)$ of $f \in \Gamma_0(\mathbb{R}^N)$ at $x \in \mathbb{R}^N$ relative to the metric induced by $U \in \mathcal{S}^+(\mathbb{R}^N)$ is the unique vector $\widehat{y} \in \mathbb{R}^N$ such that

$$f(\widehat{y}) + \frac{1}{2}\|\widehat{y} - x\|_U^2 = \inf_{y \in \mathbb{R}^N} f(y) + \frac{1}{2}\langle y - x \mid U(y - x)\rangle.$$

Proximal operator http://proximity-operator.net

The proximal operator $\operatorname{prox}_{U,f}(x)$ of $f \in \Gamma_0(\mathbb{R}^N)$ at $x \in \mathbb{R}^N$ relative to the metric induced by $U \in \mathcal{S}^+(\mathbb{R}^N)$ is the unique vector $\widehat{y} \in \mathbb{R}^N$ such that

$$f(\widehat{y}) + \frac{1}{2}\|\widehat{y} - x\|_U^2 = \inf_{y \in \mathbb{R}^N} f(y) + \frac{1}{2}\langle y - x \mid U(y - x)\rangle.$$

Characterization of proximal operator

$$(\forall x \in \mathbb{R}^N) \quad \widehat{y} = \operatorname{prox}_{U,f}(x) \Leftrightarrow x - \widehat{y} \in U^{-1}\partial f(\widehat{y}).$$

with ∂f the Moreau sub differential of f:

$$(\forall x \in \mathbb{R}^N) \quad \partial f(x) = \{t \in \mathbb{R}^N | (\forall y \in \mathbb{R}^N)\, f(y) \geqslant f(x) + \langle t | y - x\rangle\}.$$

Proximal operator http://proximity-operator.net

The proximal operator $\operatorname{prox}_{U,f}(x)$ of $f \in \Gamma_0(\mathbb{R}^N)$ at $x \in \mathbb{R}^N$ relative to the metric induced by $U \in \mathcal{S}^+(\mathbb{R}^N)$ is the unique vector $\widehat{y} \in \mathbb{R}^N$ such that

$$f(\widehat{y}) + \frac{1}{2}\|\widehat{y} - x\|_U^2 = \inf_{y \in \mathbb{R}^N} f(y) + \frac{1}{2}\langle y - x \mid U(y - x)\rangle.$$

Characterization of proximal operator

$$(\forall x \in \mathbb{R}^N) \quad \widehat{y} = \operatorname{prox}_{U,f}(x) \Leftrightarrow x - \widehat{y} \in U^{-1}\partial f(\widehat{y}).$$

Moreau's decomposition formula

$$(\forall x \in \mathbb{R}^N) \quad \operatorname{prox}_{U,f^*}(x) = x - U^{-1}\operatorname{prox}_{U^{-1},f}(Ux)$$

Properties of proximal operator

	$f(x)$	$\operatorname{prox}_f(x) = \operatorname{prox}_{\mathrm{Id}, f}(x)$
translation $z \in \mathbb{R}^N$	$f(x-z)$	$z + \operatorname{prox}_f(x-z)$
quadratic perturbation $z \in \mathbb{R}^N$, $\alpha > 0$, $\gamma \in \mathbb{R}$	$f(x) + \alpha\|x\|^2/2 + \langle x \mid z\rangle + \gamma$	$\operatorname{prox}_{\frac{f}{\alpha+1}}\left(\frac{x-z}{\alpha+1}\right)$
scaling $\rho \in \mathbb{R}^*$	$f(\rho x)$	$\frac{1}{\rho}\operatorname{prox}_{\rho^2 f}(\rho x)$
quadratic function $L \in \mathbb{R}^{M\times N}$, $\gamma > 0$, $z \in \mathbb{R}^M$	$\gamma\|Lx - z\|^2/2$	$(\mathrm{Id} + \gamma L L^\top)^{-1}(\mathsf{x} - \gamma L^* z)$
isomorphism $L \in \mathbb{R}^{M\times N}$, $LL^\top = \mu\mathrm{Id}$, $\mu > 0$	$f(Lx)$	$x - \mu^{-1}L^\top(x - \operatorname{prox}_{\mu f}(Lx)$
reflexion	$f(-x)$	$-\operatorname{prox}_f(-x)$
separability	$\sum_{n=1}^{N} \varphi_n(\mathsf{x}^{(n)})$ $x = (\mathsf{x}^{(n)})_{1\leqslant n\leqslant N}$	$\left(\operatorname{prox}_{\varphi_n}(\mathsf{x}^{(n)})\right)_{1\leqslant n\leqslant N}$
indicator function	$\iota_{\mathsf{C}}(x)$	$\mathsf{P}_{\mathsf{C}}(x)$
support function	$\iota^*_{\mathsf{C}}(x) = \sigma_{\mathsf{C}}(x)$	$x - \mathsf{P}_{\mathsf{C}}(x)$
composite function	$f(x) + \sum_{j=1}^{J} h_j(A_j x)$	**Iterative strategy!**

Properties of proximal operator

	$f(x)$	$\operatorname{prox}_f(x) = \operatorname{prox}_{\mathrm{Id},f}(x)$
translation $z \in \mathbb{R}^N$	$f(x-z)$	$z + \operatorname{prox}_f(x-z)$
quadratic perturbation $z \in \mathbb{R}^N$, $\alpha > 0$, $\gamma \in \mathbb{R}$	$f(x) + \alpha\|x\|^2/2 + \langle x \mid z\rangle + \gamma$	$\operatorname{prox}_{\frac{f}{\alpha+1}}\left(\frac{x-z}{\alpha+1}\right)$
scaling $\rho \in \mathbb{R}^*$	$f(\rho x)$	$\frac{1}{\rho}\operatorname{prox}_{\rho^2 f}(\rho x)$
quadratic function $L \in \mathbb{R}^{M\times N}$, $\gamma > 0$, $z \in \mathbb{R}^M$	$\gamma\|Lx - z\|^2/2$	$(\mathrm{Id} + \gamma L L^\top)^{-1}(\mathsf{x} - \gamma L^* z)$
isomorphism $L \in \mathbb{R}^{M\times N}$, $LL^\top = \mu \mathrm{Id}$, $\mu > 0$	$f(Lx)$	$x - \mu^{-1} L^\top (x - \operatorname{prox}_{\mu f}(Lx)$
reflexion	$f(-x)$	$-\operatorname{prox}_f(-x)$
separability	$\sum_{n=1}^{N} \varphi_n(\mathsf{x}^{(n)})$ $x = (\mathsf{x}^{(n)})_{1\leqslant n\leqslant N}$	$\left(\operatorname{prox}_{\varphi_n}(\mathsf{x}^{(n)})\right)_{1\leqslant n\leqslant N}$
indicator function	$\iota_C(x)$	$P_C(x)$
support function	$\iota_C^*(x) = \sigma_C(x)$	$x - P_C(x)$
composite function	$f(x) + \sum_{j=1}^{J} h_j(A_j x)$	**Iterative strategy!**

Which strategy in the context of large scale optimization ?

Dual forward-backward algorithms

Primal problem

PRIMAL PROBLEM

Compute $\mathrm{prox}_g(\widetilde{x})$ with $g = f + h \circ A$ and $\widetilde{x} \in \mathbb{R}^N$:

$$\underset{x \in \mathbb{R}^N}{\text{minimize}} \;\; f(x) + h(Ax) + \frac{1}{2}\|x - \widetilde{x}\|^2$$

where

- f belongs to $\Gamma_0(\mathbb{R}^N)$,
- h belongs to $\Gamma_0(\mathbb{R}^M)$,
- $A \in \mathbb{R}^{M \times N}$.

Qualification condition: $\mathrm{ri}\,\big(A(\mathrm{dom}\, f)\big) \cap \mathrm{ri}\,(\mathrm{dom}\, h) \neq \varnothing$.

Dual problem

Dual problem

$$\underset{y\in\mathbb{R}^M}{\text{minimize}} \;\; \varphi\Big(-A^\top y+\widetilde{x}\Big)+h^*(y),$$

- $\varphi = \left(f+\frac{1}{2}\|\cdot\|^2\right)^*$ is the Moreau envelope of parameter 1 of f^* with a nonexpansive (i.e. 1-Lipschitzian) gradient.

Dual problem

DUAL PROBLEM

$$\underset{y\in\mathbb{R}^M}{\text{minimize}}\ \varphi\Big(-A^\top y+\widetilde{x}\Big)+h^*(y),$$

- $\varphi = \left(f + \frac{1}{2}\|\cdot\|^2\right)^*$ is the Moreau envelope of parameter 1 of f^* with a nonexpansive (i.e. 1-Lipschitzian) gradient.

$\Rightarrow$ we can apply the forward-backward algorithm:

Initialization
$\left\lfloor\ \beta = \|A\|^2,\ \epsilon \in]0,1],\ y_0 \in \mathbb{R}^M.\right.$
For $n = 0, 1, \ldots$

$$\left\lfloor\begin{array}{l}\gamma_n \in [\epsilon\beta^{-1}, (2-\epsilon)\beta^{-1}]\\ \widetilde{y}_n = y_n - \gamma_n \nabla\left(\varphi\circ(-A^\top\cdot+\widetilde{x})\right)(y_n),\\ y_{n+1} = \text{prox}_{\gamma_n h^*}(\widetilde{y}_n)\end{array}\right.$$

Dual forward-backward algorithm

According to Moreau's decomposition formula, and the expression $\nabla\varphi = \mathrm{Id} - \mathrm{prox}_{f^*}$, the previous algorithm is equivalent to:

Initialization

$\beta = \|A\|^2$, $\epsilon \in]0,1]$, $y_0 \in \mathbb{R}^M$.

For $n = 0, 1, \ldots$

$$\begin{aligned}
&\gamma_n \in [\epsilon\beta^{-1}, (2-\epsilon)\beta^{-1}] \\
&x_n = \mathrm{prox}_f(\widetilde{x} - A^\top y_n) \\
&\widetilde{y}_n = y_n + \gamma_n A x_n, \\
&y_{n+1} = \widetilde{y}_n - \gamma_n \mathrm{prox}_{\gamma_n^{-1} h}(\gamma_n^{-1}\widetilde{y}_n).
\end{aligned}$$

$\Rightarrow$ Convergence of both $(x_n)_{n\in\mathbb{N}}$ and $(y_n)_{n\in\mathbb{N}}$ proved in [Combettes *et al.*, 2011].

Dual forward-backward algorithm

According to Moreau's decomposition formula, and the expression $\nabla\varphi = \mathrm{Id} - \mathrm{prox}_{f^*}$, the previous algorithm is equivalent to:

Initialization
$\lfloor\ \beta = \|A\|^2,\ \epsilon \in]0,1],\ y_0 \in \mathbb{R}^M.$
For $n = 0, 1, \ldots$

$$
\left\lfloor
\begin{array}{l}
\gamma_n \in [\epsilon\beta^{-1}, (2-\epsilon)\beta^{-1}] \\
x_n = \mathrm{prox}_f(\widetilde{x} - A^\top y_n) \\
\widetilde{y}_n = y_n + \gamma_n A x_n, \\
y_{n+1} = \widetilde{y}_n - \gamma_n \mathrm{prox}_{\gamma_n^{-1} h}(\gamma_n^{-1}\widetilde{y}_n).
\end{array}
\right.
$$

$\Rightarrow$ A special case is Dykstra-like algorithm [Bauschke *et al*, 2007], itself generalizing the Dykstra algorithm for computing the projection onto the intersection of convex sets.

Dual forward-backward algorithm

According to Moreau's decomposition formula, and the expression $\nabla\varphi = \mathrm{Id} - \mathrm{prox}_{f^*}$, the previous algorithm is equivalent to:

Initialization

$\lfloor\ \beta = \|A\|^2,\ \epsilon \in]0,1],\ y_0 \in \mathbb{R}^M.$

For $n = 0, 1, \dots$

$$
\left\lfloor
\begin{array}{l}
\gamma_n \in [\epsilon\beta^{-1}, (2-\epsilon)\beta^{-1}] \\
x_n = \mathrm{prox}_f(\widetilde{x} - A^\top y_n) \\
\widetilde{y}_n = y_n + \gamma_n A x_n, \\
y_{n+1} = \widetilde{y}_n - \gamma_n \mathrm{prox}_{\gamma_n^{-1} h}(\gamma_n^{-1}\widetilde{y}_n).
\end{array}
\right.
$$

$\Rightarrow$ The DFB algorithm is also known, in machine learning, as the dual ascent algorithm.

Proposed acceleration strategy

✗ In the context of large scale problems, accelerating the DFB algorithm is of main interest:

- ⇝ A variable metric strategy is introduced to improve the convergence rate (see [Repetti *et al.*, 2014]).
- ⇝ A block-coordinate strategy is adopted for better flexibility, and reduction of the computational cost per iteration ($\sim$ Gauss-Seidel methods).

Proposed acceleration strategy

✗ In the context of large scale problems, accelerating the DFB algorithm is of main interest:

⇝ A variable metric strategy is introduced to improve the convergence rate (see [Repetti *et al.*, 2014]).

⇝ A block-coordinate strategy is adopted for better flexibility, and reduction of the computational cost per iteration ($\sim$ Gauss-Seidel methods).

LINK WITH EXISTING WORKS:

- Dual coordinate ascent algorithms [Shalev-Shwartz *et al.*, 2013] , [Jaggi *et al.*, 2014] $\Rightarrow$ Stochastic selection of blocks.

Proposed acceleration strategy

✗ In the context of large scale problems, accelerating the DFB algorithm is of main interest:

⇝ A variable metric strategy is introduced to improve the convergence rate (see [Repetti *et al.*, 2014]).

⇝ A block-coordinate strategy is adopted for better flexibility, and reduction of the computational cost per iteration (∼ Gauss-Seidel methods).

LINK WITH EXISTING WORKS:

- Dual coordinate ascent algorithms
- Accelerated proximal alternating descent [Chambolle *et al.*, 2015]
  ⇒ FISTA-like acceleration.

Proposed acceleration strategy

✗ In the context of large scale problems, accelerating the DFB algorithm is of main interest:

$\rightsquigarrow$ A variable metric strategy is introduced to improve the convergence rate (see [Repetti *et al.*, 2014]).

$\rightsquigarrow$ A block-coordinate strategy is adopted for better flexibility, and reduction of the computational cost per iteration ($\sim$ Gauss-Seidel methods).

LINK WITH EXISTING WORKS:

- Dual coordinate ascent algorithms
- Accelerated proximal alternating descent
- Sparse Kaczmarz algorithm [Lorentz *et al.*, 2014] $\Rightarrow$ When f is the ℓ_1 norm, and h is the sum of indicator functions of singletons.

Primal problem

PRIMAL PROBLEM

Compute $\mathrm{prox}_g(\widetilde{x})$ with $(\forall x \in \mathbb{R}^N) \quad g(x) = f(x) + \sum_{j=1}^{J} h_j(A_j x)$,

where

- f belongs to $\Gamma_0(\mathbb{R}^N)$,
- $(\forall j \in \{1, \ldots, J\})$ h_j belongs to $\Gamma_0(\mathbb{R}^{M_j})$,
- $(\forall j \in \{1, \ldots, J\})$ $A_j \in \mathbb{R}^{M_j \times N}$.

Primal problem

Primal problem

$$\underset{x\in\mathbb{R}^N}{\text{minimize}} \quad f(x) + \sum_{j=1}^{J} h_j(A_j x) + \frac{1}{2}\|x - \widetilde{x}\|^2,$$

where

- f belongs to $\Gamma_0(\mathbb{R}^N)$,
- $(\forall j \in \{1,\dots,J\})$ h_j belongs to $\Gamma_0(\mathbb{R}^{M_j})$,
- $(\forall j \in \{1,\dots,J\})$ $A_j \in \mathbb{R}^{M_j \times N}$.

Qualification condition:

$$(\forall j \in \{1,\dots,J\}) \quad \mathrm{ri}\,(A_j(\mathrm{dom}\, f)) \cap \mathrm{ri}\,(\mathrm{dom}\, h_j) \neq \varnothing$$

Dual problem

DUAL PROBLEM

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

▶ The right part of the criterion is now separable with respect to the dual components.

⇒ We can apply the block-coordinate variable metric strategy from [Chouzenoux *et al.*, 2014] to solve the dual problem:

At each iteration $n \in \mathbb{N}$, a block index j_n is selected. The corresponding dual variable $y_n^{j_n}$ is updated, according to a preconditioned forward-backward rule, while all the other dual variables are kept unchanged.

Dual block preconditioned forward-backward algorithm

Algorithm DBFB:

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 $n = 0,1,\ldots$

$\gamma_n \in [\epsilon, 2-\epsilon]$

$j_n \in \{1,\ldots,J\}$

$x_n = \operatorname{prox}_f(\widetilde{x} + z_n)$

$\widetilde{y}_n^{j_n} = y_n^{j_n} + \gamma_n B_{j_n}^{-1} A_{j_n} x_n$

$y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n B_{j_n}^{-1} \operatorname{prox}_{\gamma_n B_{j_n}^{-1}, h_{j_n}} (\gamma_n^{-1} B_{j_n} \widetilde{y}_n^{j_n})$

$y_{n+1}^j = y_n^j, \quad \forall j \in \{1,\ldots,J\} \setminus \{j_n\}$

$z_{n+1} = z_n - A_{j_n}^\top (y_{n+1}^{j_n} - y_n^{j_n})$.

Proximity operator within a general metric

- Computation of $\operatorname{prox}_{C,g}(\widetilde{x})$ with $C \in \mathcal{S}^+(\mathbb{R}^N)$?

Algorithm DBFB:

Initialization

$B_j \in \mathbb{R}^{M_j \times M_j}$ with $B_j \succeq A_j C^{-1} A_j^\top, \quad \forall j \in \{1, \ldots, J\}$

$\epsilon \in]0,1]$, $(y_0^j)_{1 \leqslant j \leqslant J} \in \mathbb{R}^M$, $z_0 = -C^{-1} \sum_{j=1}^J A_j^\top y_0^j$.

For $n = 0, 1, \ldots$

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

Particular case when $f = 0$

- When $f = 0$, the algorithm simplifies as follows:

Algorithm DBFB0:

Initialization

$$B_j \in \mathbb{R}^{M_j \times M_j} \text{ with } B_j \succeq A_j A_j^\top, \quad \forall j \in \{1, \ldots, J\}$$
$$\epsilon \in]0,1],\ (y_0^j)_{1 \leqslant j \leqslant J} \in \mathbb{R}^M,\ x_0 = \widetilde{x} - \textstyle\sum_{j=1}^J A_j^\top y_0^j.$$

For $n = 0, 1, \ldots$

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

Convergence result

ASSUMPTIONS:

1. The functions f and $(h_j)_{1 \leqslant j \leqslant J}$ are semi-algebraic.
2. For every $j \in \{1, \ldots, J\}$, the restriction of h_j^* to its domain is continuous.
3. For every $j \in \{1, \ldots, J\}$, matrix B_j is definite positive.
4. 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 $n \in \mathbb{N}$, $\{1, \ldots, J\} \subset \{j_n, \ldots, j_{n+K-1}\}$.

If the sequence $(y_n)_{n \in \mathbb{N}} = \big((y_n^j)_{1 \leqslant j \leqslant J}\big)_{n \in \mathbb{N}}$ is bounded, then this sequence converges to a solution to the dual problem. In addition, the sequence $(x_n)_{n \in \mathbb{N}}$ converges to the proximity operator of g evaluated at $\widetilde{x}$.

Convergence rate result

Suppose that the previous assumptions hold and that $\widehat{x}$ and $\widehat{y}$ are the limits of $(x_n)_{n\in\mathbb{N}}$ and $(y_n)_{n\geqslant 1}$, respectively.
There exists $\alpha \in]0,+\infty[$ and $\lambda \in]0,+\infty[$ such that, for every $n \geqslant 1$,

$$\|x_n - \widehat{x}\| \leqslant \lambda \|A\| n^{-\alpha}, \quad \|y_n - \widehat{y}\| \leqslant \lambda n^{-\alpha}.$$

In addition, if one of the following conditions is met:

1. the dual cost function is strongly convex,
2. f is Lipschitz differentiable and A is surjective,
3. for every $j \in \{1,\ldots,J\}$, h_j is Lipschitz differentiable,
4. the dual cost function is a piecewise polynomial function of degree 2,
5. f is a quadratic function and, for every $j \in \{1,\ldots,J\}$, h_j^* is a piecewise polynomial function of degree 2,

then, there exists $\tau \in [0,1[$ and $\lambda' \in]0,+\infty[$ such that, for every $n \geqslant 1$,

$$\|x_n - \widehat{x}\| \leqslant \lambda' \|A\| \tau^n, \quad \|y_n - \widehat{y}\| \leqslant \lambda' \tau^n.$$

Parallel dual block forward-backward algorithm

- Link with the method from [Combettes *et al.*,2011]:

Algorithm PDBFB:

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 $j = 1, \dots, J$

$\widetilde{y}_n^j = y_n^j + \gamma_n B_j^{-1} A_j x_n$

$y_{n+1}^j = \widetilde{y}_n^j - \gamma_n B_j^{-1} \mathrm{prox}_{\gamma_n B_j^{-1}, h_j}\left(\gamma_n^{-1} B_j \widetilde{y}_n^j\right)$

$x_{n+1} = x_n - \sum_{j=1}^J A_j^\top (y_{n+1}^j - y_n^j)$.

Experimental results

Simulation framework

Test video sequences:

⇝ Synthetic sequences **Foreman** and **Claire** ($N = 352 \times 288$, resp. $N = 360 \times 288$, $T = 50$), corrupted with a blur and a white Gaussian noise.

⇝ Real blurry interlaced video sequences **Tachan** and **Au théâtre ce soir** provided by INA ($L = 720 \times 288$, $T = 80$).

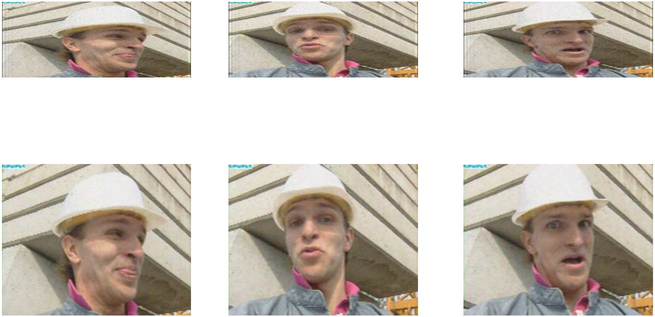
Restoration strategy:

⋆ Minimization using PALM algorithm.

⋆ $(\forall t \in \{1, \dots, T\})\ \ \Psi_t = \eta\ \mathrm{sltv} + \iota_{[x_{\min}, x_{\max}]^N}$ with $\eta > 0$, sltv the semi-local total variation [Condat, 2014], and M is a nonsmooth temporal regularization term [Abboud *et al*, 2014].

Among Algorithms BDFB, BDFB0 and PBDFB, which one is the most efficient for the computation of the proximal inner loops?

Visual results



Frames extracted from the noisy blurred interlaced field (top) and restored progressive image (bottom), of the Foreman sequence. Input SNR = 25.54*dB, output SNR =* 28.95*dB.*

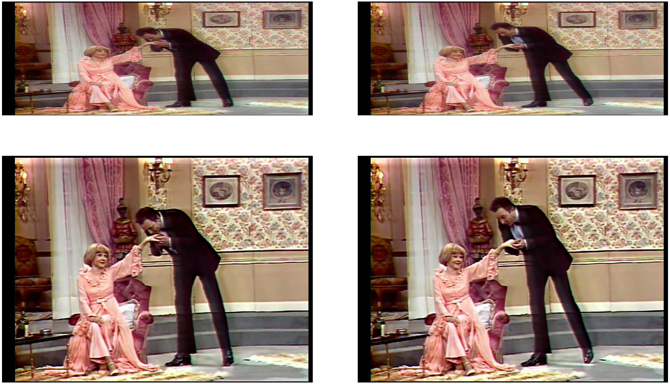
Visual results

Frames extracted from the noisy blurred interlaced field (top) and restored progressive image (bottom), of the Claire sequence. Input SNR = 25.27*dB, output SNR =* 29.21*dB.*

Visual results

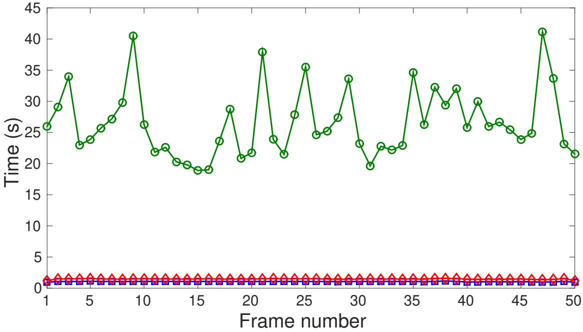
Frames extracted from the noisy blurred interlaced field (top) and restored progressive image (bottom), of ***Tachan*** *sequence.*

Visual results



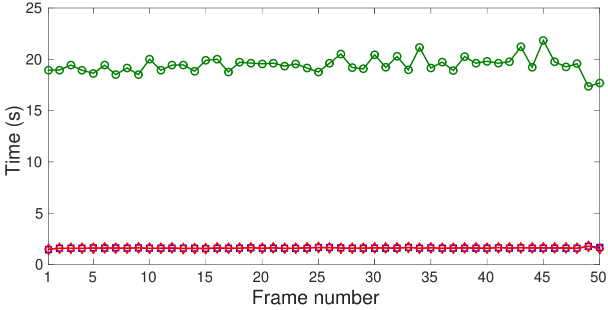
Frames extracted from the noisy blurred interlaced field (top) and restored progressive image (bottom), of ***Au théâtre ce soir*** *sequence.*

Speed



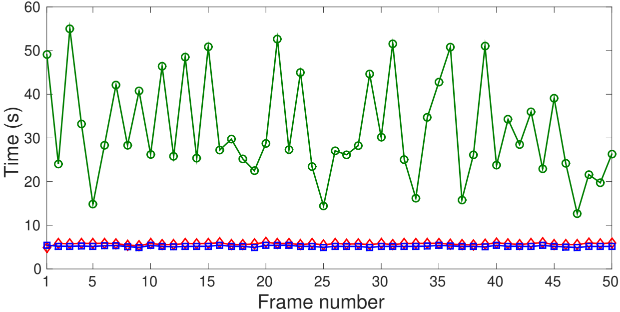
Averaged execution time (in s.) per **Foreman** frame using BDFB □, BDFB0 ◇ or PBDFB ○

Speed



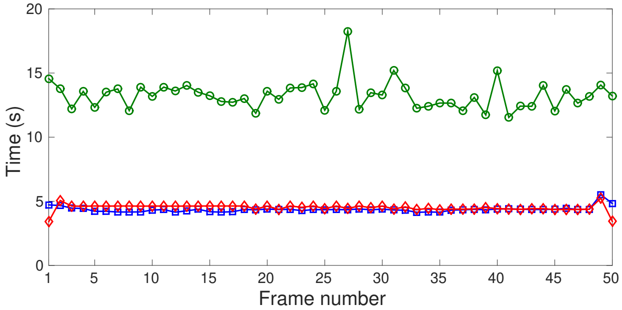
Averaged execution time (in s.) per **Claire** frame using BDFB □, BDFB0 ◇ or PBDFB ○

Speed



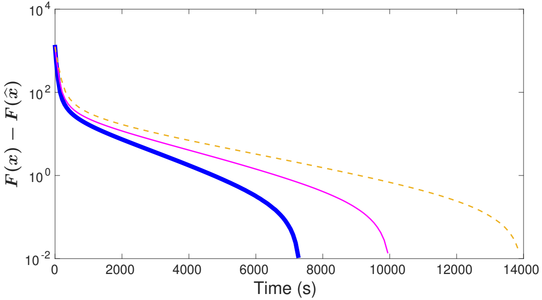
Averaged execution time (in s.) per **Tachan** frame using BDFB □, BDFB0 ◇ or PBDFB ○

Speed



Averaged execution time (in s.) per **Au théâtre ce soir** frame using BDFB □, BDFB0 ◇ or PBDFB ○

Benefit of preconditioning



Average execution time per **Foreman** frame for proximity step in BDFB algorithm using preconditioning, no preconditioning using exact norms and no preconditioning using approximate norms.

Distributed strategy

Distributed formulation

ORIGINAL PROBLEM

Find $\widehat{x} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{j=1}^{J} h_j(A_j x) + \frac{1}{2}\|x - \widetilde{x}\|^2.$

REFORMULATED PROBLEM

Find $\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}=(x^j)_{1\leqslant j\leqslant J} \in \Lambda_J}{\operatorname{argmin}} \sum_{j=1}^{J} h_j(A_j x^j) + \frac{1}{2}\sum_{j=1}^{J} \omega_j \|x^j - \widetilde{x}\|^2.$

Standard coupling constraint:

$$\Lambda_J = \left\{ \begin{bmatrix} x^1 \\ \vdots \\ x^J \end{bmatrix} \in \mathbb{R}^{NJ} \mid x^1 = \ldots = x^J \right\}.$$

Local form of consensus

- We define $(\mathbb{V}_\ell)_{1\leqslant\ell\leqslant L}$of $\{1,\ldots,J\}$ with cardinality $(\kappa_\ell)_{1\leqslant\ell\leqslant L}$ such that

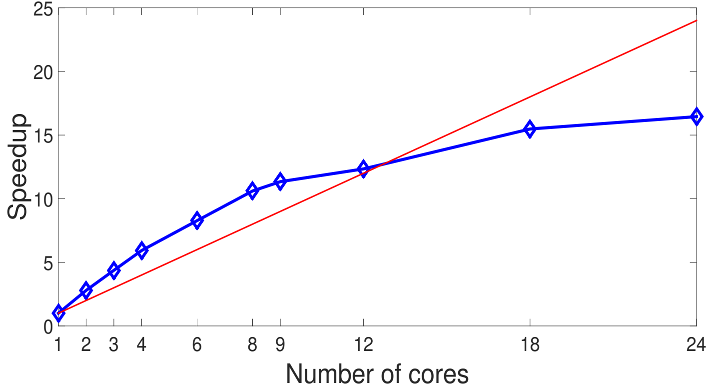
$$\boldsymbol{x}\in\Lambda \quad\Leftrightarrow\quad (\forall\ell\in\{1,\ldots,L\}) \quad (x^j)_{j\in\mathbb{V}_\ell}\in\Lambda_{\kappa_\ell}.$$

- $\mathbb{V}_\ell$ can be viewed as the ℓ-th hyperedge of a connected hypergraph with nodes $\{1,\ldots,J\}$.
- Resolution by application of BDFB to the problem:

$$\underset{\boldsymbol{x}=(x^j)_{1\leqslant j\leqslant J}\in\mathbb{R}^{NJ}}{\text{minimize}} \quad \sum_{j=1}^{J} h_j(A_j x^j)+\sum_{\ell=1}^{L}\iota_{\Lambda_{\kappa_\ell}}(\boldsymbol{S}_\ell\,\boldsymbol{x})+\frac{1}{2}\sum_{j=1}^{J}\omega_j\|x^j-\widetilde{x}\|^2.$$

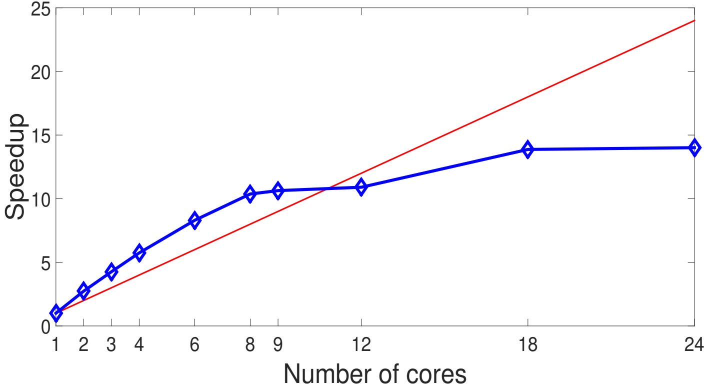
where, for every $\ell\in\{1,\ldots,\ell\}$ $\boldsymbol{S}_\ell\in\mathbb{R}^{N\kappa_\ell\times NJ}$ is some decimation operator.

Experimental results (Julia programming language)



Speedup w.r.t. the number of used cores for **Foreman** sequence:proposed method versus linear speedup.

Experimental results (Julia programming language)



Speedup w.r.t. the number of used cores for **Claire** sequence:'proposed method versus linear speedup.

Conclusion

Deterministic primal-dual splitting algorithm to handle efficiently the computation of the proximity operator of composite convex functions.

- ✓ Convergence guaranteed for both its primal and dual iterates,
- ✓ High flexibility in the selection of blocks,
- ✓ Possibility to include sophisticated preconditioning strategy,
- ✓ Good performance in the context of video restoration,
- ✓ Extension to distributed optimization.

Some of our references ...

F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot and L. Laborelli
Dual Block Coordinate Forward-Backward Algorithm with Application to Deconvolution and Deinterlacing of Video Sequences
Journal of Mathematical Imaging and Vision, (to appear), 2017.

F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot and L. Laborelli
A Distributed Strategy for Computing Proximity Operators
49th Asilomar Conference on Signals, Systems and Computers, Asilomar, California, United States, 8-11 Nov. 2015.

F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot and L. Laborelli
A Hybrid Alternating Proximal Method for Blind Video Restoration
European Signal Processing Conference (EUSIPCO 2014), pp. 1811-1815, Lisboa, Portugal, 1-5 Sep. 2014.

E. Chouzenoux, J.-C. Pesquet and A. Repetti
A Block Coordinate Variable Metric Forward-Backward Algorithm
Journal of Global Optimization, Vol. 66, No. 3, pages 457–485, 2016.

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 editors. Springer-Verlag, New York, pp. 185-212, 2011.

N. Komodakis and J.-C. Pesquet
Playing with Duality: An Overview of Recent Primal-Dual Approaches for Solving Large-Scale Optimization Problems
IEEE Signal Processing Magazine, vol. 32, no. 6, pp. 31-54, 2015.