Restoration super-resolution of image sequences : application to TV archive documents

Feriel Abboud

Université Paris-Est

Lab. Informatique Gaspard Monge

UMR CNRS 8049

THESIS

presented by

Feriel Abboud

Restoration and super-resolution of image sequences
Application to TV archive documents

Reviewers	Valeria Ruggiero	*Universitá di Ferrara*
	Yves Wiaux	*Heriot-Watt University*
Examiners	Ewa Bednarczuk	*Warsaw University of Technology*
	Nikos Paragios	*CentraleSupelec - Université Paris-Saclay*
	Laurent Najman	*Université Paris-Est, ESIEE*
PhD supervisor	Jean-Christophe Pesquet	*CentraleSupelec - Université Paris-Saclay*
PhD co-supervisors	Émilie Chouzenoux	*Université Paris-Est Marne-la-Vallée*
	Jean-Hugues Chenot	*Institut National de l'Audiovisuel*

Résumé

Au cours du dernier siècle, le volume de vidéos stockées chez des organismes tel que l'Institut National de l'Audiovisuel a connu un grand accroissement. Ces organismes ont pour mission de préserver et de promouvoir ces contenus, car, au-delà de leur importance culturelle, ces derniers ont une vraie valeur commerciale grâce à leur exploitation par diverses médias. Cependant, la qualité visuelle des vidéos est souvent moindre comparée à celles acquises par les récents modèles de caméras. Ainsi, le but de cette thèse est de développer de nouvelles méthodes de restauration de séquences vidéo provenant des archives de télévision française, grâce à de récentes techniques d'optimisation.

La plupart des problèmes de restauration peuvent être résolus en les formulant comme des problèmes d'optimisation, qui font intervenir plusieurs fonctions convexes mais non-nécessairement différentiables. Pour ce type de problèmes, on a souvent recourt à un outil efficace appelé opérateur proximal. Le calcul de l'opérateur proximal d'une fonction se fait de façon explicite quand cette dernière est simple. Par contre, quand elle est plus complexe ou fait intervenir des opérateurs linéaires, le calcul de l'opérateur proximal devient plus compliqué et se fait généralement à l'aide d'algorithmes itératifs.
Une première contribution de cette thèse consiste à calculer l'opérateur proximal d'une somme de plusieurs fonctions convexes composées avec des opérateurs linéaires. Nous proposons un nouvel algorithme d'optimisation de type primal-dual, que nous avons nommé *Algorithme Explicite-Implicite Dual par Blocs*. Notre algorithme se base sur un schéma explicite-implicite préconditionné pour minimiser le problème dual, en alternant à chaque itération, entre une étape de gradient sur la partie différentiable du problème dual et une étape proximale sur le partie restante. L'algorithme proposé permet de ne mettre à jour, à chaque itération, qu'un sous-ensemble de blocs choisi selon une règle déterministe acyclique. Notons aussi que notre algorithme permet de traiter les opérateurs linéaires sans passer par leur inversion ni par des sous-itérations, ce qui réduit grandement la complexité d'implémentation et de calcul. Des résultats de convergence ont été établis pour les deux suites primales et duales générées par notre algorithme.

Nous avons appliqué notre algorithme au problème de déconvolution et désentrelacement de séquences vidéo. Ce dernier consiste à estimer une séquence vidéo progressive à partir d'une séquence entrelacée, où chaque image entrelacée résulte de la combinaison des lignes impaires d'une image progressive et des lignes paires de l'image progressive suivante. Ainsi, notre tâche consiste à estimer les lignes paires/impaires supprimées, tout en réalisant une opération de déconvolution sur les images estimées. Pour cela, nous avons modélisé notre problème sous la forme d'un problème d'optimisation dont la solution est obtenue à l'aide de l'algorithme explicite-implicite dual par blocs. Nous avons comparé l'algorithme proposé à des algorithmes d'optimisation de la littérature et les résultats obtenus montrent l'efficacité de notre méthode en terme de qualité de restauration et de vitesse de convergence. Ceci est essentiellement dû au traitement par blocs et à l'utilisation de matrices de préconditionnement.

Dans la deuxième partie de cette thèse, nous nous sommes intéressés au développement d'une version asynchrone de notre l'algorithme explicite-implicite dual par blocs, pour le calcul de l'opérateur proximal d'une somme de fonctions convexes composées avec des opérateurs linéaires. Dans cette nouvelle extension, chaque fonction composant le critère est considérée comme locale et rattachée à une unité de calcul. Ces unités de calcul traitent les fonctions et les données qui lui sont associées de façon indépendante les unes des autres. Afin d'obtenir une solution de consensus, il est nécessaire d'établir une stratégie de communication efficace, qui est définie dans notre cas par un hypergraphe connecté. Un point crucial dans le développement d'un tel algorithme est le choix de la fréquence et du volume de données à échanger entre les unités de calcul, dans le but de préserver de bonnes performances d'accélération.
Nous avons évalué numériquement notre algorithme distribué sur un problème de débruitage de séquences vidéo. Nous avons implémenté l'algorithme sur une architecture multi-coeurs à l'aide de l'outil "MPI" *(Message-Passing Interface)* pour la gestion des communications entre processeurs. Les images composant la vidéo sont partitionnées de façon équitable sur les processeurs, puis, chaque processeur exécute une instance de l'algorithme de façon asynchrone et communique avec les processeurs voisins uniquement. Les bons résultats obtenus en terme de temps d'exécution et en fonction du nombre de processeurs utilisé soulignent l'efficacité de notre méthode.

Finalement, nous nous sommes intéressés au problème de déconvolution aveugle de séquences vidéo, qui vise à estimer le noyau de convolution et la séquence originale à partir de la séquence dégradée observée. Ce type de problème inverse est sévèrement *mal posé*, par conséquent, il est nécessaire de recourir à une approche régularisée afin de le résoudre.
Nous avons proposé dans cette dernière partie une nouvelle méthode de déconvolution

aveugle basée sur la formulation d'un problème non-convexe, résolu par un algorithme itératif qui, à chaque itération, alterne entre l'estimation de la séquence originale et l'identification du noyau de convolution. Notre méthode a la particularité de pouvoir intégrer diverses types de fonctions de régularisations avec des propriétés mathématiques différentes. Nous avons réalisé des tests et simulations sur des séquences synthétiques et réelles, avec différents noyaux de convolution. Nous avons décomposé la tâche de restauration en deux étapes : (i) une étape de déconvolution aveugle permettant d'identifier le noyau de convolution à partir de la séquence dégradée, (ii) une étape de déconvolution non-aveugle visant à estimer la séquence originale à partir de la séquence dégradée et du noyau identifié. La flexibilité de notre approche nous a permis de réaliser des comparaisons entre plusieurs fonctions de régularisation spatiale, convexes et non-convexes, en terme de qualité d'estimation au sein de la première et de la deuxième étape du processus.

Contents

- Chapter 1 -

Introduction

Context

The last century has witnessed an explosion in the amount of video data stored with holders such as the National Audiovisual Institute whose mission is to preserve and promote the content of French broadcast programs. Beyond the cultural impact of these records, their value is increased due to commercial reexploitation through recent visual media. However, the perceived quality of the old data fails to satisfy the current public demand. The purpose of this thesis is to propose new methods for restoring video sequences supplied from television archive documents, using modern optimization techniques with proven convergence properties.

In a large number of restoration issues, the underlying optimization problem is made up with several functions which might be convex and non-necessarily smooth. In such instance, the proximity operator, a fundamental concept in convex analysis, appears as the most appropriate tool. These functions may also involve arbitrary linear operators that need to be inverted in a number of optimization algorithms. In this spirit, we developed a new primal-dual algorithm for computing non-explicit proximity operators based on forward-backward iterations. The proposed algorithm is accelerated thanks to the introduction of a preconditioning strategy and a block-coordinate approach in which at each iteration, only a "block" of data is selected and processed according to a quasi-cyclic rule. This approach is well suited to large-scale problems since it reduces the memory requirements and accelerates the convergence speed, as illustrated by some experiments in deconvolution and deinterlacing of video sequences.
Afterwards, a close attention is paid to the study of distributed algorithms on both theoretical and practical viewpoints. We proposed an asynchronous extension of the

dual forward-backward algorithm, that can be efficiently implemented on a multi-cores architecture. In our distributed scheme, the primal and dual variables are considered as private and spread over multiple computing units, that operate independently one from another. Nevertheless, communication between these units following a predefined strategy is required in order to ensure the convergence toward a consensus solution.

We also address in this thesis the problem of blind video deconvolution that consists in inferring from an input degraded video sequence, both the blur filter and a sharp video sequence. Hence, a solution can be reached by resorting to nonconvex optimization methods that estimate alternatively the unknown video and the unknown kernel. In this context, we proposed a new blind deconvolution method that allows us to implement numerous convex and nonconvex regularization strategies, which are widely employed in signal and image processing.

Collaborations

This thesis is the result of a collaboration with the National Audiovisual Institute under the supervision of Jean-Hugues CHENOT and in collaboration with Louis LABORELLI. The thesis was financially supported by the National Research and Technology Association (ANRT), within a CIFRE convention. This collaboration provided new insights and a deep understanding on real-word problems arising in video processing, and more precisely for the restoration of old films and television shows, which is an original and less investigated problem in the literature of video processing.

Organization of the document

Chapter 2 provides an introduction to inverse problems in the context of image/video processing, where one seeks to recover the closest version of an original sharp image (or a sequence of images) from the observed one. We are interested in this thesis in two main image and video restoration issues, namely deconvolution (both its blind and non-blind schemes) and super-resolution. In this context, we present in that chapter an overview of state-of-the-art image/video deconvolution and super-resolution approaches. Then, we pay attention to degradations and artifacts observed on videos supplied from old television archives.
Inverse problems are usually solved thanks to formulations as optimization problems where one minimizes a criterion expressed as a sum of functions, representing a description of the degradation process and some *a priori* knowledge on the expected sharp image/video. However, the solution to this problem is usually not

explicit and iterative methods are hence needed. We present in Chapter 3 a review on standard iterative algorithms for addressing different types of minimization problems depending on the mathematical properties of the involved functions. More precisely, we cover gradient methods, proximal algorithms, Majorize-Minimize approaches, primal-dual techniques and dual forward-backward algorithms.

Chapter 3 shows that a solution to numerous optimization problems involves the computation of proximity operators. Depending on the underlying objective function, a closed form expression of the latter may not be available, so that inner iterations are necessary. We focus in Chapter 4 on the evaluation of the proximity operator of a sum of convex possibly nonsmooth functions composed with arbitrary matrices. We design a new dual forward-backward algorithm that relies on a block-coordinate approach. At each iteration, only a subset of the processed data is selected according to a quasi-cyclic rule, which allows us to reduce the complexity of the algorithm and its memory requirements. The convergence is also accelerated thanks to the introduction of preconditioning matrices that are designed using Majorize-Minimize strategies. One advantage of the proposed method is that it does not require the computation of the norm of the involved linear operators, which in our applications is intractable.
The performance of our algorithm are assessed in terms of acceleration and restoration quality on synthetic and real sequences in Chapter 5. We focus on the problem of deconvolution and deinterlacing of video sequences, namely recovering the missing even/odd rows of each frame of the observed interlaced video, while ensuring good restoration quality. The experimental results illustrate that the dual bock-coordinate forward-backward algorithm achieves satisfactory acceleration rates compared to existing algorithms in the literature.

Then, we consider in Chapter 6 a distributed version of the dual block-coordinate forward-backward algorithm form Chapter 4. In the asynchronous framework, the blocks are partitioned over several computing units that manage their associated primal and dual variables locally. The computing units can process data independently one from each others, and communication between them are required and realized according to a predefined strategy, in order to ensure the convergence towards an aggregate solution. The amount and frequency of transfers between the computing units is adjusted so that the acceleration gain obtained through parallel architectures is not compromised. We evaluate our distributed algorithm on the problem of video denoising, where we provide an implementation on a multi-core machine using the *MPI* library. The numerical experiments evaluate the execution time with respect to the number of used processors. Our results show that the proposed algorithm runs efficiently on a distributed architecture, and speedup rates up to a factor 16 have been achieved.

In the aforementioned applications, the deconvolution process was non-blind, which means that the blur filter was assumed to be known. However, this assumption may not be consistent for real-world image and video restoration problems. Thus, we consider in Chapter 7 the blind video deconvolution problem, that consists in estimating both the convolution filter and the original video sequence from the observed degraded sequence. There exists a large number of regularization-based methods for image deconvolution problems which rely on several regularization functions with different mathematical properties (e.g., total variation, total generalized variation, ℓ_1/ℓ_2 norm ....). Few studies have been performed to compare them in an exhaustive manner. In this chapter, we propose a unified framework for blind video deconvolution that allows to take into account these various regularization strategies. We provide some experiments on synthetic blurred video sequences, comparing the regularization approaches, in the context of blind deconvolution first, and once the convolution kernels are identified, we perform a non-blind deconvolution stage on the same degraded sequences. Finally, in Chapter 8 we draw some conclusions and present some perspectives and future works.

Publications

Journal papers :

1) Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Jean-Hugues Chenot, Louis Laborelli. **Dual block-coordinate forward-backward algorithm with application to deconvolution and deinterlacing of video sequences**. To appear in Journal of Mathematical Imaging and Vision, 2016.

2) Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Hugues Talbot. **Distributed proximity operator computation with applications to video deconvolution and super-resolution**. To be submitted, 2017.

3) Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Jean-Hugues Chenot, Louis Laborelli. **An alternating proximal method for blind video deconvolution**. To be submitted, 2017.

Conference papers :

3) Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Jean-Hugues Chenot, Louis Laborelli. **A distributed strategy for computing proximity operators**. 49th Asilomar Conference on Signals, Systems and Computers, pp. 396-400, Pacific Grove, California, USA, 8-11 Nov. 2015.

4) Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Jean-Hugues Chenot, Louis Laborelli. **A dual block-coordinate proximal algorithm with application to deconvolution of interlaced video sequences**. In International Conference on Image Processing (ICIP 2015), pp. 4917- 4921, Quebec City, Canada, 27-30 Sep. 2015.

5) Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Jean-Hugues Chenot, Louis Laborelli. **A hybrid alternating proximal method for blind video restoration**. In European Signal Processing Conference (EUSIPCO 2014), pp. 1811-1815, Lisboa, Portugal, Sep. 2014.

Workshops/Seminars:

6) Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Jean-Hugues Chenot, Louis Laborelli. **Accelerated dual forward-backward algorithms. Application to video restoration**. Optimization techniques for inverse problems, Modena, Italy, 19-21 Sep. 2016.

7) Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Jean-Hugues Chenot, Louis Laborelli. **Alternating proximal strategy for blind video restoration involving various regularization strategies**. Applied Inverse Problems Conference, Helsinki, Finland, 25-29 May 2015.

- Chapter 2 -

Background on inverse problems for image and video processing

Contents

2.1 Introduction

Nowadays, images and videos play a prominent part in our personal and professional life, therefore the need for a higher image quality becomes increasingly important. In this thesis We are mainly interested in two fundamental topics in image and video processing, namely deconvolution and super-resolution. First, we begin by providing in Section 2.2 an overview on inverse problems in the context of image processing. Then, we introduce in Sections 2.3 and 2.4 the image deconvolution and super-resolution problems for still images, and present some standard and recent approaches to address them. Afterwards, we focus on video restoration issues in Section 2.5, especially in the case of videos from television archives.

2.2 Inverse problems

2.2.1 Introduction

A wide number of optimization problems aim at recovering an image from corrupted measurements that cannot be processed in a direct way. Usually, the target image is related to the observations through a system which describes the degradation process, covering lens imperfections, quantization or down-sampling operations. The recovery of the true image is known as an *inverse problem* [Demoment, 1989; Bertero and Boccacci, 1998]. This class of problems is encountered in image and video restoration problems such as denoising, deconvolution, inpainting, tomography, arising in various application fields [Jansson, 1997; Hammond et al., 2011; Pock et al., 2010].
Solving an inverse problem requires the design of a *direct model* that links the target image to the observations. Within the context of image restoration, the generic form of a direct model is usually as follows

$$y = H\overline{x} + w, \tag{2.1}$$

where $y \in \mathbb{R}^M$ denotes the observed image, $\overline{x} \in \mathbb{R}^N$ represents the original sought image, and $H \in \mathbb{R}^{M \times N}$ is an operator describing the degradation system. In this thesis, we focus on linear degradation operators, such as convolution and decimation operators. The observed image y is usually affected by perturbations arising during the acquisition and transmission stages. These perturbations are cumulated and represented in the model (2.1) as an additive random noise $w \in \mathbb{R}^M$.
The inverse problem associated to the model (2.1) consists in finding an estimate $\widehat{x} \in \mathbb{R}^N$ of the original image $\overline{x}$ from the measured image y knowing the degradation operator H. Nevertheless, in some applications, the linear operator H is also unknown. In such cases, the *blind inverse problem* amounts to inferring both the original image and the linear operator from the observations.

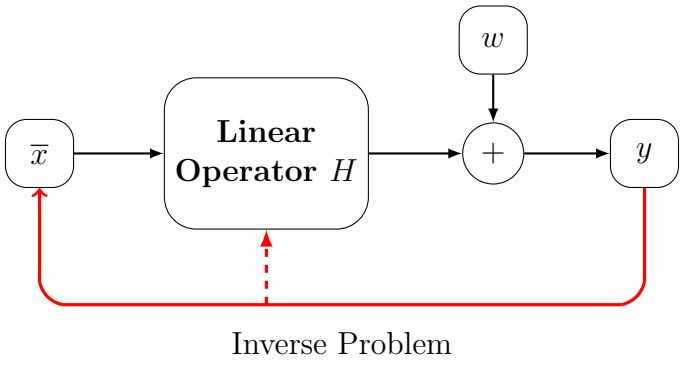


Figure 2.1: *Diagram representing a linear inverse problem.*

Inverse problems are usually *ill-posed* in the sense of Hadamard's definition [Hadamard, 1902], which states the following conditions that have to be fulfilled by a *well-posed* problem:

1. ***Existence***: there exists a solution to the problem.
2. ***Uniqueness***: the solution to the problem is unique.
3. ***Stability***: the solution depends continuously on the observations, namely when the perturbation on the observed image y tends towards zero, the error on the estimated image $\widehat{x}$ tends as well towards zero.

In the case of an inverse problem involving a degradation operator H, the *well-posedness* of the problem is ensured if $\mathrm{Im}(H) = \mathbb{R}^M$ (*existence*), which means that each observation $y \in \mathbb{R}^M$ is an image of $\overline{x} \in \mathbb{R}^N$ by the linear operator H. The second condition (*uniqueness*) is guaranteed if $\mathrm{Ker}(H) = \{0\}$ which refers to the fact that the solution to $Hx = 0$ is the singleton $\{0\}$. Finally the inverse mapping has to be continuous (*stability*) which is satisfied in finite dimension when H is invertible.

Let us consider the case when the linear mapping H is an invertible square matrix in $\mathbb{R}^{N\times N}$. Then a direct solution can be computed as follows:

$$\widehat{x} = H^{-1}y. \tag{2.2}$$

However, for most degradation operators in image restoration tasks, H is *ill-conditioned*, which means that the condition number of H, given by the ratio between its maximum and minimum singular values $\sigma_{\max}/\sigma_{\min}$, is high. Hence, the obtained image $\widehat{x} = \overline{x} + H^{-1}w$ becomes highly sensitive to the noise component w due to the prominence of the term $H^{-1}w$, leading to an unsatisfactory restoration quality. The same problem occurs in the case of a non-invertible linear operator, when the *pseudo-inverse* matrix $H^\top(H\,H^\top)^{-1}$ is used instead.

Figure 2.2: *Deconvolution problem: original image $\overline{x}$ (left), degraded image y with a motion blur and a zero mean additive Gaussian noise (right). The inverse problem consists in finding an estimate of the image $\overline{x}$ from the observed image y. In the case of a blind deconvolution problem, the blur operator is unknown and has to be estimated as well.*

An efficient approach to find a consistent estimate $\widehat{x}$ of $\overline{x}$ is to define it as a solution to a minimization problem formulated as follows

$$\text{Find } \widehat{x} \in \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \ \Phi(x) + \Psi(x), \tag{2.3}$$

where Φ denotes a *data fidelity term* that measures the residual error between the observed image y and Hx, and Ψ is a *regularization term* [Demoment, 1989] which is introduced in order to provide a stable solution to the *ill-posed* problem by incorporating *a priori* knowledge on the sought image.
In the blind context, the inverse problem is *ill-posed* with respect to the image and to the linear operator. Thus, prior information on the operator H is needed in order to get a stable identification. Estimates $\widehat{x}$ and $\widehat{H}$ can be obtained by solving the following problem

$$\text{Find } \left(\widehat{x}, \widehat{H}\right) \in \underset{x \in \mathbb{R}^N, H \in \mathbb{R}^{M \times N}}{\operatorname{argmin}} \ \Phi(x, H) + \Psi(x) + \Theta(H), \tag{2.4}$$

where Θ represents a regularization function on the linear mapping H.

2.2.2 Variational formulation

Bayesian interpretation

Inverse problems are often (almost always) *ill-posed*, and a formulation as a minimization problem is a suitable way to solve them. A Bayesian approach for choosing

an adequate data fidelity and regularization terms can be adopted to construct this minimization problem. In the Bayesian framework, all the observed and unknown quantities are viewed as stochastic variables with probability densities based on prior convictions. Let us assume that y and $\overline{x}$ are realizations of random variables $\mathbf{Y}$ and $\mathbf{X}$, respectively. Then finding an estimate $\widehat{x}$ using a *Maximum A Posteriori (MAP)* approach reads

$$\text{Find } \widehat{x} \in \underset{x\in\mathbb{R}^N}{\text{argmax}} \; \mathcal{P}_{\mathbf{X}|\mathbf{Y}=y}(x), \tag{2.5}$$

where $\mathcal{P}_{\mathbf{X}|\mathbf{Y}=y}$ is the posterior probability distribution. This probability density can be decomposed into a likelihood function and prior probability density thanks to the Bayes rule

$$\begin{aligned} &\text{Find } \widehat{x} \in \underset{x\in\mathbb{R}^N}{\text{argmax}} \; \mathcal{P}_{\mathbf{Y}|\mathbf{X}=x}(y)\, \frac{\mathcal{P}_{\mathbf{X}}(x)}{\mathcal{P}_{\mathbf{Y}}(y)}, \\ \Leftrightarrow\ &\text{Find } \widehat{x} \in \underset{x\in\mathbb{R}^N}{\text{argmax}} \; \mathcal{P}_{\mathbf{Y}|\mathbf{X}=x}(y)\, \mathcal{P}_{\mathbf{X}}(x), \end{aligned} \tag{2.6}$$

where $\mathcal{P}_{\mathbf{X}}$ and $\mathcal{P}_{\mathbf{Y}}$ are the probability densities of $\mathbf{X}$ and $\mathbf{Y}$ respectively.
By using the monotonicity property of the logarithm function, (2.6) can be expressed as

$$\text{Find } \widehat{x} \in \underset{x\in\mathbb{R}^N}{\text{argmin}} \; \left(-\log \mathcal{P}_{\mathbf{Y}|\mathbf{X}=x}(y) - \log \mathcal{P}_{\mathbf{X}}(x)\right). \tag{2.7}$$

Hence, by setting

$$(\forall x \in \mathbb{R}^N) \quad \begin{cases} \Phi(x) = -\log \mathcal{P}_{\mathbf{Y}|\mathbf{X}=x}(y) \\ \\ \Psi(x) = -\log \mathcal{P}_{\mathbf{X}}(x), \end{cases} \tag{2.8}$$

one can retrieve the minimization problem (2.3).
By defining the probability distribution related to the noise component $\mathcal{P}_{\mathbf{W}}$, we have

$$\left(\forall\, (x,y) \in \mathbb{R}^N \times \mathbb{R}^M\right) \quad \mathcal{P}_{\mathbf{Y}|\mathbf{X}=x}(y) = \mathcal{P}_{\mathbf{W}|\mathbf{X}=x}(w) = \mathcal{P}_{\mathbf{W}}(w), \tag{2.9}$$

with $w = y - Hx$, and assuming that the random variables $\mathbf{X}$ and $\mathbf{W}$ are independent.
Thus, in a Bayesian framework, the choice of the data fidelity term Φ depends on the noise type, whereas the regularization term Ψ is related to prior knowledge on the sought image.

Data fidelity term

In this thesis, we will consider data corrupted by a zero mean Gaussian noise with independent and identically distributed components, and a covariance matrix $\Gamma \in \mathbb{R}^{M\times M}$. Thus the probability density $\mathcal{P}_{\mathbf{W}}(w)$ is given by a normal distribution:

$$\left(\forall w \in \mathbb{R}^M\right) \quad \mathcal{P}_{\mathbf{W}}(w) = \frac{1}{\sqrt{(2\pi)^M \det(\Gamma)}} \; \exp\left(-\frac{1}{2}\, w^\top \Gamma^{-1} w\right), \tag{2.10}$$

where "det(Γ)" denotes the determinant of the matrix Γ.
By introducing the minus-logarithm function, we obtain:

$$\left(\forall w \in \mathbb{R}^M\right) \quad -\log\left(\mathcal{P}_{\mathbf{W}}(w)\right) = \frac{1}{2}\, w^\top \Gamma^{-1} w \,+\, \frac{1}{2}\log\left((2\pi)^M \det(\Gamma)\right). \tag{2.11}$$

Since $\frac{1}{2}\log\left((2\pi)^M\det(\Gamma)\right) \in \mathbb{R}$ is a constant that does not depend on $w = y - Hx$, the function Φ can be defined as

$$\left(\forall x \in \mathbb{R}^N\right) \quad \Phi(x) = -\log\left(\mathcal{P}_{\mathbf{W}}(y - Hx)\right) = \frac{1}{2}\,(y - Hx)^\top\,\Gamma^{-1}\,(y - Hx). \tag{2.12}$$

When $\Gamma = \sigma^2\,\mathrm{Id}_M$ with Id_M the identity matrix of $\mathbb{R}^{M\times M}$, Φ reduces to

$$\left(\forall x \in \mathbb{R}^N\right) \quad \Phi(x) = \frac{1}{2\sigma^2}\|y - Hx\|^2. \tag{2.13}$$

Regularization term

Thanks to regularization, a stable and computable solution to inverse problems can be supplied. The *ill-posedness* is circumvented through the introduction of prior knowledge on the sought image. A first regularization approach has been proposed by Tikhonov [Tikhonov, 1943; Tikhonov and Arsenin, 1977]. It plays an analogous role to the minus-logarithm prior of the MAP estimation and employs a quadratic functional on the estimated image. The corresponding variational model is the following

$$\text{Find } \widehat{x} \in \underset{x\in\mathbb{R}^N}{\operatorname{argmin}} \; \frac{1}{2}\|y - Hx\|^2 + \lambda\|\Lambda x\|^2, \tag{2.14}$$

where $\Lambda \in \mathbb{R}^{N\times N}$ is a linear operator that can represent for instance the identity matrix Id_N, which leads to a resulting image with low norm. In most cases, Λ corresponds to a finite differences operator denoted $\nabla \in \mathbb{R}^{2N\times N}$, thereby, a global smoothness on the image derivatives is imposed. The weight $\lambda \in]0, +\infty]$ is a *regularization parameter* which is selected to balance the contribution of the data fidelity and regularization terms in order to obtain a well restored image.

Example 2.1 Let us consider the case of an image denoising problem i.e., $H = \mathrm{Id}_N$. The denoised image can be inferred by solving the following minimization problem

$$\text{Find } \widehat{x} \in \underset{x\in\mathbb{R}^N}{\operatorname{argmin}} \; \frac{1}{2}\|x - y\|^2 + \frac{\lambda}{2}\|\nabla x\|^2. \tag{2.15}$$

where $\nabla = \begin{bmatrix} \nabla_{\mathsf{H}} \\ \nabla_{\mathsf{V}} \end{bmatrix}$ is the concatenation of gradient operators in the horizontal and vertical directions ∇_{H} and ∇_{V} respectively, that is:

$$(\nabla_{\mathsf{H}} f)_{(i,j)} = \begin{cases} f_{(i,j)} - f_{(i,j-1)} & \text{if } j > 1 \\ f_{(i,j)} & \text{if } j = 1, \end{cases} \tag{2.16}$$

(a) *Noisy image (SNR = 18.14 dB)*

(b) *Tikhonov regularization (SNR = 21.16 dB)*

(c) *Total Variation regularization (SNR = 24.30 dB)*

Figure 2.3: *Image denoising: Degraded image with Gaussian noise $\sigma = 15$ (left), restored image using the Tikhonov model (middle), restored image using the TV model (right).*

and

$$(\nabla_{\mathsf{V}} f)_{(i,j)} = \begin{cases} f_{(i,j)} - f_{(i-1,j)} & \text{if } i > 1 \\ f_{(i,j)} & \text{if } i = 1, \end{cases} \tag{2.17}$$

where $f_{(i,j)}$ denotes the pixel intensity of f at pixel (i, j). An explicit solution to Problem (2.15) is given by

$$\widehat{x} = \left(\mathrm{Id}_N + \lambda \nabla^\top \nabla\right)^{-1} y. \tag{2.18}$$

The restoration quality can be assessed with the *Signal to Noise Ratio (SNR)*, which is usually expressed as a measurement in decibels (dB) and defined as

$$\mathrm{SNR}(\widehat{x}, \overline{x}) = 20 \log_{10} \left(\frac{\|\overline{x}\|}{\|\widehat{x} - \overline{x}\|} \right). \tag{2.19}$$

The Tikhonov model can be implemented in an efficient way, however, the restored image is generally over-smoothed as shown in Figure 2.3b. Hence, the main discontinuities that characterise the image such as the edges, are not well preserved. A very popular approach to overcome the drawbacks of the Tikhonov regularization is the Total Variation (TV) model that was proposed by Rudin, Osher, and Fatemi (also known as the ROF model) [Rudin et al., 1992]. The Total Variation is defined as follows

$$(\forall z \in \mathbb{R}^{N_1 \times N_2}) \quad \mathrm{tv}(z) = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sqrt{(\nabla_h z)^2_{(i,j)} + (\nabla_v z)^2_{(i,j)}}, \tag{2.20}$$

The performance of the TV model in the context of image denoising is illustrated

Figure 2.4: *Wavelet decomposition: Original image (left), Symmlet-4 decomposition with* 1 *level of resolution (right).*

in Figure 2.3c.
Even though the difference between the Tikhonov and TV models only lies in the substitution of the ℓ_2 norm by the ℓ_1 norm, a significant improvement in terms of image quality is observed. The TV regularization penalizes small discontinuities such as noise, while it preserves large discontinuities that correspond to the edges of the image. Nevertheless, the restored images by the TV model suffer from a major drawback that is the introduction of the so-called *staircase* artifacts [Nikolova, 2009]. This is due to the fact that the minimizer of the underlying variational problem favours denoised images with piecewise constant regions. Numerous works have been undertaken in order to attenuate the staircase effect [Buades et al., 2006; Gilboa and Osher, 2008; Condat, 2014] essentially relying on higher order derivatives [Benning et al., 2013; Bredies et al., 2010; Chan et al., 2000; You and Kaveh, 2000]. Other interesting regularization strategies for image restoration have been proposed. They are based on the wavelet/frame decomposition [Mallat, 2008; Pustelnik et al., 2016] where the sparsity of the frame coefficients of the sought image is assumed. Figure 2.4 shows an example involving a symmlet-4 wavelet decomposition of an image with one level of resolution.
There exist two classes of frame-based sparse representations [Selesnick and Figueiredo, 2009; Elad et al., 2007; Wallis et al., 2017]. The first method is the *synthesis model* [Figueiredo et al., 2007; Pustelnik et al., 2010] where the image $x \in \mathbb{R}^N$ is expressed by means of its wavelet coefficients $x = Fu$, with $F \in \mathbb{R}^{N \times Q}$ a synthesis operator and $u \in \mathbb{R}^Q$ a vector of wavelet coefficients. The general form of the underlying minimization problem is

$$\widehat{x} = F\widehat{u} \tag{2.21}$$

$$\text{with } \widehat{u} = \underset{u \in \mathbb{R}^Q}{\operatorname{argmin}} \ \frac{1}{2}\|HFu - y\|^2 + \Psi(u), \tag{2.22}$$

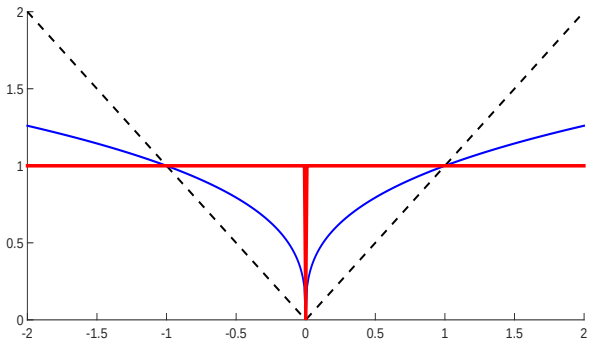


Figure 2.5: *Sparsity promoting norms: ℓ_0 norm (thick, red), ℓ_1 norm (dashed, black), and ℓ_p quasi-norm with $p = \frac{1}{3}$ (solid, blue).*

where Ψ is a sparsity inducing function, typically an ℓ_1 norm. On the other hand, there is the *analysis model*, where the image is directly processed in such a way that its frame analysis coefficients are sparse [Almeida and Figueiredo, 2013]:

$$\text{Find } \widehat{x} = \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \ \frac{1}{2}\|Hx - y\|^2 + \Psi(Ax), \tag{2.23}$$

where $A \in \mathbb{R}^{Q \times N}$ is an analysis operator and Ψ a sparsity promoting function.
Note that, when the operators F and A are orthonormal matrices, both analysis and synthesis models are equivalent. Thus, $A = F^\top = F^{-1}$ and $FF^{-1} = \mathrm{Id}_N$. This result does not apply in the overall framework, e.g., for overcomplete frames [Elad et al., 2007].

The above-mentioned regularizations involve sparsity promoting convex functions, which aim at approximating the ℓ_0 norm of the signal, i.e., the total number of its non-zero coefficients. Nevertheless, solving inverse problems regularized by the ℓ_0 norm is very challenging, since this latter function is nonconvex and discontinuous, which makes the underlying optimization problem difficult to solve. As an alternative, a number of image deconvolution algorithms resort to nonconvex but continuous regularization functions which approximate more closely the ℓ_0 norm compared to its convex relaxation by the ℓ_1 norm, as illustrated in Figure 2.5. Nonconvex regularizations are yet challenging from an optimization point of view, due to the presence of local minima that may not correspond to the sought solution. However, a number of image restoration methods employ such regularizations [Krishnan et al., 2011; Repetti et al., 2015; Perrone and Favaro, 2016; Chouzenoux et al., 2013] and achieve satisfying image quality, although leading to a local minimum.

A shared feature between the latter regularization-based methods is that they rely on penalization functions. The variational model (2.3) can also include hard

constraints through *indicator functions*. An indicator function associated to a convex subset $C \subset \mathbb{R}^N$ is denoted by ι_C, and is equal to zero if the estimated image belongs to C, and to $+\infty$ otherwise (see definition 3.2). Thus, Ψ is defined as

$$\Psi = \lambda\, \iota_C = \iota_C. \tag{2.24}$$

The set C corresponds to one constraint or to the intersection of several elementary constraints imposed on the reconstructed image $x \in \mathbb{R}^N$. Note that a regularization parameter is useless when considering hard constraints.

Example 2.2 In blind deconvolution problems where one also aims to identifying the blur filter $h \in \mathbb{R}^P$, a normalization constraint is usually defined on the convolution filter in order to avoid the *scaling ambiguity*. Moreover, depending on the imaging system, a positivity condition can also be imposed. Thus, the associated regularization function is defined by

$$\Theta = \iota_C, \tag{2.25}$$

where

$$C = \left\{ h \in [0,1]^P \;\middle|\; \sum_{p=1}^{P} h_p = 1 \right\}, \tag{2.26}$$

with h_p the p-th component of the convolution filter h.

It should be noted that a single regularization term can be represented as a sum of several functions, possibly including hard constraints. Hence, the variational model (2.3) can be expressed as:

$$\text{Find } \widehat{x} \in \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \; \Phi(x) + \sum_i \Psi_i(x), \tag{2.27}$$

where each function Ψ_i ensures some *a priori* feature of the unknown image x. This kind of regularization strategies occurs frequently in image restoration problems, as illustrated in a number of examples in Chapters 5 and 6, where video restoration problems are addressed.

2.3 Introduction to image deconvolution

Image deconvolution is a classical inverse problem in image processing. It appears in various application fields such as photography, medical imaging, astronomy and remote sensing. Despite huge improvements have been achieved in current technologies, the acquisition process and imaging materials are not yet perfect, resulting in a degraded image which is typically modelled by convolution with some impulse response of a blur kernel, also called point spread function (PSF) [Chantas et al.,

2008; Danielyan et al., 2012; Figueiredo and Nowak, 2003]. Thus, the need of deconvolution can appears in any application where image acquisition occurs. Some of these blurring processes are nonlinear or spatially variant [Campisi and Egiazarian, 2007]. However, most of contributions in image restoration assume a linear spatially invariant blur, as it is the case in this thesis. Efficiently solving the deconvolution problem has been of main interest in image processing for several decades, leading to numerous contributions in the field, which can be divided into two classes of methods depending on the number of available observed images.

2.3.1 Single-channel deconvolution

Let us first consider problems with only one degraded image, which is also refered to as single-channel deconvolution. The associated direct model is hence given by

$$y = \overline{x} * h + w, \tag{2.28}$$

where $y \in \mathbb{R}^N$ denotes the observed image, $x \in \mathbb{R}^N$ the original unknown image, $h \in \mathbb{R}^P$ denotes the PSF of the blur kernel, and $w \in \mathbb{R}^N$ represents an additive noise. Note that (2.28) is an instance of the general degradation model (2.1) in the case when the linear operator H is the Hankel-block-Hankel matrix associated to the blurring filter h. Image deconvolution can be categorized into two types: *non-blind deconvolution* and *blind deconvolution*. The latter consists in estimating both the sharp image and the blur kernel from the observation, which makes it much more difficult to solve.

Non-blind deconvolution

Although the blur kernel is known, estimating the original image remains an *ill-posed* problem due to the *ill-conditioned* nature of the convolution operators. One of the oldest deconvolution methods is Richardson-Lucy deconvolution approach, originally proposed independently by [Richardson, 1972] and [Lucy, 1974]. It is an iterative method ensuring non-negativity of the estimated image. However, this turns out to be insufficient for the cancellation of restoration artifacts and the recovery of high frequency details. Other popular deconvolution routines resort to regularization-based methods by including available *a priori* information, that ensure various statistical properties of natural images. For images with sharp changes of intensity, an appropriate regularization is based on variational integrals such as the total variation regularization [Rudin et al., 1992]. Minimizing variational integrals preserves the edges of images, and provides satisfactory results in denoising and deblurring problems [Vogel and Oman, 1999; Oliveira et al., 2009; Chan et al., 2013]. Another popular approach is to represent the unknown image as a linear combination of few coefficients (usually an overcomplete dictionary) and to enforce this sparse representation by using the ℓ_p norm with $0 \leqslant p \leqslant 1$. A solution can be obtained

in the transform domain (coefficients of the frame elements) using either a synthesis or an analysis model. Conclusions presented in [Selesnick and Figueiredo, 2009] suggest that, for deconvolution problems, the analysis approach might be preferable because sparsity should be enforced only on high-pass coefficients, which can be easily implemented in the analysis approach. It has been shown that the analysis approach can be solved efficiently using variable splitting methods such as Bregman iterative method [Goldstein and S. Osher, 2009] and augmented Lagrangian method [Afonso et al., 2010; Chan et al., 2013].

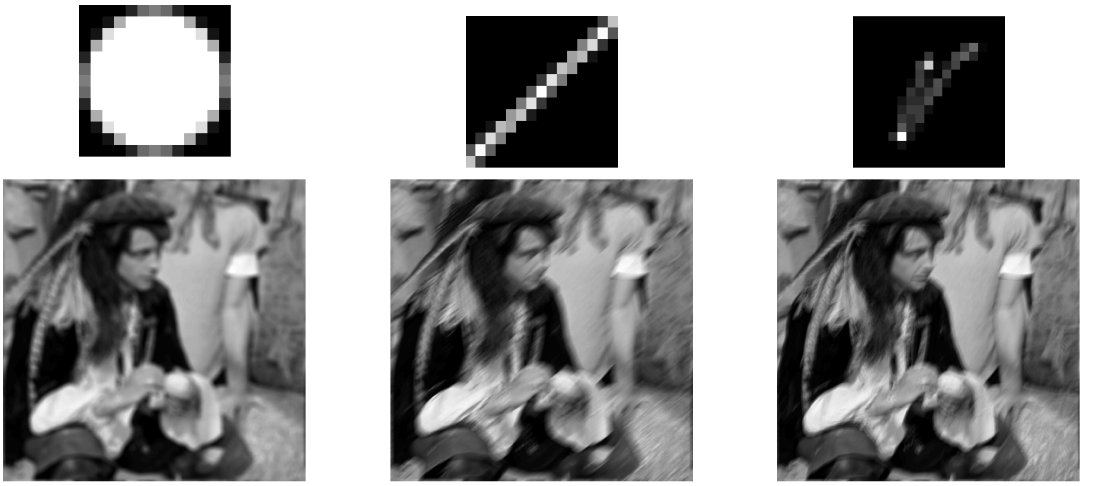
It is worth mentioning that some deconvolution methods resort to modifying the imaging device in order to obtain higher image quality, such as specially designed CMOS sensors [Liu and Gamal, 2001] and hybrid imaging systems [Ben-Ezra and Nayar, 2004; Levin et al., 2007].

Blind deconvolution

If the blur kernel is unknown, we deal then with the single-channel blind deconvolution problem, which is clearly more complicated than the classical deconvolution problem. Blind image deconvolution is severely *ill-posed* since there exist an infinity of pairs (image/blur) that lead to the same observed image. The blind case is however, a much more realistic framework that can be encountered frequently in optics, where the optical instruments are imperfect, and in photography due to misfocusing or camera shake, resulting in blurry images with unknown blur kernel. Typical kinds of blur in real-word images are thus, the out-of-focus blur and the motion blur, caused by camera shake or by the motion of the scene. Some simple examples of these blur kernels are shown in Figure 2.6.

Blind deconvolution methods fall into two categories according to the blur identification process, the *prior blur identification methods* in which the blur kernel is identified first, then the unknown image is inferred using a non-blind method [Carasso, 2011]. On the other hand, in the *joint estimation methods*, the blur kernel and unknown image are simultaneously estimated [Chan and Wong, 1998; Bar et al., 2004]. In practice the unknown image and the blur filter are estimated in an alternative manner using prior knowledge on both of them. However, in some blind problems, a parametric form of the blur is known, which makes the deconvolution process easier and reduces to estimating its specific parameters [Carasso, 2002; Chang et al., 2007; Krylov and Nasonov, 2009]. For example, a motion blur can be approximated by a linear segment characterized by its length L and its angle θ. Nevertheless, this may lead to poorly structured optimization problems.

Example 2.3 In the case of an out-of-focus blur caused by camera defocusing, a complete description of the blurring model includes the distance between the camera and the object, the aperture shape and size, focal length... Nevertheless, most of these information are not available when the picture is taken. When the blur is



(a) *Out-of-focus blur* **(b)** *Uniform motion blur* **(c)** *Nonuniform motion blur*

Figure 2.6: *Examples of blur models: Blur filters (top), associated blurred images (bottom).*

large, it can be approximated by a circular blur with radius $r > 0$ as follows

$$h_{p_1,p_2} = \begin{cases} \dfrac{1}{\pi r^2} & \text{if } \sqrt{p_1^2 + p_2^2} \leqslant r, \\ 0 & \text{otherwise.} \end{cases} \tag{2.29}$$

Another rough approximation can be obtained by the following square parametric model

$$h_{p_1,p_2} = \begin{cases} \dfrac{1}{L^2} & \text{if } |p_1| \leqslant \dfrac{L-1}{2} \text{ and } |p_2| \leqslant \dfrac{L-1}{2}, \\ 0 & \text{otherwise,} \end{cases} \tag{2.30}$$

where L is an odd integer.

However, for more complex blurs such as nonuniform motion blurs, the blurring operation cannot be described by a parametric model. In this case, regularizations are employed generally, to incorporate prior knowledge on the blur kernel and the sought image trough penalizations and hard constraints. A variety of regularization-based methods have been designed [Fergus et al., 2006; Levin, 2007; Shan et al., 2008; Joshi et al., 2008; Levin et al., 2009; Krishnan et al., 2011], which try to make use of statistical priors of natural images in order to reduce the set of possible solutions satisfying Model (2.28). They usually resort to the well known property of natural images that gradients follow a heavy tailed distribution, which implies to impose a sparsity constraint on the gradient of the unknown image. Methods of this type

have recently been able to achieve good deconvolution results. However, a number of these approaches rely on sophisticated energy minimization methods [Fergus et al., 2006; Shan et al., 2008], and as a result they can often induce a high computational cost. On the other hand, regularizations and hard constrains are also applied to the blur kernel. They usually cover the physical properties of the imaging system such as the positivity of the kernel coefficients, smoothness, sparsity and energy preservation which is guaranteed when the sum of the kernel coefficients is equal to one [Krishnan et al., 2011; Komodakis and Paragios, 2012; Kotera et al., 2013]. This last condition is widely used in blind deconvolution issues since it allows to overcome the scaling ambiguity.

2.3.2 Multichannel deconvolution

In some applications, the acquisition system is able to provide multiple observations of the original scene. In electron microscopy, for example, many differently focused version of the same image are acquired during a single experiment, due to an intrinsic trade-off between the bandwidth of the imaging system and the contrast of the resulting image. Besides, pushbroom imaging systems [Song et al., 2017] can acquire hyperspectral images in which each pixel provides local spectral information about a scene of interest across a large number of contiguous bands. In other applications, such as photography or video acquisition, continuous shooting or video capture with the camera provides several images that are blurred in a different way since our hand moves randomly (see Figure 2.7). In case of multiple measurements, the restoration algorithm can exploit the redundancy present in the observations and, in principle, it can achieve performance not obtainable from a single measure. The degradation model relating the observations $(y_k)_{1\leqslant k\leqslant K}$ to the unknown image $\overline{x}$ can be expressed as follows:

$$\forall k \in \{1,\dots,K\} \quad y_k = \overline{x} * \overline{h}_k + w_k, \tag{2.31}$$

where $\overline{h}_k$ for $k \in \{1,\dots,K\}$ denotes blur filters and $(w_k)_{1\leqslant k\leqslant K}$ is a random noise realization.

Numerous methods based on multiple observed images have been proposed to obtained a good deblurred image. Earlier works in [Harikumar and Bresler, 1999a,b; Giannakis and Heath Jr, 2000] developed algorithms for multichannel blind image restoration. They estimated the blur functions first and/or find restoration filters (deconvolver). The original image is restored using classical image deconvolution methods or by convolving the observed image with the restoration filter. In the noise-free case, the aforementioned algorithms can recover the original image, up to a scalar multiplier. However, in the noisy case, the obtained image suffers from noise amplification. Afterwards, energy minimization-based approaches have been proposed [Li et al., 2014; Chen et al., 2008; Zhuo et al., 2010; Sroubek and Milanfar, 2012]. In some of these approaches, a pair of images is employed, for example, in [Li

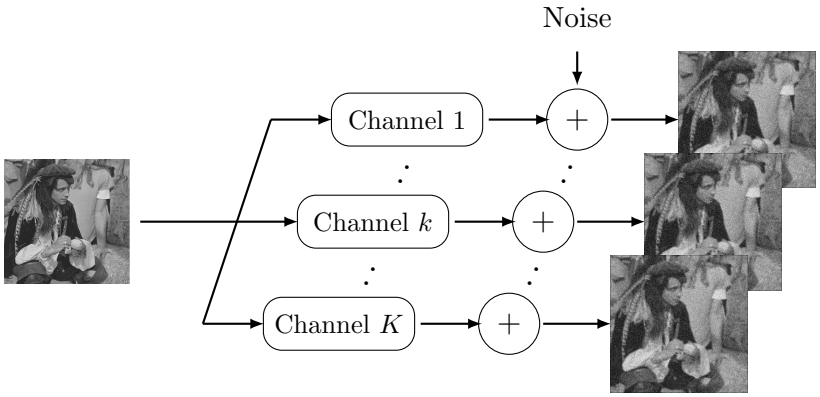


Figure 2.7: *Degradation model of the multichannel deconvolution problem : The original image is captured by K channels and corrupted with noise, resulting in K different degraded images.*

et al., 2014], a pair of blurred and noisy images is used, and in [Chen et al., 2008] two blurred images with different blur kernels are employed. The unknown image is inferred iteratively with the two blur kernels. [Sroubek and Milanfar, 2012] proposed a method that reformulates the one in [Harikumar and Bresler, 1999a] as a multichannel regularization term and simultaneously minimizes an energy function with respect to the image and blur kernels. This allows to handle inexact kernel sizes and to compensate for small misalignment in input images, which makes multichannel deconvolution more practical and able to handle large blur kernels.

In this thesis, we will propose a number of energy minimization-based blind and non-blind deconvolution approaches. Our algorithms are applied to videos with added complexity of content motion, which allows us to benefit from the advantages of the multichannel methods. In addition, when considering the non-blind problem, super-resolution is simultaneously performed on the observed video sequences.

2.4 Introduction to image super-resolution

Super-Resolution (SR) denotes the class of methods that search for one or several high resolution images (HR) from one or several low resolution images (LR) of the same scene. Spatial resolution of an image varies depending on the imaging device. Resolution grows by increasing the number of sensors in a unit area, or equivalently by decreasing the pixel size [Park et al., 2003]. However, such a strategy is not optimal since it induces an increase in the cost of the imaging device and may result in images affected by shot noise, due the decrease of the amount of light captured by each sensor. Thereby, a suitable solution to obtain sharp and high resolution images results from a compromise between hardware performance and

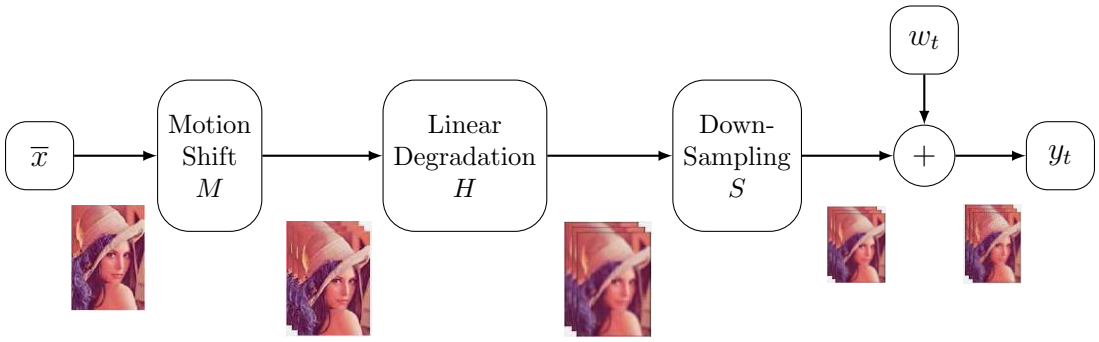


Figure 2.8: *Direct model associated to super-resolution.*

some post-processing techniques. Super-resolution techniques have been utilized in many real-word applications such as satellite image processing [Tsai and Huang, 1984; Zhang et al., 2012], medical imaging [Maintz and Viergever, 1998; Peleg and Yeshurun, 2001], surveillance video [Cristani et al., 2004; Zhang et al., 2010], facial image analysis [Zou and Yuen, 2012; Li and Chang, 2009; Ma et al., 2009], remote sensing [Tatem et al., 2001; Li et al., 2008] to name a few. They can be implemented in both spatial and frequency domains. Super-resolution methods can be classified into two categories depending on the number of involved low resolution images, *multiple-images based super-resolution* [Tsai and Huang, 1984] and *single-image based super-resolution* [Fattal, 2007; Mjolsness, 1985]. The first category gathers most popular SR algorithms which assume that K low resolution images representing the same scene from different perspectives, are supplied. Single-image based super-resolution algorithms resort to several techniques to infer the sharp details, such as regularization approaches and learning-based methods. It should be noted that interpolation techniques (usually on a single image) differ completely from super-resolution methods. The former increases the size of the image, but it does not reconstruct its missing details.

2.4.1 Multiple-image super-resolution

Multiple-images based super-resolution issues are *ill-posed* inverse problems which aim at inferring a high resolution image from the set of low resolution observations as shown in Figure 2.8. The associated direct model is the following [Irani and Peleg, 1990]

$$(\forall k \in \{1, \ldots, K\}) \quad y_k = S H_k M_k \overline{x} + w_k, \tag{2.32}$$

where $y_k \in \mathbb{R}^M$ is the k-th observed low resolution image of the target high resolution image denoted by $\overline{x} \in \mathbb{R}^N$. $M_k \in \mathbb{R}^{N \times N}$ is a linear operator modelling the displacement between the observed image y and the original scene, $H_k \in \mathbb{R}^{N \times N}$ denotes a blurring operator, and $S \in \mathbb{R}^{M \times N}$ is a down-sampling or decimation

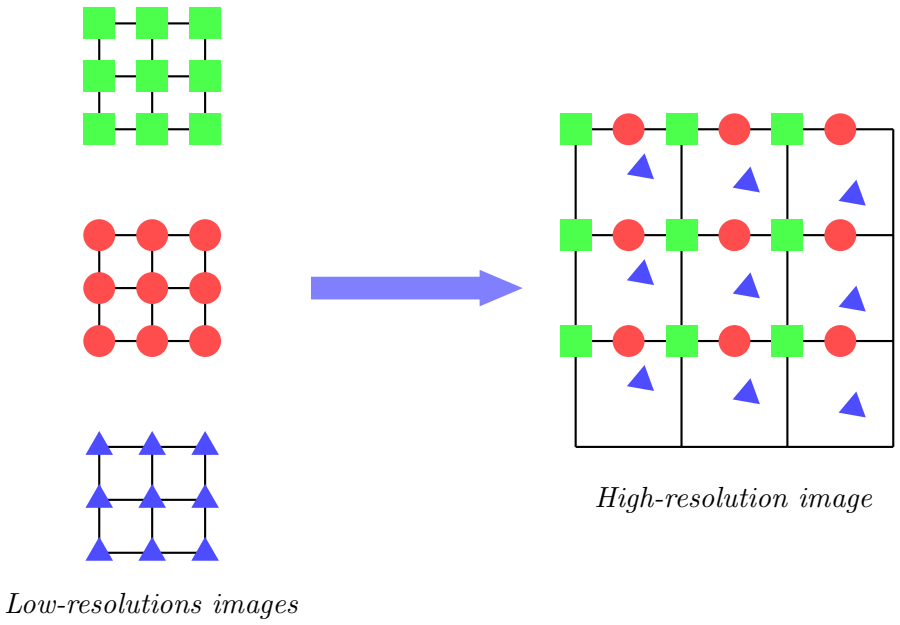


Figure 2.9: *Concept of multiple-based image super-resolution: Subpixel displacement between the LR images provide the necessary information to construct the HR image.*

operator of factor $q = N/M$. Finally, $w_k \in \mathbb{R}^M$ represents additive noise.
Figure 2.9 illustrates the idea behind multiple-image based SR, where three LR images resulting from different shifting operations of the original image followed by a decimation, are used in order to obtain the underlying HR image .

Multiple-image super-resolution in the frequency domain

The pioneer work on multiple-image based SR methods in the frequency domain has been proposed by [Tsai and Huang, 1984], where the authors exploit the aliasing and translation properties of the continuous and discrete Fourier transforms. This algorithm was initially proposed in order to process several low resolution images acquired by Landsat4 satellite. These images capture the same area of the Earth up to some translation. The initial algorithm considers noiseless images but subsequent works have been carried out in order to address the case of noisy and blurry low resolution images [Kim et al., 1990; Tekalp et al., 1992; Bose et al., 1994; Nguyen et al., 1999].
Super-resolution algorithms in the frequency domain have a low computational cost, nevertheless, they fail to achieve satisfactory results in real-world applications where the degradation models are more involved and the displacement between the LR images is not necessary translational or when there is motion in the image content.

Multiple-image super-resolution in the spatial domain

Almost all recent super-resolution techniques address the multiple-image SR problem in the spatial domain, by utilizing complementary information between the low resolution images. The simplest method to estimate the high resolution image is to decompose the problem into three sub-problems [Elad and Hel-Or, 2001; Farsiu et al., 2004]. This method assumes that the blur operator H_k is identical for all $k \in \{1, \ldots, K\}$. Hence, the motion shift operator M_k and H of model (2.32) can be reexpressed as follows

$$\begin{aligned} (\forall k \in \{1, \ldots, K\}) \quad y_k &= SM_kH\overline{x} + w_k \\ &= SM_kz + w_k, \end{aligned} \tag{2.33}$$

where $z \in \mathbb{R}^N$ is the blurry version of the unknown x with the blur operator H. The high resolution image is generated after the following simple steps:

1. a *registration* stage in-which one of LR images is chosen as reference, and the others are aligned with it at a subpixel precision [Zhao and Sawhney, 2002; Lee et al., 2010].

2. a *non-uniform interpolation* is performed based on the registered images in order to obtain a high resolution blurred image z [Nasonov and Krylov, 2010; Shimizu et al., 2008; Nasir et al., 2012].

3. a *deblurring* step is applied to the interpolated image z using a deconvolution algorithm.

This super-resolution scheme is intuitive and computationally efficient for simple observation models [Farsiu et al., 2004; Chiang and Boulte, 2000]. However, it is clearly suboptimal in the sense that the registration errors are propagated to the last stage of the method. Furthermore, the non-uniform interpolation does not guarantee good results since it does not take into account blur and noise effects. Numerous works have been performed based on the same scheme where the order of the steps has been changed [Mancas-Thillou and Mirmehdi, 2005; Thillou and Mirmehdi, 2007], or in-which the two first steps have been combined [Protter and Elad, 2009]. Super-resolution techniques may resort to the Bayesian framework to obtain an estimate of the sought high resolution image from a set of low resolution observations [Shekarforoush et al., 1996; Shen et al., 2007; Purkait and Chanda, 2012]. The proposed SR algorithms mainly vary in the choice of the prior distribution. One can mention the Gaussian Markov Random Field prior (GMRF) [Chen et al., 2012; Joshi et al., 2004] which is reminiscent of the Tikhonov model in the deterministic scheme, the Huber-based GMRF in [Chakrabarti et al., 2007; Pickup et al., 2003] and the well known Total-Variation prior in [Yuan et al., 2012].
Many MAP-based SR algorithms assume that the motion operators are known or

pre-estimated, however, there exist variants which estimate the high resolution image and the motion operators jointly [Tom and Katsaggelos, 1995; Segall et al., 2004]. The estimation of the high resolution image and the computation of motion parameters can benefit one from each other to achieve a good image quality. Nevertheless, they are computationally more expensive.

2.4.2 Single-image super-resolution

Multiple-image super-resolution algorithms utilize a set of LR images to reconstruct the sharp details of the sought HR image. However, in various real-world applications, several low resolution observations of the same scene is not available. As an extreme but frequent situation, only one single LR image can be provided. One of the proposed approaches to address the single-image SR problem tries to solve the underlying *ill-posed* problem through the introduction of prior knowledge on the expected high resolution image. This class of methods is termed *reconstruction-based methods* [Fattal, 2007; Fan and Yeung, 2007; Dai et al., 2009] and aims at reconstructing the missing details of the upscaled image while preserving the continuity and sharpness of the edges [Li and Orchard, 2001; Dai et al., 2007; Fattal, 2007]. This can also be performed by considering regularization functions that preserve strong discontinuities in the spatial domain such as the Total Variation regularization [Aly and Dubois, 2005], or in the frequency domain through wavelet transforms, where the wavelet coefficients of the estimated HR image are enforced to be sparse [Sen and Darabi, 2009].

Another class of single-image SR algorithms is referred to as *learning-based methods* [Mjolsness, 1985; Capel and Zisserman, 2001; Zou and Yuen, 2012] and resorts to trained dictionaries in order to infer fine details. The dictionary contains pairs of HR images/patches and the corresponding degraded LR images/patches. In most learning-based methods, the image is partitioned into (possibly overlapping) square patches of pixels. For each input LR patch, a set of neighboor LR patches from the dictionary is selected. The candidate LR patches are weighted and a reconstruction model is derived from the selected low and high resolution pairs. Afterwards, this model is applied to the input LR patch in order to reconstruct its associated HR patch [Stephenson and Chen, 2006]. Finally, the HR image is obtained by merging all the reconstructed HR patches as illustrated in Figure 2.10.

The dictionary can be *external* i.e., built from external training images, or *internal*, namely the dictionary is built from the input LR image itself. The latter type of dictionaries relies on the *self-similarity* property which states that image structures of natural images are repeated across different scales [Lu et al., 2012]. Hence, for a given input LR patch, a correspondence can be found within the same image, possibly with different scales and locations.

Learning-based methods have shown their efficiency in recovering sharp high resolution images. However, a number of parameters have to be appropriately adjusted,

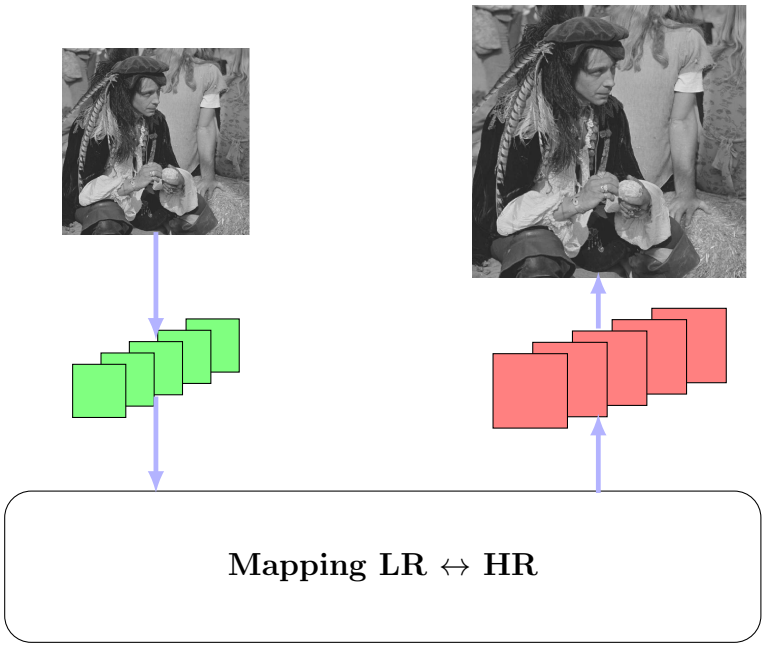


Figure 2.10: *Generic diagram of learning-based single-image SR algorithms: the LR image is partitioned into patches, Afterwards, the HR patches are constructed thanks to the self-similarity property and using a dataset of (LR/HR) patches. Finally the HR image is generated by assembling all the HR patches.*

such as the patch size and the content and size of the dictionary, which has a strong impact on the performance of these methods. In fact, large-size datasets do not necessary yield an improvement of the SR results since irrelevant image may increase the search time and lead to poorly reconstructed HR images [Li et al., 2009]. Thereby, the dictionary should involve images sharing the same statistics (e.g., fingerprint or face database).

Recent learning-based super-resolution approaches rely on machine learning techniques and more precisely on Convolutional Neural Networks (CNN) [LeCun et al., 1989]. The CNN are mainly inspired from biological nervous system and consist of computational processing units termed *neurons*, which exchange and communicate via weighted connections that are adjusted during the training step. A CNN is made up of several layers. Each layer receives inputs which represent features of the LR image from the previous one, processes them and then sends them to the next layer, as illustrated in Figure 2.11. Recently, after the phenomenal success of convolutional neural networks in image classification problems [Krizhevsky et al., 2012; He et al., 2014] CNN-based super-resolution methods attracted an increasing attention. In this context, we mention the works in [Dong et al., 2014; Romano et al., 2016], which propose a patch-based SR approach in-which the input LR image is upscaled using a cheap interpolation (e.g., Bilinear, Bicubic) and then partitioned into over-

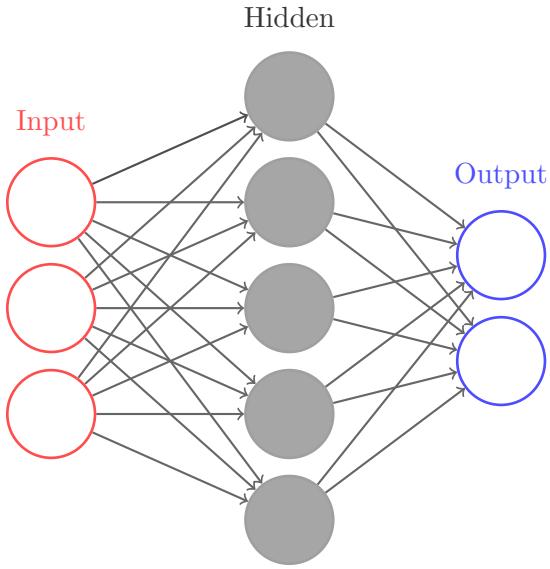


Figure 2.11: *Convolutional Neural Network composed of* 3 *layers and* 10 *neurons.*

lapping patches which are refined using a small trained CNN, on a large number of examples, and then merged to reconstruct the HR image. Similar approaches have been proposed in [Shi et al., 2016; Wang et al., 2016] where, instead of interpolating the LR image, an upscaling filter is learned and applied at the final layer. This allows to reduce the computational cost while keeping good super-resolution results.

2.5 Introduction to video restoration

We are interested in this thesis in two main aspects of video restoration, namely video deconvolution and super-resolution. The developed techniques are usually inspired from multi-image deconvolution and super-resolution methods that we have discussed in the previous section, and further exploit the temporal correlation existing between neighbor frames. A particular attention is paid in this manuscript to the restoration of videos from old television archive videos.

2.5.1 Video deconvolution

Real-life video sequences are usually blurred. This results from the overall effect of different factors such as defocus, motion blur, and optical blur. However, motion blur represents one of the artifacts that causes the most visually annoying blurry images. In addition, videos are subject to spatial and temporal aliasing which appears in images or video frames when the cut-off frequency of the sensor is lower

than that of the lens. Temporal aliasing arises in video sequences when the frame rate of the camera is not fast enough to capture high frequencies caused by fast moving objects. As for single images, video deblurring is usually formulated as an inverse deconvolution problem which aims at jointly estimating the blurring operator and the underlying sharp video sequence. The t-th frame of the latter is related to the observation as follows

$$(\forall t \in \{1, \dots, T\}) \quad y_t = \overline{h}_t * \overline{x}_t + w_t, \tag{2.34}$$

where T denotes the number of frames of the video sequence, $\boldsymbol{y} = (y_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$ is the observed video sequence, $\boldsymbol{x} = (x_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$ is the original sharp video sequence, $(h_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TP}$ denotes spatial convolution kernels, and $(w_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$ represents additive noise.
Video deblurring is highly related to multi-image deconvolution [Paragios et al., 2006; Chen et al., 2008; Cai et al., 2009; Sroubek and Milanfar, 2012]. It has been shown in [Cai et al., 2009] that given multiple observations, enforcing the frame sparsity improves the accuracy of identifying the blur kernels and reduces the *ill-posedness* of the problem. However, multi-image deconvolution algorithms require that all the input images are aligned and that the content is the same (static scene). On the other hand, the authors in [Li et al., 2010] proposed to estimate the camera motion and to explicitly model the video blur as a function of the motion being estimated. A joint energy function is formulated between the underlying sharp sequence and motion parameters. Recently, [Kim and Lee, 2015] proposed to simultaneously tackle the problem of optical flow estimation and frame restoration in general blurred videos. This is done by simultaneously estimating the optical flow and latent sharp frames through the minimization of a single nonconvex energy function. Addressing these two problems simultaneously requires a much more complex optimization, due to the more sophisticated direct model linking all the blurry observations.

2.5.2 Video super-resolution

Video super-resolution consists in estimating high-resolution frames from low-resolution input sequences. Video super-resolution has become one of the fundamental problems in image and video processing that has been extensively investigated for decades. With the popularity of high-definition display devices, such as High-definition television (HDTV), or even Ultra-high-definition television (UHDTV), on the market, there is an increasing demand for transferring LR videos into HR videos so that they can be displayed on high-resolution television screens. The sought HR frames are related through warping based on motion fields. They are smoothed with blur kernels, down-sampled and corrupted with additive noise to generate the observed LR frames. The direct degradation model is therefore given by

$$(\forall t \in \{1, \dots, T\}) \quad y_t = S(\overline{h}_t * \overline{x}_t) + w_t, \tag{2.35}$$

where $S \in \mathbb{R}^{M \times N}$ is a decimation operator. There have been great advances in super-resolution algorithms recently, and some of these works handle the case of video sequences. Among them, we mention the works in [Liu and Sun, 2011, 2014] in which a Bayesian approach is proposed for simultaneously estimating the HR frames, motion blur and noise parameter. The regularization terms for all unknowns are ℓ_1 norms. An estimated noise parameter is used to update the weight of the fidelity term at each iteration of the optimization algorithm. Promising results are shown for $4\times$ upscaling of real-life videos, but only Gaussian blurs are considered. [Ma et al., 2015] presented an algorithm that extended the same idea to handle motion blur. [Zhou et al., 2012] proposed to retrieve high-frequency details from complementary multiframes by non-uniform interpolation, depending on registered LR frames with sub-pixel accuracy. They further improved the SR performance in [Zhou et al., 2014] when the number of LR input frames is small, by taking advantage of nonlocal self-similarity to fit local surfaces. [Liao et al., 2015] proposed to apply the multi-frame SR method of [Elad and Hel-Or, 2001] to obtain SR drafts with different motion estimation parameters, and then to combine them through a deep convolutional neural network (CNN).

2.5.3 Television archives

The last century has witnessed an explosion in the amount of video data stored with holders such as Institut National Audiovisuel (France), British Broadcasting Company (United Kingdom), Radiotelevisão Portuguesa (Portugal), Beeld en Geluid (Netherlands) and Library of Congress (USA) to name a few. Beyond the cultural heritage these data represent, their value is increased by their commercial reexploitation through digital visual media [Elgharib et al., 2013]. This requires the archived data to be compatible with the current standards of visual quality. This, however, is usually not the case as the data are often visually degraded during their recording and long-term storage under poor physical and climate condition. Television archives are affected by several artifacts such as blur, echoes and many types of noise. Among them, let us mention the Gaussian noise, the grain noise [Naranjo et al., 2006], and the flicker noise which consists in global intensity variations between two consecutive frames. Noise reduction is essential for high image quality, however, removing all of it can be undesirable, especially for the grain noise since it can be considered as part of the video or film "feel" [Kokaram, 1998]. The content of video archive documents can be corrupted by other kinds of impairments such as blotches that occur mainly due to dirt particles adhering to the film material or due to film abrasion. Blotches appear as temporally impulsive dark and bright spots which are distributed randomly over an image sequence. Another type of artifacts that may rise in analog television records is the ghost effect [Dhake, 1999]. It appears as a weak replica of the transmitted image. In fact, waves of television signals can take paths of different length and some of them may be reflected and attenuated

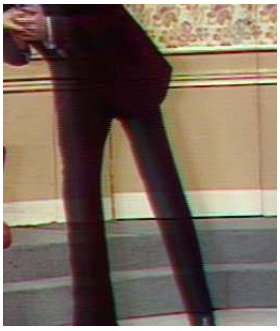


Figure 2.12: *Ghost effect that appears as a shadow at the right of the main image.*

by walls of buildings. The reflected signal is then weaker than the one committed directly from the transmitter. This causes a lagging ghost image since the reflected signal reaches the receiver device after a longer path, appearing as a displacement in time as shown in the example in Figure 2.12.

Unlike films, in which the whole image is projected on a screen at once, a television video frame is composed of rapidly scanning lines across a screen starting at the top of the screen and moving to the bottom [Dhake, 1999]. These lines can be scanned in two ways. The first way is to split the lines into two fields in which all of the odd numbered lines are scanned first and then all of the even numbered lines are displayed next, producing a complete frame. This process is called *interlacing* or *interlaced scan* [Mallat, 2006]. Figure 2.13 shows an example of an interlaced frame composed of odd and even fields. The second method, is referred to as *progressive scan* which allows the lines to be displayed sequentially. This means that both the odd and even numbered lines are displayed in the same image. Progressive scan is more and more used in digital video recording, digital televisions, and computer monitors.

Several analog television standards have been around for six decades: the *National Television Systems Committee* (NTSC) which is used mainly in north America and Japan, the *Séquentiel Couleur Avec Mémoire* (SECAM) that is adopted in France and Russia, and the *Phase Alternating Line rate* (PAL) used in western Europe and Asia. The aforementioned standards are quite similar and differ mainly in image resolution, frequency sampling, and color representation. The NTSC standard is a 60 fields/30 frames-per-second system for transmission and display of video images. It is based on interlaced scan, where each frame is decomposed in two fields of 244 lines. Besides, PAL and SECAM format for video broadcasting and display are interlaced systems clocked at 50 field/25 frames per second with 288 lines per field.

With the new large HD screens and flat panel televisions such as Plasma and LCD, the resolution produced by traditional television are not reproduced nicely from the interlaced scanning method. To overcome this limitation, the concept of *deinterlacing* was introduced. Deinterlacing is the process of constructing a progressive video

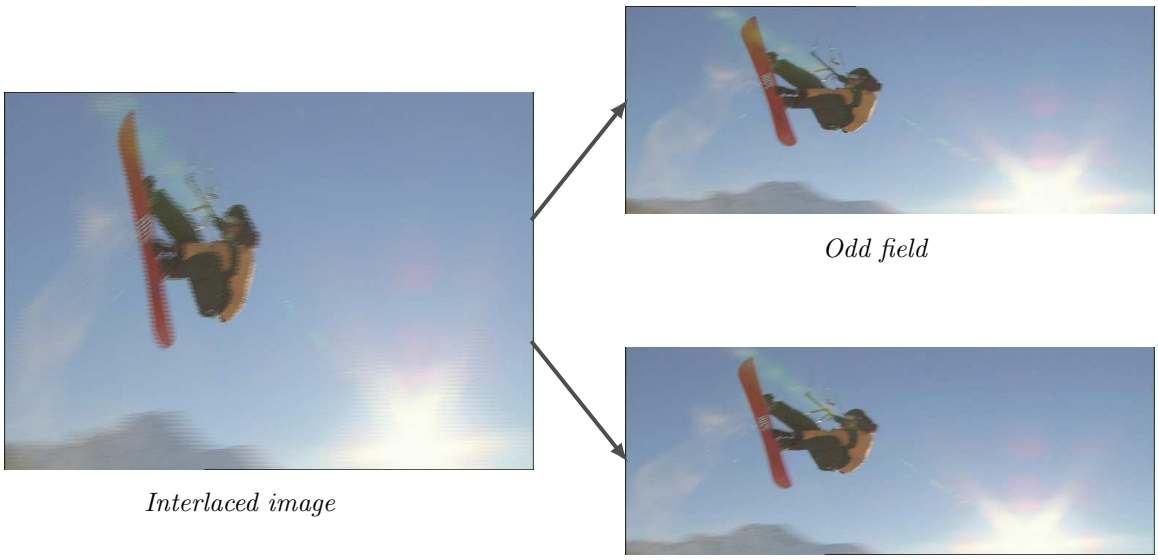


Figure 2.13: *Interlaced image (left), its decomposition into odd and even fields (right).*

sequence from an interlaced one. This is realized by splitting each interlaced frame into an odd and even fields, and then estimating the missing even/odd rows using a super-resolution technique, as illustrated in Figure 2.14.
We provide in Chapter 5 a new joint video deblurring and deinterlacing method based on a variational formulation of the problem. We define an energy function which is minimized by a recent iterative optimization algorithm, that will be presented in Chapter 4.

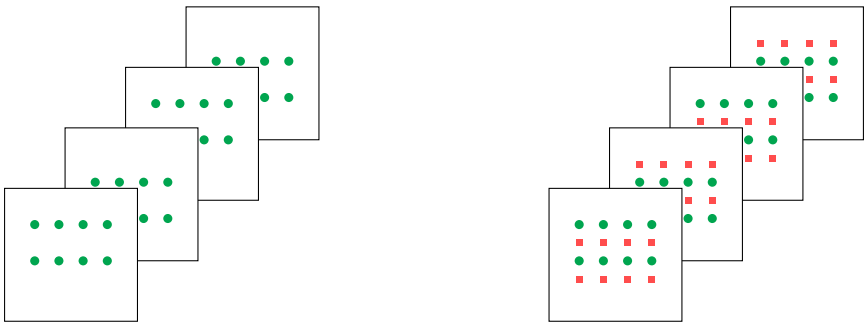


Figure 2.14: *Deinterlacing as a super-resolution problem: interlaced video of four odd and even fields (left). Progressive video with the estimated missing rows displayed as red squares (right).*

2.6 Conclusion

We have seen in this chapter that numerous image and video processing issues can be cast as inverse problems whose solution is obtained through the resolution of an optimization problem. The latter consists in minimizing an objective function composed of a data fidelity and regularization terms. Regularization functions are chosen according to established assumptions on sharp natural images in order to enhance the perceived quality. This can be done either by exploiting the degraded image itself, or by utilizing a number of images, provided that correlations exist between them. In the next chapter, we shall detail a number of optimization methods that are widely used in image and video restoration problems.

- Chapter 3 -

Optimization methods

Contents

3.1 Introduction

An efficient way to address image and video restoration issues is to express them as optimization problems which aim at minimizing a sum of data fidelity and regularization terms. Usually, optimization problems do not have an explicit and closed form solution, so an iterative algorithm is developed for this purpose. These algorithms depend heavily on the mathematical properties of the terms constituting the cost function. Large efforts have been made during the last decades in developing efficient algorithms for solving a wide range of optimization problems, while being careful to the complexity. This chapter is dedicated to the presentation of some fundamental tools in large-scale optimization. First, we recall in Section 3.2 some basic definitions and set up our notations, then we give in Section 3.3 a number of efficient minimization algorithms.

3.2 Definitions and notation

We recall below some definitions and theorems of convex optimization analysis, and introduce the notation that will be used throughout the thesis.

Definition 3.1 Let ψ be a function from $\mathbb{R}^N$ to $]-\infty,+\infty]$

(i) The domain of ψ, denoted by $\operatorname{dom}\psi$, is given by

$$\operatorname{dom}\psi = \{x \in \mathbb{R}^N \;\mid\; \psi(x) < +\infty\}.$$

(ii) ψ is proper if and only if its domain $\operatorname{dom}\psi$ is nonempty.

(iii) ψ is coercive if

$$\lim_{\|x\|\to+\infty} \psi(x) = +\infty.$$

(iv) ψ is lower-semicontinuous at $x_0 \in \mathbb{R}^N$ if

$$\lim_{x\to x_0} \inf \psi(x) \geqslant \psi(x_0).$$

ψ is said to be lower-semicontinuous if the above inequality is satisfied for every $x_0 \in \mathbb{R}^N$.

Definition 3.2 Let E be a subset of $\mathbb{R}^N$. The indicator function relative to the set E is denoted by $\iota_E : \mathbb{R}^N \to]-\infty,+\infty]$ and defined as

$$\iota_E(x) = \begin{cases} 0 & \text{if } x \in E, \\ +\infty & \text{otherwise.} \end{cases} \tag{3.1}$$

Definition 3.3 [Bauschke and Combettes, 2017] Let C be a subset of $\mathbb{R}^N$. C is convex if

$$(\forall \lambda \in]0,1[)\ \big(\forall (x,y) \in C^2\big) \quad \lambda x + (1-\lambda)y \in C.$$

Definition 3.4 [Bauschke and Combettes, 2017] Let ψ be a function from $\mathbb{R}^N$ to $]-\infty,+\infty]$

(i) The function ψ is convex if

$$(\forall \lambda \in]0,1[)\ \big(\forall (x,y) \in (\mathrm{dom}\psi)^2\big) \quad \psi(\lambda x + (1-\lambda)y) \leqslant \lambda\psi(x) + (1-\lambda)\psi(y).$$

(ii) ψ is strictly convex if

$$(\forall \lambda \in]0,1[)\ \big(\forall (x,y) \in (\mathrm{dom}\psi)^2, x \neq y\big) \quad \psi(\lambda x + (1-\lambda)y) < \lambda\psi(x) + (1-\lambda)\psi(y).$$

(iii) ψ is strongly convex with constant β if $\psi - (\beta/2)\|\cdot\|^2$ is convex.

Note that if ψ is convex, then its domain dom ψ is convex.

Definition 3.5 Let ψ and φ be functions from $\mathbb{R}^N$ to $]-\infty,+\infty]$. The infimal convolution of ψ and φ is defined as

$$\psi \,\square\, \varphi : \mathbb{R}^N \to [-\infty,+\infty] : x \to \inf_{y \in \mathbb{R}^N} \left(\psi(y) + \varphi(x-y)\right). \tag{3.2}$$

The neutral element of the infimal convolution corresponds to the indicator function of the set containing the zero vector of $\mathbb{R}^N$ denoted by $\{0\}$:

$$\big(\forall x \in \mathbb{R}^N\big) \quad \psi \,\square\, \iota_{\{0\}}(x) = \psi(x).$$

The Moreau envelope of ψ of parameter $\gamma > 0$ is

$${}^{\gamma}\psi = \psi \,\square\, \left(\frac{1}{2\gamma}\|\cdot\|^2\right). \tag{3.3}$$

Definition 3.6 [Bauschke and Combettes, 2017; Rockafellar, 1974] Let $\psi : \mathbb{R}^N \to [-\infty,+\infty]$ be a proper function. ψ^* denotes the conjugate function of ψ (see Figure 3.1) and is defined as follows:

$$\psi^* : \mathbb{R}^N \to [-\infty,+\infty] : u \to \sup_{x \in \mathbb{R}^N} \left(\langle x|u\rangle - \psi(x)\right). \tag{3.4}$$

Definition 3.7 Let $A \in \mathbb{R}^{N\times N}$ and $B \in \mathbb{R}^{N\times N}$ be symmetric positive definite matrices. The Loewner partial order is defined as

$$A \succeq B \ \ (\text{resp. } A \succ B) \ \Leftrightarrow\ (\forall x \in \mathbb{R}^N \setminus \{0\})\ x^\top A x \geqslant x^\top B x \ \ (\text{resp. } x^\top A x > x^\top B x). \tag{3.5}$$

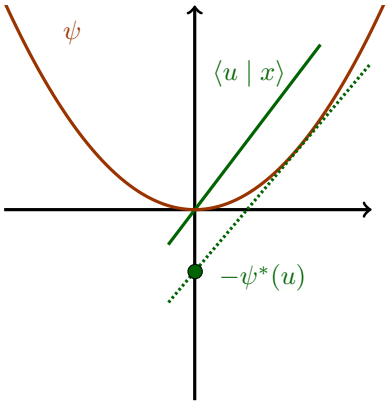


Figure 3.1: *$\psi^*(u)$ is the supremum of the signed vertical difference between ψ and the continuous linear functional $\langle \cdot \mid u \rangle$.*

The weighted dot product and norm associated with a symmetric positive definite matrix $A \in \mathbb{R}^{N \times N}$, will be denoted respectively by

$$\left(\forall x \in \mathbb{R}^N\right) \left(\forall y \in \mathbb{R}^N\right) \quad \langle x|y\rangle_A = \langle x|Ay\rangle \quad \text{and} \quad \|x\|_A = \langle x|Ax\rangle^{\frac{1}{2}} .$$

Note that, if $A = \mathrm{Id}_N$ the identity matrix of $\mathbb{R}^N$, the weighted norm $\|x\|_A$ reduces to the standard euclidean norm $\|x\|$.

Definition 3.8 [Bauschke and Combettes, 2017; Rockafellar, 1974] Let $\Gamma_0(\mathbb{R}^N)$ denote the set of convex proper lower-semicontinuous functions from $\mathbb{R}^N$ to $]-\infty, +\infty]$, and $\psi \in \Gamma_0(\mathbb{R}^N)$. The Moreau sub-differential of ψ at $x \in \operatorname{dom} \psi$ is defined as follows (see Figure 3.2 for an example)

$$\partial\psi(x) = \left\{t \in \mathbb{R}^N : \ \forall y \in \mathbb{R}^N, \quad \psi(y) - \langle y - x \,|\, t\rangle \geqslant \psi(x)\right\}. \tag{3.6}$$

Proposition 3.9

(i) If $\psi : \mathbb{R}^N \to]-\infty, +\infty]$ is a proper function, then

$$x \in \operatorname{argmin} \ \psi \quad \Leftrightarrow \quad 0 \in \partial\psi(x),$$

where 0 denotes the zero vector of $\mathbb{R}^N$.

(ii) If $\psi \in \Gamma_0(\mathbb{R}^N)$ is Gâteaux differentiable, then

$$\left(\forall x \in \mathbb{R}^N\right) \quad \partial\psi(x) = \{\nabla\psi(x)\},$$

where $\nabla\psi(x)$ is the gradient of ψ at x.

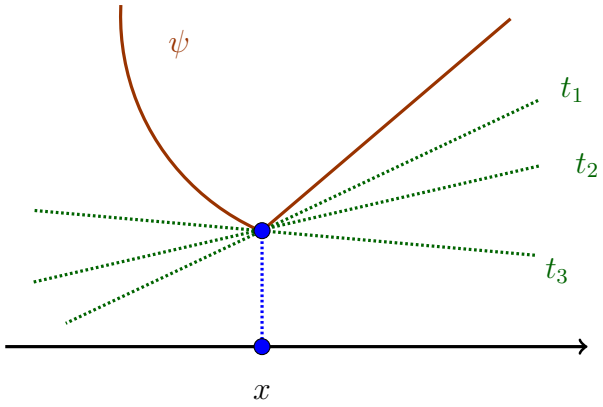


Figure 3.2: *An example of three sub-gradients, t_1, t_2 and t_3 of ψ at x.*

Definition 3.10 The differentiable function $\psi : \mathbb{R}^N \to]-\infty, +\infty]$ has a β-Lipschitz gradient on a subset E of $\mathbb{R}^N$ with $\beta > 0$, if

$$\big(\forall (x, y) \in E^2\big) \quad \|\nabla\psi(x) - \nabla\psi(y)\| \leqslant \beta\|x - y\|.$$

Definition 3.11 Let ψ be a function in $\Gamma_0(\mathbb{R}^N)$. The proximity operator of ψ at $\widetilde{x} \in \mathbb{R}^N$ is defined as the unique minimizer of the following problem [Moreau, 1965]

$$\text{prox}_{\psi}(\widetilde{x}) = \underset{x\in\mathbb{R}^N}{\text{argmin}} \;\; \psi(x) + \frac{1}{2}\|x - \widetilde{x}\|^2. \tag{3.7}$$

Moreover, for every $\gamma > 0$, $\text{prox}_{\gamma\psi}(\widetilde{x})$ is given by

$$\text{prox}_{\gamma\psi}(\widetilde{x}) = \underset{x\in\mathbb{R}^N}{\text{argmin}} \;\; \psi(x) + \frac{1}{2\gamma}\|x - \widetilde{x}\|^2. \tag{3.8}$$

Thus $\text{prox}_{\gamma\psi}(\widetilde{x})$ corresponds to the point where the infimum is reached in the calculation of the Moreau envelope of parameter γ of ψ at $\widetilde{x}$.
The proximity operator $\text{prox}_{\psi}(x)$ is characterized by the following inclusion

$$\big(\forall (x, p) \in \mathbb{R}^N \times \mathbb{R}^N\big) \quad p = \text{prox}_{\psi}(x) \;\Leftrightarrow\; x - p \in \partial\psi(p). \tag{3.9}$$

Definition 3.12 Let $\psi \in \Gamma_0(\mathbb{R}^N)$ and $B \in \mathbb{R}^{N\times N}$ be a symmetric positive definite matrix. The proximity operator of ψ at $\widetilde{x} \in \mathbb{R}^N$ relative to the metric induced by B is denoted by $\text{prox}_{B,\psi}(\widetilde{x})$ and defined as

$$\text{prox}_{B,\psi}(\widetilde{x}) \;=\; \underset{x\in\mathbb{R}^N}{\text{argmin}} \;\; \psi(x) + \frac{1}{2}\|x - \widetilde{x}\|_B^2. \tag{3.10}$$

The proximity operator of ψ^*, the conjugate function of ψ, can be expressed by means of $\text{prox}_{B,\psi}$ using the Moreau decomposition formula given by

$$\text{prox}_{B,\psi^*} = \text{Id} - B^{-1}\text{prox}_{B^{-1},\psi}(B\,\cdot). \tag{3.11}$$

When B is equal to the identity matrix of $\mathbb{R}^N$, one retrieves the classical proximity operator in (3.7). Besides, when $B = \gamma^{-1}\text{Id}_N$, (3.8) is recovered.

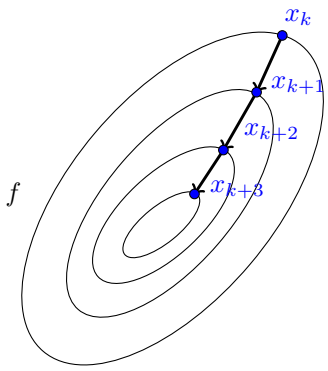


Figure 3.3: *Iterative gradient descent for minimizing a function f. Each generated iterate has a lower cost than its predecessor.*

3.3 Optimization algorithms

As pointed out in Chapter 2, a solution to inverse problems, and especially to image and video restoration issues, can be obtained by formulating them as minimization problems. The adopted resolution strategy depends heavily on the mathematical properties of the involved functions. Hereafter, we will present some of the main algorithms that are employed in the field of inverse problems in image processing.

3.3.1 Gradient descent algorithm

Let us first consider the problem of minimizing a differentiable function $f \in \Gamma_0(\mathbb{R}^N)$ with a β-Lipschitz gradient, where $\beta > 0$, namely

$$\text{Find } \widehat{x} \in \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \; f(x). \tag{3.12}$$

Then, the solution $\widehat{x}$ to problem (3.12) satisfies

$$\nabla f(\widehat{x}) = 0. \tag{3.13}$$

Iterative algorithms are usually employed to find a solution to minimization problems. This is achieved by constructing a sequence $(x_n)_{n \in \mathbb{N}}$ converging toward a minimizer. A well-known algorithm for solving (3.12) is the *gradient descent algorithm* [Bertsekas, 1999] which relies on the evaluation of the gradient of f at each iteration $n \in \mathbb{N}$ in order to identify a direction leading to a minimizer, as illustrated in Figure 3.3.

Algorithm 1 Gradient descent algorithm

Initialization:

Let $x_0 \in \mathbb{R}^N$

For every $n \in \mathbb{N}$, and $0 < \gamma_n < 2\beta^{-1}$

for $n = 0, 1, \ldots$ **do**

$x_{n+1} = x_n - \gamma_n \nabla f(x_n)$

end for

3.3.2 Proximal point algorithm

In various optimization problems, the function f in $\Gamma_0(\mathbb{R}^N)$ to minimize is not differentiable. A popular approach to address such case is to express the next iterate x_{n+1}, for $n \in \mathbb{N}$, as a function of the current one, as follows [Rockafellar, 1976]

$$x_{n+1} = x_n - \gamma_n t_n \quad \text{with } t_n \in \partial f(x_{n+1}) \text{ and } \gamma_n > 0. \tag{3.14}$$

Thereby, we have

$$x_n - x_{n+1} \in \gamma_n \partial f(x_{n+1}). \tag{3.15}$$

Using the characterization of the proximity operator in (3.9), (3.15) is equivalent to

$$x_{n+1} = \operatorname{prox}_{\gamma_n f}(x_n). \tag{3.16}$$

Thus, the *proximal point algorithm* given by Algorithm 2 can be derived.

Algorithm 2 Proximal point algorithm

Initialization:

Let $x_0 \in \mathbb{R}^N$

For every $n \in \mathbb{N}$, $\gamma_n \in]0, +\infty[$

for $n = 0, 1, \ldots$ **do**

$x_{n+1} = \operatorname{prox}_{\gamma_n f}(x_n)$

end for

When f is is in $\Gamma_0(\mathbb{R}^N)$, $\operatorname{prox}_{\gamma_n f}(x_n)$ is unique. The convergence of Algorithm 2 is then established by Theorem 3.13.

Theorem 3.13 *[Bauschke and Combettes, 2017] Let f be a function in $\Gamma_0(\mathbb{R}^N)$ with* argmin $f \neq \varnothing$. *Assume that $\sum_{n\in\mathbb{N}} \gamma_n = +\infty$, then the sequence $(x_n)_{n\in\mathbb{N}}$ generated by Algorithm 2 converges to the solution to problem* (3.12).

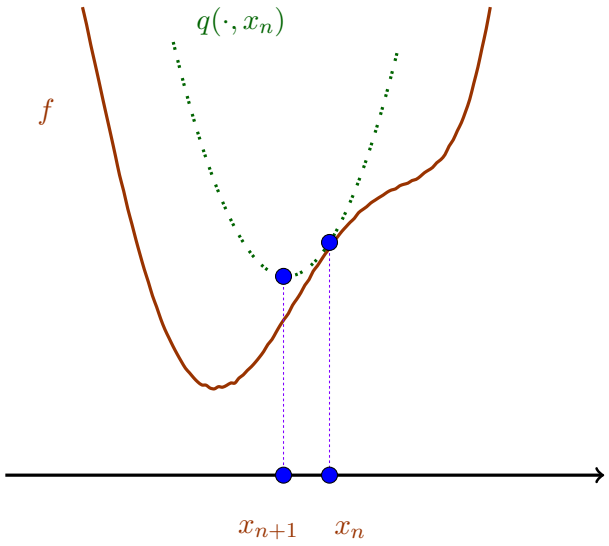


Figure 3.4: *An illustrative example of the Majorize-Minimize strategy: at each iteration $n \in \mathbb{N}$, a tangent majorant $q(\cdot, x_n)$ of f at x_n is built and the next iterate x_{n+1} is defined as the minimizer of $q(\cdot, x_n)$.*

3.3.3 Majorize-Minimize strategy

Another minimization strategy can be obtained by resorting to the Majorize-Minimize (MM) framework. MM techniques represent a class of iterative methods that substitute at each iteration $n \in \mathbb{N}$, the function $f : \mathbb{R}^N \to]-\infty, +\infty]$ to be minimized by a surrogate *tangent majorant* function $q : \mathbb{R}^N \times \mathbb{R}^N \to]-\infty, +\infty]$ which fulfils the following conditions at $x' \in \mathbb{R}^N$:

$$\begin{aligned} \left(\forall x \in \mathbb{R}^N\right) \quad & q(x, x') \geqslant f(x) \\ & q(x', x') = f(x'). \end{aligned} \tag{3.17}$$

An illustration of MM principle is shown in Figure 3.4. A generic MM algorithm is given by Algorithm 3 [Ortega and Rheinboldt, 1970].

Algorithm 3 Majorize-Minimize algorithm

Initialization:

Let $x_0 \in \mathbb{R}^N$

for $n = 0, 1, \ldots$ **do**

$$x_{n+1} = \operatorname*{argmin}_{x \in \mathbb{R}^N} \; q(x, x_n)$$

end for

Majorize-Minimize methods require to construct a sequence of tangent majorant functions. These functions may depend on the previous iterate or not [Jacobson

and Fessler, 2007; Hunter and Lange, 2004]. However, the common focus of these methods is that the surrogate function has to be computationally simpler to minimize than the original one. One can find in [Lange, 2010; Erdogan and Fessler, 1999] several methods for constructing such majorant functions.

An interesting case of MM techniques is provided when f is a differentiable function with a Lipschitz gradient. Then, the tangent majorant function q can be chosen quadratic and defined as follows: Let $x_n \in \mathbb{R}^N$ be the vector obtained at iteration $n \in \mathbb{N}$,

$$\left(\forall x \in \mathbb{R}^N\right) \quad q(x, x_n) = f(x_n) + \langle x - x_n \mid \nabla f(x_n)\rangle + \frac{1}{2}\|x - x_n\|^2_{A_n}, \tag{3.18}$$

with $A_n \in \mathbb{R}^{N\times N}$ a symmetric semi-definite positive matrix that allows to satisfy the majoration condition (3.17) for f at x_n. Within this context, the MM algorithm can be formulated as a preconditioned version of Algorithm 1:

Algorithm 4 Quadratic Majorize-Minimize algorithm

Initialization:

Let $x_0 \in \mathbb{R}^N$

for $n = 0, 1, \ldots$ **do**

$x_{n+1} = x_n - A_n^{-1}\nabla f(x_n)$

end for

Remark 3.14 if $f : \mathbb{R}^N \to]-\infty, +\infty]$ is a differentiable function with a β-Lipschitz gradient, then the quadratic function $q(\cdot, x_n)$ defined according to (3.18) with $A_n = \alpha\,\mathrm{Id}_N$ with $\alpha \geqslant \beta$, is a tangent majorant function of f at x_n. In that case, Algorithm 1 is recovered.

Note that many works have been supplied in which non-necessarily quadratic majorant functions are utilized [Bolte and Pauwels, 2016; Ochs et al., 2015], which allows to address a larger class of problems.

3.3.4 Forward-backward algorithm

Let us now consider minimization problems involving a sum of two functions with different properties. This kind of problems is widely encountered in image restoration tasks where an estimate of the sought image is obtained by minimizing the sum of a data fidelity term and a regularization term. In other words, let us consider the following problem:

$$\text{Find } \widehat{x} \in \operatorname*{argmin}_{x\in\mathbb{R}^N} \chi(x) = f(x) + g(x), \tag{3.19}$$

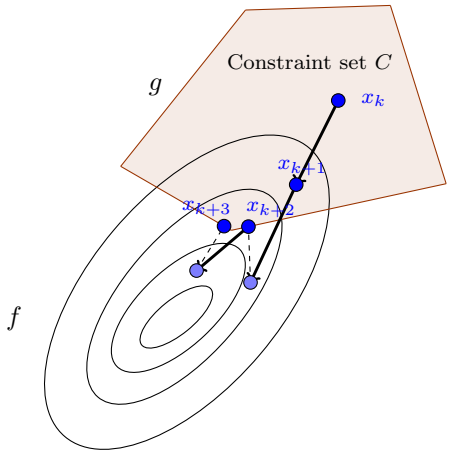


Figure 3.5: *An illustration of few iteration of the forward-backward algorithm when f is a smooth function and g is the indicator function of a convex set. Note that when $x_n - \gamma_n \nabla f(x_n)$ belongs to C, the iteration reduces to a simple gradient descent step.*

where $f \in \Gamma_0(\mathbb{R}^N)$ is a differentiable function with a β-Lipschitz gradient, and $g \in \Gamma_0(\mathbb{R}^N)$. It can be shown from [Combettes and Wajs, 2005] that the solution to problem (3.19) satisfies the following fixed point equation:

$$\widehat{x} = \text{prox}_{\gamma g}\left(\widehat{x} - \gamma_n \nabla f(\widehat{x})\right), \tag{3.20}$$

with $\gamma \in]0, +\infty[$. Using the characterization (3.20), the *forward-backward algorithm* is developed. This algorithm comprises a gradient descent step on the differentiable function f *(forward step)*, followed by a proximal step on the nonsmooth and convex function g *(backward step)* [Chen and Rockafellar, 1997], as shown by Algorithm 5.

Algorithm 5 Forward-backward algorithm

Initialization:

Let $x_0 \in \mathbb{R}^N$

For every $n \in \mathbb{N}$, $0 < \gamma_n < 2\beta^{-1}$ and $\lambda_n \in]0, 1]$

for $n = 0, 1, \ldots$ **do**

$\widetilde{x}_n = x_n - \gamma_n \nabla f(x_n)$

$\widetilde{x}_{n+1} = \text{prox}_{\gamma_n g}(\widetilde{x}_n)$

$x_{n+1} = x_n + \lambda_n \left(\widetilde{x}_{n+1} - x_n\right)$

end for

Theorem 3.15 *Let $f \in \Gamma_0(\mathbb{R}^N)$ be a differentiate function with β-Lipschitz gradient, and $g \in \Gamma_0(\mathbb{R}^N)$. Assuming that* argmin $\chi \neq \varnothing$ *and*

- $0 < \inf_{n\in\mathbb{N}} \gamma_n \leqslant \sup_{n\in\mathbb{N}} \gamma_n < 2\beta^{-1}$,
- $0 < \inf_{n\in\mathbb{N}} \lambda_n \leqslant \lambda_n \leqslant 1$,

then, the sequence $(x_n)_{n\in\mathbb{N}}$ *generated by Algorithm 5 converges towards the solution to problem* (3.19).

It should be noted that the authors in [Attouch and Bolte, 2009; Attouch et al., 2011] have proven that the convergence guarantees of Algorithm 5 to a critical point of χ remain valid even when f and g are non-necessary convex functions with $\lambda_n = 1$. In addition, [Cruz and Nghia, 2016; Salzo, 2016] have also investigated the case when ∇f does not fulfill the Lipschitz assumption. Furthermore, in many cases, the proximity operator of g can be non-explicit and thus non accurate. In such instance, numerous variants have been proposed in order to take into account the potential errors and preserve the convergence properties of Algorithm 5. This can be realized by incorporating additional terms accounting for errors [Combettes and Wajs, 2005], or by designing an inexact version with relative errors [Attouch et al., 2011].

3.3.5 Variable metric forward-backward algorithm

As discussed in the previous section, the forward-backward algorithm is well-adapted to minimization problems involving a sum of a differentiable and convex functions. However, despite its low computational cost, it suffers from a slow convergence rate [Chouzenoux et al., 2014]. Thereby, a close attention has been paid to accelerate it. An interesting approach consists in utilizing a variable metric at each iteration of the algorithm thanks to the introduction of preconditioning matrices [Combettes and Vũ, 2014a; Chouzenoux et al., 2014]. Hence, let us consider again the minimization problem (3.19) with f and g satisfying the following assumptions:

Assumption 3.16

1. $f : \mathbb{R}^N \to]-\infty, +\infty]$ *is differentiable with a* β*-Lipschitz gradient and* $\beta > 0$.
2. $g \in \Gamma_0\left(\mathbb{R}^N\right)$ *and its restriction to dom* g *is continuous.*
3. χ *is a coercive function.*

The following algorithm has been proposed to solve problems in the form of (3.19) in an effective way.

Algorithm 6 Variable metric forward-backward algorithm

Initialization:

Let $x_0 \in \mathbb{R}^N$

For every $n \in \mathbb{N}$, $0 < \gamma_n < 2$ and $\lambda_n \in]0, 1]$

for $n = 0, 1, \ldots$ **do**

$\widetilde{x}_n = x_n - \gamma_n A_n^{-1} \nabla f(x_n)$

$\widetilde{x}_{n+1} = \mathrm{prox}_{\gamma_n^{-1} A_n, g}(\widetilde{x}_n)$

$x_{n+1} = x_n + \lambda_n \left(\widetilde{x}_{n+1} - x_n\right)$

end for

Inexact versions of Algorithm 6 have been supplied in order to take into account potential errors in the computation of the gradient of f and the proximity operator of g [Chouzenoux et al., 2014; Bonettini et al., 2016a]. The convergence of Algorithm 6 has been established in [Combettes and Vũ, 2014a; Frankel et al., 2015] under different assumptions on f, g and the metrics $(A_n)_{n\in\mathbb{N}}$, and in [Bonettini et al., 2016b] where an inexact version of Algorithm 6 has been proposed and analyzed based on an Armijo-type line-search along a suitable descent direction. Here, we focus on the result from [Chouzenoux et al., 2014] that relies on the following MM assumption:

Assumption 3.17

1. *For all $n \in \mathbb{N}$, the function $q(\cdot, x_n)$ defined by*

$$\left(\forall x \in \mathbb{R}^N\right) \quad q(x, x_n) = f(x_n) + \langle x - x_n \mid \nabla f(x_n)\rangle + \frac{1}{2}\|x - x_n\|_{A_n}^2, \quad (3.21)$$

is a majorant function of f at x_n.

2. *There exist $(\underline{\nu}, \overline{\nu}) \in]0, +\infty[^2$ such that, for all $n \in \mathbb{N}$: $\underline{\nu}\,\mathrm{Id}_N \leqslant A_n \leqslant \overline{\nu}\,\mathrm{Id}_N$*

Assumption 3.18

1. *There exists $\underline{\lambda} \in]0, +\infty[$ such that, for all $n \in \mathbb{N}$: $\underline{\lambda} \leqslant \lambda_n \leqslant 1$.*

2. *There exist $(\underline{\nu}, \overline{\nu}) \in]0, +\infty[^2$ such that, for all $n \in \mathbb{N}$: $\underline{\nu} \leqslant \gamma_n \lambda_n \leqslant 2 - \overline{\nu}$.*

Furthermore, χ has to satisfy the so-called Kurdyka-Łojasiewicz inequality given in the next definition.

Definition 3.19 A function $\psi : \mathbb{R}^N \to \mathbb{R}$ satisfies the Kurdyka-Łojasiewicz inequality if for every $\xi \in \mathbb{R}$ and for every bounded subset $E \in \mathbb{R}^N$, there exist three constants $\kappa > 0, \zeta > 0$ and $\theta \in [0, 1[$ such that [Lojasiewicz, 1963]

$$\left(\forall t \in \partial\psi(y)\right) \qquad \|t\| \geqslant \kappa |\psi(y) - \xi|^\theta, \quad (3.22)$$

for every $y \in E$ such that $|\psi(y) - \xi| \leqslant \zeta$ (with the convention $0^0 = 0$).

The inexact variable metric forward-backward algorithm is given by Algorithm 7.

Algorithm 7 Inexact variable metric forward-backward algorithm

Initialization:

Let $x_0 \in \operatorname{dom} g$ and $b \in]0, +\infty[$

For every $n \in \mathbb{N}$, $\gamma_n \in]0, +\infty[$ and $\lambda_n \in]0, 1]$

for $n = 0, 1, \ldots$ **do**

Find $\widetilde{x}_n \in \mathbb{R}^N$ and $r_n \in \partial g(\widetilde{x}_n)$ such that

$g(\widetilde{x}_n) + \langle \widetilde{x}_n - x_n \mid \nabla f(x_n) \rangle + \gamma_n^{-1} \|\widetilde{x}_n - x_n\|_{A_n}^2 \leqslant g(x_n)$

$\|\nabla f(x_n) + r_n\| \leqslant b \|\widetilde{x}_n - x_n\|_{A_n}$

$x_{n+1} = x_n + \lambda_n (\widetilde{x}_n - x_n)$

end for

Theorem 3.20 *Assuming that Assumptions 3.16, 3.17 and 3.18 are fulfilled and χ satisfies the Kurdyka-Łojasiewicz inequality, then*

1. *The sequences $(x_{n+1})_{n\in\mathbb{N}}$ and $(\widetilde{x}_{n+1})_{n\in\mathbb{N}}$ generated by Algorithms 6 or 7 converge to a critical point $\widehat{x}$ of χ.*
2. *For all $n \in \mathbb{N}$, the sequences $\chi(x_{n+1})$ and $\chi(\widetilde{x}_{n+1})$ converge toward $\chi(\widehat{x})$. Moreover, $\chi(x_{n+1})$ is a non-increasing sequence.*
3. *The following inequalities are satisfied*

$$\sum_{n=0}^{+\infty} \|x_{n+1} - x_n\| < +\infty \quad \text{and} \quad \sum_{n=0}^{+\infty} \|\widetilde{x}_{n+1} - \widetilde{x}_n\| < +\infty. \tag{3.23}$$

3.3.6 Block-coordinate variable metric forward-backward algorithm

We will consider in this section a special instance of Problem (3.19) dealing with the case when g is a separable function. We assume that the vector $x \in \mathbb{R}^N$ can be decomposed into J blocks $(x^j)_{1\leqslant j\leqslant J}$ where J is the cardinal number of the set of partitions of $\{1, \ldots, N\}$ and $(x^j)_{1\leqslant j\leqslant J} \in \mathbb{R}^{N_1} \times \ldots \times \mathbb{R}^{N_J}$, i.e.,

$$\big(\forall x \in \mathbb{R}^N\big) \quad g(x) = \sum_{j=1}^{J} g_j(x^j), \tag{3.24}$$

The following algorithm generalizing Algorithm 6, has been proposed in [Chouzenoux et al., 2016] to handle efficiently minimization problems of this form, where at each

iteration $n \in \mathbb{N}$, only a block of index $j \in \{1, \dots, J\}$ is selected and its associated variable x_j is updated.

Algorithm 8 Block-coordinate variable metric forward-backward algorithm

Initialization:

Let $x_0 \in \mathbb{R}^N$

For every $n \in \mathbb{N}$, $\gamma_n \in]0, +\infty[$

for $n = 0, 1, \dots$ **do**

Let $j_n \in \{1, \dots, J\}$

$\widetilde{x}_n^{j_n} = x_n^{j_n} - \gamma_n (A_n^{j_n})^{-1} \nabla_{j_n} f(x_n)$

$\widetilde{x}_{n+1}^{j_n} = \operatorname{prox}_{\gamma_n^{-1} A_n^{j_n}, g_{j_n}}(\widetilde{x}_n^{j_n})$

$x_{n+1}^{\overline{j}} = x_n^{\overline{j}}$

end for

For every $x \in \mathbb{R}^N$ and $j \in \{1, \dots J\}, \nabla_j f(x) \in \mathbb{R}^{N_j}$ denotes the gradient of f with respect to x^j computed at x, $\overline{j}$ denotes the complementary set of j i.e., $\overline{j} = \{1, \dots, J\} \backslash \{j\}$, and $x^{\overline{j}} = (x^1, \dots, x^{j-1}, x^{j+1}, \dots, x^J)$. The convergence is established assuming that f and $(g_j)_{1 \leqslant j \leqslant J}$ fulfill the following assumptions:

Assumption 3.21

1. *$f : \mathbb{R}^N \to]-\infty, +\infty]$ is differentiable with a β-Lipschitz gradient and $\beta > 0$.*
2. *$\forall j \in \{1, \dots, J\}$ $g_j : \mathbb{R}^{N_j} \to]-\infty, +\infty]$ is proper, lower semicontinuous, bounded from below by an affine function and its restriction to its domain is continuous.*
3. *χ is a coercive function.*

Assumption 3.22

1. *Let us define the partial function $f_j(\cdot, x^{\overline{j}}) : \mathbb{R}^{N_j} \to \mathbb{R}$ as follows*

$$(\forall y \in \mathbb{R}^{N_j}) \quad f_j(y, x^{\overline{j}}) = (x^1, \dots, x^{j-1}, y, x^{j+1}, \dots, x^J). \tag{3.25}$$

For every $n \in \mathbb{N}$, the function $q_{j_n}(\cdot, x_n)$ defined by

$$\big(\forall x \in \mathbb{R}_j^N\big) \quad q_{j_n}(x, x_n) = f(x_n) + \langle x - x_n^{j_n} \mid \nabla_{j_n} f(x_n) \rangle + \frac{1}{2} \|x - x_n^{j_n}\|_{A_n^{j_n}}^2, \tag{3.26}$$

is a majorant function of $f_{j_n}(\cdot, x^{\overline{j}_n})$ at $x_n^{j_n}$.

2. *There exist $(\underline{\nu}, \overline{\nu}) \in]0, +\infty[^2$ such that, for all $n \in \mathbb{N}$: $\underline{\nu} \operatorname{Id}_{N_j} \leqslant A_n^{j_n} \leqslant \overline{\nu} \operatorname{Id}_{N_j}$*

Assumption 3.23 *Let $(j_n)_{n\in\mathbb{N}}$ be the sequence of updated block indices. There exists a constant $K \geqslant J$ such that, for every $n \in \mathbb{N}, \{1, \dots J\} \subset \{j_n, \dots, j_{n+K-1}\}$.*

Assumption 3.24 *One of the following statements holds:*

1. *There exists $(\underline{\gamma}, \overline{\gamma}) \in]0, +\infty[^2$ such that, for every $n \in \mathbb{N}, \underline{\gamma} \leqslant \gamma_n \leqslant 1 - \overline{\gamma}$*
2. *For every $j \in \{1, \dots, J\}$ g_j is a convex function and there exist $(\underline{\gamma}, \overline{\gamma}) \in]0, +\infty[^2$ such that for every $n \in \mathbb{N}, \underline{\gamma} \leqslant \gamma_n \leqslant 2 - \overline{\gamma}$*

Theorem 3.25 *Assuming that Assumptions 3.21, 3.22, 3.23 and 3.24 are fulfilled and χ satisfies the Kurdyka-Łojasiewicz inequality, then*

1. *The sequences $(x_n)_{n\in\mathbb{N}}$ generated by Algorithm 8 converges to a critical point $\widehat{x}$ of χ.*
2. *For all $n \in \mathbb{N}$, the sequences $\chi(x_n)$ is a non-increasing sequence converging to $\chi(\widehat{x})$.*
3. *This sequence has a finite length in the sense that*

$$\sum_{n=0}^{+\infty} \|x_{n+1} - x_n\| < +\infty. \tag{3.27}$$

3.3.7 Primal-dual algorithms

Let us now consider minimization problems involving nonsmooth terms composed with linear operators. This can be encountered for example when dealing with an image in the wavelet domain, or when penalizing its gradient. Hence in the following, we focus on minimization problems of the form:

$$\text{Find } \widehat{x} \in \operatorname*{argmin}_{x\in\mathbb{R}^N} f(x) + g(x) + h(Ax), \tag{3.28}$$

where $f \in \Gamma_0(\mathbb{R}^N)$ is a differentiable function with a β-Lipschitz gradient, $g \in \Gamma_0(\mathbb{R}^N)$, $h \in \Gamma_0(\mathbb{R}^M)$ and $A \in \mathbb{R}^{M\times N}$ is a linear operator. A solution to problem (3.28) can be obtained by solving both the *primal problem* and its associated *dual problem* [Komodakis and Pesquet, 2015; Bonettini et al., 2014; Bednarczuk et al., 2016], that is expressed by

$$\text{Find } \widehat{y} \in \operatorname*{argmin}_{y\in\mathbb{R}^M} (f^* \,\square\, g^*)(-A^\top y) + h^*(y). \tag{3.29}$$

If the pair $(\widehat{x}, \widehat{y}) \in \mathbb{R}^N \times \mathbb{R}^M$ is such that

$$-A^\top \widehat{y} - \nabla f(\widehat{x}) \in \partial g(\widehat{x}) \quad \text{and} \quad A\widehat{x} \in \partial h^*(\widehat{y}), \tag{3.30}$$

then $(\widehat{x}, \widehat{y})$ is referred to as *Kuhn-Tucker point*.

The combination of problems (3.28) and (3.29) yields the so-called search of a saddle point of the Lagrangian problem [Bauschke and Combettes, 2017] given by

$$\underset{x\in\mathbb{R}^N}{\text{minimize}} \quad \underset{y\in\text{dom}h^*}{\text{maximize}} \quad f(x) + g(x) - h^*(y) + \langle Ax \mid y\rangle. \tag{3.31}$$

If $(\widehat{x}, \widehat{y}) \in \mathbb{R}^N \times \mathbb{R}^M$ is a Kuhn-Tucker point, then $\widehat{x}$ is a solution to the primal problem (3.28), $\widehat{y}$ is a solution to the dual problem (3.29), and $(\widehat{x}, \widehat{y})$ is a solution to (3.31) [Bauschke and Combettes, 2017]. However, the converse does not necessary holds, i.e., the Kuhn-Tucker condition (3.30) may not be satisfied even though $\widehat{x}$ and $\widehat{y}$ are solutions to problems (3.28) and (3.29) respectively [Bauschke and Combettes, 2017; Komodakis and Pesquet, 2015].
In addition, if the following condition is fulfilled

$$\text{ri}(\text{dom}\, h) \cap A\,(\text{dom}\, g) \neq \varnothing, \tag{3.32}$$

where $\text{ri}(S)$ is the relative interior of a set S, then the set of solutions to problem (3.29) is nonempty and, for every solution to primal and dual problems $\widehat{x}$ and $\widehat{y}$ respectively, the pair $(\widehat{x}, \widehat{y})$ is a solution to problem (3.31), and the following equality is satisfied

$$f(\widehat{x}) + g(\widehat{x}) + h(A\widehat{x}) = -\left((f^* \,\square\, g^*)(-A^\top \widehat{y}) + h^*(\widehat{y})\right). \tag{3.33}$$

The class of algorithms that minimize both primal and dual problems are termed *primal-dual algorithms*. Among them, we mention the one proposed by Condat-Vũ [Condat, 2013; Vũ, 2013] that is given by Algorithm 9.

Algorithm 9 Primal-dual algorithm [Condat, 2013; Vũ, 2013]

Initialization:

Let $(x_0, y_0) \in \mathbb{R}^N \times \mathbb{R}^M$ and $(\sigma, \tau) \in]0, +\infty[^2$

For every $n \in \mathbb{N}$, $\lambda_n \in]0, +\infty[$

for $n = 0, 1, \ldots$ **do**

$\widetilde{x}_{n+1} = \text{prox}_{\tau g}\left(x_n - \tau(\nabla f(x_n) + b_n + A^\top y_n)\right) + a_n$

$\widetilde{y}_{n+1} = \text{prox}_{\sigma h^*}(y_n + A(2\widetilde{x}_{n+1} - x_n)) + c_n$

$x_{n+1} = \lambda_n \widetilde{x}_{n+1} + (1 - \lambda_n) x_n$

$y_{n+1} = \lambda_n \widetilde{y}_{n+1} + (1 - \lambda_n) y_n$

end for

The convergence of Algorithm 9 is stated by the following theorem:

Theorem 3.26 *[Condat, 2013] Let $\tau > 0$ and $\sigma > 0$ and $\beta > 0$ be the Lipschitz constant of ∇f. Assuming that the following hold*

1. $\tau^{-1} - \sigma\|A\|^2 \geqslant \beta/2$
2. $\forall n \in \mathbb{N}, \lambda_n \in]0, \overline{\lambda}[$ *where* $\overline{\lambda} = 2 - \beta\left(\tau^{-1} - \sigma\|A\|^2\right)^{-1}/2 \in [1, 2[$
3. $\sum_{n\in\mathbb{N}} \lambda_n(\overline{\lambda} - \lambda_n) = +\infty$
4. $\sum_{n\in\mathbb{N}} \lambda_n\|a_n\| < +\infty$, $\sum_{n\in\mathbb{N}} \lambda_n\|b_n\| < +\infty$ *and* $\sum_{n\in\mathbb{N}} \lambda_n\|c_n\| < +\infty$

then, there exists a pair $(\widehat{x}, \widehat{y}) \in \mathbb{R}^N \times \mathbb{R}^M$ solution to problems (3.28)-(3.29) such that the sequences $(x_n)_{n\in\mathbb{N}}$ and $(y_n)_{n\in\mathbb{N}}$ generated by Algorithm 9 converge to $\widehat{x}$ and $\widehat{y}$ respectively.

Note that when $f = 0$, one recovers the primal-dual algorithm of [Chambolle and Pock, 2010; Pock et al., 2009], whose convergence conditions are less restrictive than those of Algorithm 9.

3.3.8 Dual forward-backward algorithm

As we have observed in this chapter, the proximity operator is a prominent tool when treating optimization problems including nonsmooth terms. However, the evaluation of proximity operators can be quite involved and a closed form expression may not exist. In this section, we address the problem of computing the proximity operator of a sum of two possibly nonsmooth functions involving a linear operator, by adopting a primal-dual approach. First, let us define

$$\chi(x) = g(x) + h(Ax). \tag{3.34}$$

where $g \in \Gamma_0(\mathbb{R}^N)$, $h \in \Gamma_0(\mathbb{R}^M)$ and A is a linear operator in $\mathbb{R}^{M\times N}$. We aim at evaluating the proximity operator of χ at $\widetilde{x} \in \mathbb{R}^N$ in the metric induced by C, i.e.,

$$\begin{aligned} \text{Find } \widehat{x} &= \operatorname{prox}_{C,\chi}(\widetilde{x}), \\ &= \operatorname*{argmin}_{x\in\mathbb{R}^N} \; g(x) + h(Ax) + \frac{1}{2}\|x - \widetilde{x}\|_C^2, \end{aligned} \tag{3.35}$$

where $C \in \mathbb{R}^{N\times N}$ is a symmetric strictly positive definite matrix.
Assuming that $\operatorname{ri}\big(A(\operatorname{dom} g)\big) \cap \operatorname{ri}(\operatorname{dom} h) \neq \varnothing$, the dual problem is expressed by

$$\text{Find } \widehat{y} = \operatorname*{argmin}_{y\in\mathbb{R}^M} \; \varphi\left(-C^{-1/2}A^\top y + C^{1/2}\widetilde{x}\right) + h^*(y), \tag{3.36}$$

where $\varphi = \left(\left(g \circ C^{-1/2}\right)^* \,\square\, \frac{1}{2}\|\,.\,\|^2\right)$ has a nonexpansive (i.e., 1-Lipschitzian) gradient. The gradient of the smooth part of (3.36) can be expressed by means of g as

follows

$$\begin{aligned}\nabla\varphi \circ \left(-C^{-1/2}A^\top \cdot + C^{1/2}\widetilde{x}\right) &= -AC^{-1/2}\nabla\varphi \circ \left(-C^{-1/2}A^\top \cdot + C^{1/2}\widetilde{x}\right), \\ &= -AC^{-1/2}\,\mathrm{prox}_{g\circ C^{-1/2}}\left(C^{1/2}\widetilde{x} - C^{-1/2}A^\top \cdot\right), \\ &= -A\,\mathrm{prox}_{C,g}\left(\widetilde{x} - C^{-1}A^\top \cdot\right), \end{aligned} \tag{3.37}$$

A solution to solve problem (3.35) is to apply the forward-backward Algorithm 5 to the dual problem (3.36). By setting $x_n = \mathrm{prox}_{C,g}\left(\widetilde{x} - C^{-1}A^\top y_n\right)$ for all $n \in \mathbb{N}$, the following algorithm is obtained

Algorithm 10 Dual forward-backward algorithm

Initialization:

Let $y_0 \in \mathbb{R}^M$, $\beta = \|AC^{-1/2}\|^2$ and $\varepsilon \in]0, 1]$

For every $n \in \mathbb{N}$, $\gamma_n \in [\varepsilon\beta^{-1}, (2-\varepsilon)\beta^{-1}]$

for $n = 0, 1, \ldots$ **do**

$x_n = \mathrm{prox}_{C,g}(\widetilde{x} - C^{-1}A^\top y_n)$

$\widetilde{y}_n = y_n + \gamma_n A x_n$

$y_{n+1} = \widetilde{y}_n - \gamma_n \mathrm{prox}_{\gamma_n^{-1}h}(\gamma_n^{-1}\widetilde{y}_n)$

end for

The convergence of Algorithm 10 is established by Theorem 3.27, deduced from [Combettes et al., 2011].

Theorem 3.27 *The sequences $(x_n)_{n\in\mathbb{N}}, (y_n)_{n\in\mathbb{N}}$ generated by Algorithm 10 converge to the solution to Problems* (3.35)-(3.36)*, $\widehat{x}$ and $\widehat{y}$ respectively, and the pair $(\widehat{x}, \widehat{y})$ satisfies the Kuhn-Tucker condition* (3.30)*.*

In short, Algorithm 10 results from the application of the forward-backward algorithm to the dual problem (3.36). In the next chapter, we propose an accelerated version of this approach for the problem of computing the proximity operator of χ when h is a separable function.

3.4 Conclusion

We have highlighted in this chapter a number of optimization algorithms that are developed according to the mathematical properties of the functions which are dealt with. Some of these algorithms are applicable on differentiable functions, while others on non differentiable ones. Most of the mentioned algorithms resort to the

proximity operator. Hence, a particular attention is paid in this thesis to provide new efficient tools for evaluating it when an explicit form is not available. In this spirit, the next chapter addresses the problem of computing the proximity operator of a composite function by providing new algorithms with established convergence guarantees.

- Chapter 4 -

Dual block-coordinate forward-fackward algorithm

Contents

4.1 Introduction

This chapter is dedicated to the proposal of new primal-dual algorithms based on the dual forward-backward algorithm. These algorithms are combined with preconditioning and block coordinate strategies so as to get an improved performance. The proposed methods are used for computing proximity operators when they do not have a closed form expression, especially when dealing with a sum of several convex functions involving linear operators. In addition, we show that our algorithms benefit from convergence guaranties on both primal and dual sequences for arbitrary linear operators.

The remainder of the chapter is structured as follows: we begin by presenting in Section 4.2 a preconditioned algorithm for the non-separable case. Then, we derive in Section 4.3, new fast algorithms for the separable case by introducing a block-coordinate strategy, and investigate their convergence properties in Section 4.4. The application of the proposed algorithms to deconvolution and super-resolution of interlaced video sequences will be addressed in Chapter 5.

4.2 Non-separable case

4.2.1 Problem statement

Similarly to the work in [Combettes et al., 2011], let us focus on the computation of the proximity operator of a function F at $\widetilde{x} \in \mathbb{R}^N$, where F is defined as

$$\big(\forall x \in \mathbb{R}^N\big) \quad F(x) = f(x) + g(Ax), \tag{4.1}$$

with $f \in \Gamma_0(\mathbb{R}^N)$, $g \in \Gamma_0(\mathbb{R}^M)$ and $A \in \mathbb{R}^{M \times N}$ is a linear operator. The problem is thus equivalent to:

$$\begin{aligned} \text{Find } \widehat{x} &= \operatorname{prox}_F(\widetilde{x}), \\ &= \underset{x \in \mathbb{R}^N}{\operatorname{argmin}} \; f(x) + g(Ax) + \frac{1}{2}\|x - \widetilde{x}\|^2. \end{aligned} \tag{4.2}$$

Problem (4.2) has attracted a large interest and has been widely investigated in the literature via deterministic and stochastic approaches. Among deterministic methods, one can mention the dual parallel algorithm in [Combettes et al., 2011], that converges strongly to the sought proximity operator, in the case of convex proper lower-semicontinuous functions composed with bounded linear operators. Its particular case is the Dykstra-like algorithm [Bauschke and Combettes, 2008] when the problem reduces to evaluating the proximity operator of a sum of two convex functions. It is also worth mentioning the work in [Fu et al., 2014] which proposes a parallel splitting version of the alternating direction method of multipliers [Boyd

et al., 2011].
An appealing idea in the context of optimization is to adopt a block coordinate strategy [Chouzenoux et al., 2016], in such a way that two successive iterations deal with different blocks of variables. The block selection rule can be either deterministic (e.g., cyclic, quasi-cyclic, greedy) [Richtárik and Takác, 2011; Saha and Tewari, 2013] or random [Jaggi et al., 2014; Combettes and Pesquet, 2015; Pesquet and Repetti, 2015; Onose et al., 2016]. Based on this idea, various stochastic algorithms have been initially proposed in machine learning area, usually known as *dual ascent algorithms*. One can mention the stochastic dual coordinate ascent algorithm [Shalev-Shwartz and Zhang, 2013] where the functions are assumed to be Lipschitz continuous or smooth with a Lipschitz gradient, and its variant [Qu et al., 2015] where the selection rule of the blocks is arbitrary and the smoothness of the objective function is required. Another stochastic algorithm is the communication efficient distributed dual coordinate ascent algorithm [Jaggi et al., 2014] which has been designed in order to distribute block processing over multiple cores or remote machines. Nevertheless, the convergence guaranties shown for these dual ascent algorithms only concern decay properties on the dual of the objective function, the variables being assumed to be scalar. In the deterministic case, an accelerated FISTA-like method is proposed in [Chambolle and Pock, 2015], where the authors investigate a similar problem. They provide convergence guaranties for primal iterates as well, when each involved function deals with a variable belonging to $\mathbb{R}^2$.

4.2.2 Preconditioned dual forward-backward

As seen in the previous chapter, a solution to problem (4.2) can be obtained using Algorithm 10, that relies on its dual formulation given by

$$\text{Find } \widehat{y} = \operatorname*{argmin}_{y \in \mathbb{R}^M} \ \varphi(-A^\top y + \widetilde{x}) + g^*(y), \tag{4.3}$$

where $\varphi = f^* \,\square\, \frac{1}{2} \parallel . \parallel^2$ as defined in Section 3.3.8 with $C = \mathrm{Id}_N$, and assuming that:

Assumption 4.1 $\mathrm{ri}\big(A(\mathrm{dom}\, f)\big) \cap \mathrm{ri}(\mathrm{dom}\, g) \neq \varnothing$.

We propose a preconditioned version of Algorithm 10 by applying Algorithm 6 to the dual problem (4.3). This consists in introducing a preconditioning matrix B in the gradient step on the smooth function φ, and the proximal step on the convex function g^* as follows

$$\begin{aligned} &\text{For } n = 0, 1, \ldots \\ &\left\lfloor \begin{aligned} &\widetilde{y}_n = y_n - \gamma_n B^{-1} \nabla\big(\varphi \circ (-A^\top \cdot + \widetilde{x})\big)(y_n) \\ &y_{n+1} = \mathrm{prox}_{\gamma_n^{-1} B, g^*}(\widetilde{y}_n) \end{aligned} \right. \end{aligned} \tag{4.4}$$

where $y_0 \in \mathbb{R}^M$, $B \in \mathbb{R}^{M\times M}$ is a symmetric positive definite matrix with $B \succeq AA^\top$ and

$$(\forall n \in \mathbb{N}) \qquad \gamma_n \in [\epsilon, 2-\epsilon] \quad \text{with } \epsilon \in]0,1]. \tag{4.5}$$

By setting

$$(\forall n \in \mathbb{N}) \qquad x_n = \mathrm{prox}_f(\widetilde{x} - A^\top y_n), \tag{4.6}$$

the gradient step in (4.4) can be formulated by means of x_n using (4.6), whereas, the proximity operator of g^* can be rewritten in terms of g thanks to Moreau decomposition formula (3.11). This leads to the following algorithm:

Algorithm 11 Preconditioned dual forward-backward algorithm

Initialization:

Let $y_0 \in \mathbb{R}^M$

For every $n \in \mathbb{N}, \gamma_n \in]0, +\infty[$

for $n = 0, 1, \ldots$ **do**

$x_n = \mathrm{prox}_f(\widetilde{x} - A^\top y_n)$

$\widetilde{y}_n = y_n + \gamma_n B^{-1} A x_n$

$y_{n+1} = \widetilde{y}_n - \gamma_n B^{-1} \mathrm{prox}_{\gamma_n B^{-1}, g}(\gamma_n^{-1} B \widetilde{y}_n)$

end for

It can be shown that the sequences $(x_n)_{n\in\mathbb{N}}$ and $(y_n)_{n\in\mathbb{N}}$ generated by Algorithm 11 converge to the solutions to the primal and dual problems $\widehat{x}$ and $\widehat{y}$ respectively [Combettes et al., 2010]. Moreover the following relation is satisfied

$$\widehat{x} = \mathrm{prox}_f(\widetilde{x} - A^\top \widehat{y}). \tag{4.7}$$

4.2.3 Link with Dykstra algorithm

Let us introduce the variables

$$(\forall n \in \mathbb{N}) \quad p_n = \mathrm{prox}_{\gamma_n B^{-1}, g}(\gamma_n^{-1} B \widetilde{y}_n), \tag{4.8}$$

$$\begin{aligned} q_n &= -A^\top y_{n+1} + \widetilde{x} - \gamma_n A^\top B^{-1} p_n, \\ &= \widetilde{x} - A^\top(\gamma_n B^{-1} A x_n + y_n), \end{aligned} \tag{4.9}$$

and let us notice that, if $\gamma_n \equiv \gamma$ satisfies (4.5) , then the following recursive relation is fulfilled:

$$q_{n+1} = q_n + \gamma A^\top B^{-1}(p_n - A x_{n+1}). \tag{4.10}$$

Algorithm 11 can then be rewritten as

Algorithm 12 Dykstra-like formulation of Algorithm 11

Initialization:

Let $y_0 \in \mathbb{R}^M$, $x_0 = \mathrm{prox}_f(\widetilde{x} - A^\top y_0)$ and $q_0 = \widetilde{x} - A^\top(\gamma B^{-1} A x_0 + y_0)$

For every $n \in \mathbb{N}, \gamma \in]0, +\infty[$

for $n = 0, 1, \ldots$ **do**

$p_n = \mathrm{prox}_{\gamma B^{-1}, g}(\gamma^{-1} B y_n + A x_n)$

$y_{n+1} = y_n + \gamma B^{-1}(A x_n - p_n)$

$x_{n+1} = \mathrm{prox}_f(q_n + \gamma A^\top B^{-1} p_n)$

$q_{n+1} = q_n + \gamma A^\top B^{-1}(p_n - A x_{n+1})$

end for

In particular, as pointed out in [Combettes et al., 2010], if $A = \mathrm{Id}_N$ and $\gamma B^{-1} = \mathrm{Id}_M$ we retrieve the same iterative structure as the Dykstra-like algorithm which was proposed in [Bauschke and Combettes, 2008] and whose convergence was proved for another initialization strategy.

4.3 Separable case

4.3.1 Problem statement

Algorithm 11 exhibits some limitations in the case when g in (4.1) is a separable function, since one has to deal with the full linear operator A at each iteration, which may be very costly when the size of A is large.
Now, we will derive new algorithms based on Algorithm 11 when g is a separable function, namely:

$$(\forall x \in \mathbb{R}^N) \qquad g(Ax) = \sum_{j=1}^{J} g_j(A_j x), \tag{4.11}$$

where, for every $j \in \{1, \ldots, J\}$, A_j is a non null matrix in $\mathbb{R}^{M_j \times N}$ with $\sum_{j=1}^{J} M_j = M$, $g_j \in \Gamma_0(\mathbb{R}^{M_j})$, and

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_J \end{bmatrix}. \tag{4.12}$$

Then, problem (4.2) becomes:

$$\begin{aligned} \text{Find } \widehat{x} &= \operatorname{prox}_F(\widetilde{x}), \\ &= \operatorname*{argmin}_{x\in\mathbb{R}^N} f(x) + \sum_{j=1}^{J} g_j(A_j x) + \frac{1}{2}\|x-\widetilde{x}\|^2. \end{aligned} \tag{4.13}$$

According to (4.12), the dual problem reads:

$$\text{Find } \widehat{y} = \operatorname*{argmin}_{y=(y^j)_{1\leqslant j\leqslant J}\in\mathbb{R}^M} \varphi\Big(\widetilde{x} - \sum_{j=1}^{J} A_j^\top y^j\Big) + \sum_{j=1}^{J} g_j^*(y^j). \tag{4.14}$$

Note that the dual variable y is now decomposed into J blocks of variables $(y^j)_{1\leqslant j\leqslant J}$.

4.3.2 Dual block forward-backward algorithm

The application of the variable metric block-coordinate forward-backward Algorithm 8 to the dual problem (6.4) yields the new Algorithm 13 where, at each iteration $n\in\mathbb{N}$, a block of index j_n is activated and its associated dual variable $y_n^{j_n}$ is updated by performing a proximal step on the function g_{j_n}, in the metric induced by a preconditioning matrix B_{j_n} satisfying (4.15). Note that the dual variable $y_n^{j_n}$ is the only one to be processed at the n-th iteration, whereas the other dual variables of index $j\neq j_n$ are kept intact during this iteration.

Algorithm 13 Dual block forward-backward algorithm

Initialization:

Let $(y_0^j)_{1\leqslant j\leqslant J}\in\mathbb{R}^M$

For every $n\in\mathbb{N}, \gamma_n\in]0,+\infty[$

for $n=0,1,\ldots$ **do**

$x_n = \operatorname{prox}_f(\widetilde{x} - A^\top y_n)$

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

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

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

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

end for

The step size $(\gamma_n)_{n\in\mathbb{N}}$ fulfils (4.5), and

$$(\forall j\in\{1,\ldots,J\}) \quad B_j \succ O_{M_j} \text{ with } B_j \succeq A_j A_j^\top. \tag{4.15}$$

The simplest (non-preconditioned) version of Algorithm 13 is obtained by choosing

$$(\forall j \in \{1, \dots, J\}) \qquad B_j = \beta_j \mathrm{Id}_{M_j}, \tag{4.16}$$

where β_j is the squared spectral norm of the associated linear operator A_j, i.e., $\beta_j = \|A_j\|^2$.

Remark 4.2

1. The pair $\left(\widehat{x}, (\widehat{y}^j)_{1 \leqslant j \leqslant J}\right)$ is a solution to the primal and dual problems if and only if

$$\begin{cases} -\sum_{j=1}^{J} A_j^\top \widehat{y}^j \in \partial f(\widehat{x}) + \widehat{x} - \widetilde{x} \\ \qquad\qquad \Leftrightarrow \ \widehat{x} = \mathrm{prox}_f\Big(\widetilde{x} - \sum_{j=1}^{J} A_j^\top \widehat{y}^j\Big), & (4.17) \\ (\forall j \in \{1, \dots, J\}) \qquad \widehat{y}^j \in \partial g_j(A_j \widehat{x}). & (4.18) \end{cases}$$

   When the functions $(g_j)_{1 \leqslant j \leqslant J}$ are differentiable, the second optimality condition can be used to define a dual residue, which is exploited for the blocks selection rule in some recent dual coordinate ascent strategies [Csiba et al., 2015].

2. If relation (4.16) is satisfied, $f = \theta \|\cdot\|_1$ with $\theta \in (0, +\infty)$, and $(\forall j \in \{1, \dots, J\})$ $g_j = \iota_{\{b^j\}}$ with $b^j \in \mathbb{R}^{M_j}$, we recover an algorithm similar to the one studied in [Lorenz et al., 2014].

4.3.3 Simplified form

In Algorithm 13, the update of the primal variable x_n involves all the dual variables $(y_n^{j_n})_{1 \leqslant j \leqslant J}$ and the whole matrix $A^\top$. This is clearly suboptimal since only one block j_n is being processed at iteration $n \in \mathbb{N}$. To overcome this shortcoming, we introduce a new variable $(z_n)_{n \in \mathbb{N}} \in \mathbb{R}^N$ that takes into account only the updated dual variable.
To do so, let us define $(z_n)_{n \in \mathbb{N}}$ such that

$$z_n = -A^\top y_n = -\sum_{i=1}^{J} A_i^\top y_n^i. \tag{4.19}$$

Then we have

$$\begin{aligned} z_{n+1} &= -\sum_{i=1}^{J} A_i^\top y_{n+1}^i = -\sum_{\substack{i=1 \\ i\neq j_n}}^{J} A_i^\top y_{n+1}^i - A_{j_n}^\top y_{n+1}^{j_n}, \\ &= -\sum_{i=1}^{J} A_i^\top y_n^i - A_{j_n}^\top y_{n+1}^{j_n} + A_{j_n}^\top y_n^{j_n}, \\ &= z_n - A_{j_n}^\top (y_{n+1}^{j_n} - y_n^{j_n}). \end{aligned} \tag{4.20}$$

Hence, Algorithm 13 becomes:

Algorithm 14 Simplified dual block forward-backward algorithm

Initialization:

Let $(y_0^j)_{1\leqslant j\leqslant J} \in \mathbb{R}^M$

For every $n \in \mathbb{N}, \gamma_n \in]0, +\infty[$

for $n = 0, 1, \ldots$ **do**

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

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

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

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

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

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

end for

with the initialization

$$z_0 = -\sum_{j=1}^{J} A_j^\top y_0^j. \tag{4.21}$$

This simplified form of Algorithm 13 is more efficient in the sense that the updating steps of both primal and dual variables involves only the selected block j_n, thereby, the simplified version reduces the complexity of the algorithm and its memory requirements.

4.3.4 Particular case when $f = 0$

A special interesting case is obtained when f is the null function. Then, the update of the primal variable in Algorithm 14 reduces to

$$x_n = \widetilde{x} + z_n. \tag{4.22}$$

Thus, by changing the initialization (4.21) to $x_0 = \widetilde{x} - \sum_{j=1}^{J} A_j^\top y_0^j$ and after some simplifications, Algorithm 14 becomes:

Algorithm 15 Dual block forward-backward algorithm when $f = 0$

Initialization:

Let $(y_0^j)_{1 \leqslant j \leqslant J} \in \mathbb{R}^M$

For every $n \in \mathbb{N}, \gamma_n \in]0, +\infty[$

for $n = 0, 1, \ldots$ **do**

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

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

$y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n B_{j_n}^{-1} \mathrm{prox}_{\gamma_n B_{j_n}^{-1}, g_{j_n}} \left(\gamma_n^{-1} B_{j_n} \widetilde{y}_n^{j_n}\right)$

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

$x_{n+1} = x_n - A_{j_n}^\top (y_{n+1}^{j_n} - y_n^{j_n})$

end for

4.3.5 Link with the parallel dual forward-backward

Algorithm 14 can be compared with its parallel variant proposed in [Combettes and Vũ, 2014b, Example 5.6] given by:

Algorithm 16 Parallel dual block forward-backward algorithm [Combettes and Vũ, 2014b]

Initialization:

Let $(y_0^j)_{1 \leqslant j \leqslant J} \in \mathbb{R}^M$

For every $n \in \mathbb{N}, \gamma_n \in]0, +\infty[$

for $n = 0, 1, \ldots$ **do**

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

for $j = 1, \ldots, J$ **do**

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

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

end for

$z_{n+1} = z_n - \sum_{j=1}^{J} A_j^\top (y_{n+1}^{j} - y_n^{j})$

end for

where z_0 is defined according to (4.21) and the preconditioning matrices $(B_j)_{1 \leqslant j \leqslant J}$

are such that

$$\forall j \in \{1, \ldots, J\} \quad B_j \succeq \beta \mathrm{Id}_{M_j}, \tag{4.23}$$

with $\beta = \sum_{j=1}^{J} \|A_j\|^2$.

Some similarities existing between Algorithms 14 and 16 can be observed. However, in Algorithm 16, the dual variables $(y_n^j)_{1 \leqslant j \leqslant J}$ are updated in parallel and the update of x_n has to be performed from all these dual variables. Conversely, in Algorithm 14, the dual variables are updated sequentially, and after any update of each of them, the primal variable is also updated. When no parallel implementation is used, this second solution can be expected to be more efficient with a faster convergence speed.
Moreover, it should be noted that conditions (4.23) imposed on the matrices $(B_j)_{1 \leqslant j \leqslant J}$ in Algorithm 16 appear to be more restrictive than those imposed in Algorithm 14 (see conditions (4.15)). Since the preconditioning matrices $(B_j)_{1 \leqslant j \leqslant J}$ usually play an important role in the convergence speed, more freedom in their choice should also be beneficial to the algorithm performance.

A variant of the above parallel algorithm dealing with the case when $f = 0$ can be derived from the parallel block forward-backward algorithm proposed in [Combettes et al., 2011] which, in the absence of error terms and relaxation factor, reads:

Algorithm 17 Parallel dual block forward-backward algorithm when $f = 0$ [Combettes et al., 2011]

Initialization:

Let $(y_0^j)_{1 \leqslant j \leqslant J} \in \mathbb{R}^M$

For every $n \in \mathbb{N}, \gamma_n \in]0, +\infty[$

for $n = 0, 1, \ldots$ **do**

  for $j = 1, \ldots, J$ **do**

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

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

  end for

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

end for

where

$$\forall j \in \{1, \ldots, J\}\ B_j = \beta \omega_j^{-1} \mathrm{Id}_{M_j}, \text{ with } \beta = \max_{j \in \{1, \ldots, J\}} \|A_j\|^2,$$

$$\text{and } (\omega_j)_{1\leqslant j\leqslant J} \in]0,1]^J \text{ are such that } \sum_{j=1}^{J} \omega_j = 1.$$

Algorithms 15 and 17 exhibit several similarities, however, as mentioned hereabove, the main difference lies in the update rule of the dual variables. Another advantage of Algorithm 15 is that it leads to less restrictive conditions on the matrices $(B_j)_{1\leqslant j\leqslant J}$. Indeed, for Algorithm 17, we have

$$(\forall j \in \{1,\dots,J\}) \quad B_j \succeq \omega_j B_j = \beta \mathrm{Id}_{M_j} \succeq \|A_j\|^2 \mathrm{Id}_{M_j} \succeq A_j A_j^\top.$$

4.3.6 Extension to a general metric

In practice, one may be interested in more general problems of the form [Chouzenoux et al., 2014]:

$$\text{Find } \widehat{x} = \operatorname*{argmin}_{x\in\mathbb{R}^N} \; f(x) + \sum_{j=1}^{J} g_j(A_j x) + \frac{1}{2}\|x-\widetilde{x}\|_C^2. \tag{4.24}$$

where $C \in \mathbb{R}^{N\times N}$ is a symmetric strictly positive definite matrix. Algorithms can be deduced from Algorithms 14 and 15 by simply replacing the Euclidean metric of $\mathbb{R}^N$ by the metric induced by C (while keeping the standard Euclidean metric for the spaces $\mathbb{R}^{M_j}$ with $j \in \{1,\dots,J\}$). By noticing that in the new metric, the adjoints of operators $(A_j)_{1\leqslant j\leqslant J}$ are replaced by $(C^{-1}A_j^\top)_{1\leqslant j\leqslant J}$, Algorithm 14 yields:

Algorithm 18 Dual block forward-backward algorithm in a general metric

Initialization:

Let $(y_0^j)_{1\leqslant j\leqslant J} \in \mathbb{R}^M$

For every $n \in \mathbb{N}, \gamma_n \in]0,+\infty[$

for $n = 0, 1, \dots$ **do**

$x_n = \mathrm{prox}_{C,f}(\widetilde{x} + z_n)$

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

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

$y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n B_{j_n}^{-1} \mathrm{prox}_{\gamma_n B_{j_n}^{-1}, g_{j_n}}\left(\gamma_n^{-1} B_{j_n} \widetilde{y}_n^{j_n}\right)$

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

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

end for

where

$$z_0 = -C^{-1} \sum_{j=1}^{J} A_j^\top y_0^j \quad \text{and } \forall j \in \{1,\dots,J\} \quad B_j \succeq A_j C^{-1} A_j^\top.$$

Note that, when $J = 1$, we recover a preconditioned version of Algorithm 10. Similarly, a new algorithm can be derived from Algorithm 15 for computing the sought proximity operator in the metric induced by the matrix C when $f = 0$. This is achieved by simply substituting the adjoints operators of $(A_j)_{1\leqslant j\leqslant J}$ with $(C^{-1}A_j^\top)_{1\leqslant j\leqslant J}$, when updating the primal variable x_n.

4.4 Convergence analysis

We will need some additional assumptions in order to establish the convergence of the preconditioned dual block forward-backward Algorithm 14:

Assumption 4.3

1. *For every $j \in \{1,\dots,J\}$, the restriction of g_j^* on its domain is continuous.*
2. *The sequence $(j_n)_{n\in\mathbb{N}}$ follows a quasi-cyclic rule, i.e., there exists $K \geqslant J$ such that, for every $n \in \mathbb{N}$, $\{1,\dots,J\} \subset \{j_n,\dots,j_{n+K-1}\}$.*
3. *The functions f and $(g_j)_{1\leqslant j\leqslant J}$ are semi-algebraic.*

The following result can then be established:

Proposition 4.4 *Suppose that Assumptions 4.1 and 4.3 hold. Let $(x_n)_{n\in\mathbb{N}}$ and $\big(y_n = (y_n^j)_{1\leqslant j\leqslant J}\big)_{n\geqslant 1}$ be sequences generated by Algorithm 14. If $(y_n)_{n\geqslant 1}$ is bounded, then $(x_n)_{n\in\mathbb{N}}$ converges to the solution to the primal problem (4.13) and $(y_n)_{n\geqslant 1}$ converges to a solution to the dual one (6.4).*

Proof. We have seen that our algorithm amounts to applying a block-coordinate forward-backward approach to the function:

$$\Phi\colon (y^j)_{1\leqslant j\leqslant J} \mapsto \varphi\Big(-\sum_{j=1}^{J} A_j^\top y^j + \widetilde{x}\Big) + \sum_{j=1}^{J} g_j^*(y^j). \tag{4.25}$$

Since $\|\cdot\|^2$ is a semi-algebraic function and semi-algebraicity is preserved under standard operations such as sum, infimum, conjugate, and inf-convolution, it can be deduced from Assumption 4.3.3 that Φ is semi-algebraic. It follows from [Chouzenoux et al., 2016, Theorem 3.1] that the sequence $\big(y_n\big)_{n\geqslant 1}$ generated by Algorithm 14 converges to a critical point $\widehat{y}$ of Φ. Since Φ is a convex function, such a critical point is a (global) minimizer of Φ. By using now (4.6) and the continuity of the proximity operator, it follows that the sequence $(x_n)_{n\in\mathbb{N}}$ converges to a solution $\widehat{x}$ satisfying (4.7). As already mentioned, $\widehat{x}$ is then the solution to (4.13). □

Remark 4.5

1. The boundedness of sequence $(y_n)_{n\geqslant 1}$ is satisfied if Φ is a coercive function. This happens, in particular, if all the functions $(g_j^*)_{1\leqslant j\leqslant J}$ are coercive, that is when, for every $j \in \{1,\dots,J\}$, $0 \in \operatorname{int}(\operatorname{dom} g_j)$ [Bauschke and Combettes, 2017, Proposition 14.16].

2. The quasi-cyclic rule (also sometimes called essentially cyclic rule) provides much more flexibility than the cyclic one. In particular, some of the (blocks of) variables may be activated more frequently than others, and the order in which the variables are swept can be randomly chosen.

Some more accurate convergence rate results can also be provided. In particular, we give below conditions for which the linear convergence of the proposed algorithm is secured.

Proposition 4.6 *Suppose that Assumptions 4.1 and 4.3 hold and that $\widehat{x}$ and $\widehat{y}$ are the limits of the sequences $(x_n)_{n\in\mathbb{N}}$ and $\big(y_n = (y_n^j)_{1\leqslant j\leqslant J}\big)_{n\geqslant 1}$, respectively. assuming that $(y_n)_{n\in\mathbb{N}}$ is bounded, there exist $\alpha \in]0,+\infty[$ and $\lambda \in]0,+\infty[$ such that, for every $n \geqslant 1$,*

$$\|x_n - \widehat{x}\| \leqslant \lambda \|A\| n^{-\alpha} \tag{4.26}$$
$$\|y_n - \widehat{y}\| \leqslant \lambda n^{-\alpha}. \tag{4.27}$$

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

1. *Φ, as defined by (4.25), is strongly convex,*

2. *f is Lipschitz differentiable and A is surjective*[1]*,*

3. *For every $j \in \{1,\dots,J\}$, g_j is Lipschitz differentiable,*

4. *Φ is a piecewise polynomial function of degree 2,*

5. *f is a quadratic function and, for every $j \in \{1,\dots,J\}$, g_j^* is a piecewise polynomial function of degree 2,*

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

$$\|x_n - \widehat{x}\| \leqslant \lambda' \|A\| \tau^n \tag{4.28}$$
$$\|y_n - \widehat{y}\| \leqslant \lambda' \tau^n. \tag{4.29}$$

[1] It is sometimes said that A is full row rank.

Proof. As shown by [Chouzenoux et al., 2016, Theorem 3.2], the convergence rate of the dual forward-backward algorithm depends on the Łojasiewicz exponent of function Φ defined by (4.25) at $\widehat{y}$. Then, (4.27) corresponds to the worst case upper bound. It then follows from (4.6), (4.7), and the nonexpansiveness of the proximity operator [Bauschke and Combettes, 2017] that, for every $n \geqslant 1$,

$$\begin{aligned} \|x_n - \widehat{x}\| &= \|\mathrm{prox}_f(\widetilde{x} - A^\top y_n) - \mathrm{prox}_f(\widetilde{x} - A^\top \widehat{y})\| \\ &\leqslant \|A^\top (y_n - \widehat{y})\| \\ &\leqslant \|A\| \|y_n - \widehat{y}\|, \end{aligned} \tag{4.30}$$

which yields (4.26).

If Φ is a strongly convex function [Bolte et al., 2016] or Φ is a piecewise polynomial function of degree 2 [Bolte et al., 2016], the Łojasiewicz exponent of function Φ is equal to 1/2. It then follows from [Chouzenoux et al., 2016, Theorem 3.2] that (4.29) holds. The decay behavior of $(x_n)_{n\geqslant 1}$ in (4.28) is then deduced as previously. If f is Lipschitz differentiable, then $f + \frac{1}{2}\|\cdot\|^2$ is also Lipschitz differentiable, and its conjugate φ is thus strongly convex [Bauschke and Combettes, 2017]. Since A is surjective,

$$(y^j)_{1\leqslant j\leqslant J} \mapsto \varphi\Big(- \sum_{j=1}^{J} A_j^\top y^j + \widetilde{x}\Big)$$

is strongly convex. The strong convexity of Φ is then guaranteed.
Similarly, if Condition 3 holds, then, for every $j \in \{1, \ldots, J\}$, g_j^* is strongly convex, hence Φ.
Finally, if Condition 5 holds, $f + \frac{1}{2}\|\cdot\|^2$ is a quadratic function and so is its conjugate φ. Since functions $(g_j^*)_{1\leqslant j\leqslant J}$ are assumed to be piecewise polynomial functions of degree 2, Φ is a piecewise polynomial function of degree 2. □

4.5 Conclusion

We provided in this chapter several primal-dual splitting algorithms for computing the proximity operator of convex composite functions. These algorithms rely on primal-dual formulation of the considered minimization problem, which gives the possibility of handling numerous linear operators without having to invert them. Moreover, the proposed algorithms benefit form a block-coordinate strategy that allows to improve their flexibility and convergence speed. Hence, the proposed algorithms are well-adapted to large-scale optimization problems, which have a large spectrum of real-world applications. The convergence of our approach has been theoretically analyzed and some experimental results in the context of video deconvolution and super-resolution will be supplied in the next chapter.

- Chapter 5 -

Application to video deconvolution and deinterlacing

Contents

5.1 Introduction

We have introduced in Chapter 4 new algorithms for computing the proximity operator of a sum of convex composite functions. We will now apply the proposed algorithms to the problem of joint deconvolution and super-resolution of video sequences [Elad and Feuer, 1999; Farsiu et al., 2006; Héas et al., 2016]. More precisely, we consider the problem of estimating an unknown progressive video from an interlaced one, in-which each frame is formed by merging two successive fields resulting from odd (resp. even) horizontal lines of the first (resp. second) frame [Jensen and Anastassiou, 1993]. The human visual system becomes more sensitive to interlacing artefacts on new HD flat LCD and plasma screens [Keller et al., 2005]. Hence, the need for high quality progressive videos has become essential to meet the actual customer's demand [Keller, 2007]. For this purpose, we design a new deconvolution and deinterlacing method that resorts to the dual block-coordinate forward-backward method that has been proposed in Chapter 4.

This chapter is organised as follows: we present in Section 5.2 the underlying degradation model and provide a formulation of its resolution through the minimization of a convex cost function. Then, we propose in Section 5.3 a minimization strategy to resolve it based on the dual block-coordinate forward backward algorithm. The performance of our deconvolution and deinterlacing method is assessed in terms of restoration quality and convergence speed in Section 5.4. Finally, some conclusions are given in Section 5.5.

5.2 Observation model

Let us denote by $\overline{\boldsymbol{x}} = (\overline{x}_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^{TN}$ the original high-resolution video sequence and $\boldsymbol{y} = (y_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^{TQ}$ the observed low-resolution sequence, with T the number of time frames, and N (resp. Q) the number of pixels of each image in the HR (resp. LR) sequence. The direct model relating $\overline{\boldsymbol{x}}$ to $\boldsymbol{y}$ can be expressed by

$$(\forall t \in \{1,\dots,T\}) \quad y_t = \mathrm{S}_t\,(h * \overline{x}_t) + w_t, \tag{5.1}$$

where $\mathrm{S}_t \in \mathbb{R}^{Q\times T}$ represents a decimation operator, $h \in \mathbb{R}^P$ corresponds to a convolution kernel accounting for spatial blur, and $(w_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^{TQ}$ is a Gaussian additive noise. In our context, the down-sampling S_t is a row decimation operator where $\mathrm{S}_t = \mathrm{S}_\mathrm{o}$ for odd frames and $\mathrm{S}_t = \mathrm{S}_\mathrm{e}$ for even frames. Note that, the number of rows in the progressive video sequence $(\overline{x}_t)_{1\leqslant t\leqslant T}$ is equal to twice that of the fields in the interlaced video sequence $(y_t)_{1\leqslant t\leqslant T}$, thereby we have $N = 2Q$.

An estimate of the original unknown sequence can be obtained by finding a solution to the following penalized least squares problem:

$$\underset{\boldsymbol{x}\in\mathbb{R}^{TN}}{\text{minimize}} \;\; F(\boldsymbol{x}) = \Phi(\boldsymbol{x}) + \Psi(\boldsymbol{x}), \tag{5.2}$$

where Φ denotes the data fidelity term given by

$$(\forall \boldsymbol{x} \in \mathbb{R}^{TN}) \quad \Phi(\boldsymbol{x}) = \frac{1}{2} \sum_{t=1}^{T} \|\mathrm{S}_t(h * x_t) - y_t\|^2, \tag{5.3}$$

and Ψ is a regularization function introducing prior information on the unknown video sequence, which is defined as

$$(\forall \boldsymbol{x} \in \mathbb{R}^{TN}) \quad \Psi(\boldsymbol{x}) = \sum_{t=1}^{T} \psi_t(x_t) + \iota_{[x_{\min}, x_{\max}]^{TN}}(\boldsymbol{x}) + \mathcal{M}(\boldsymbol{x}). \tag{5.4}$$

The indicator function $\iota_{[x_{\min}, x_{\max}]^{TN}}$ imposes a range $[x_{\min}, x_{\max}]$ on each pixel value of the images composing the video sequence, ψ_t is a spatial regularization term that handles each image $x_t \in \mathbb{R}^N$ independently, while $\mathcal{M}$ accounts for a temporal regularization term.

5.2.1 Spatial regularization

For every $t \in \{1, \ldots, T\}$, ψ_t incorporates prior information on each image $x_t \in \mathbb{R}^N$. We propose to employ in this chapter the *semi-local total variation* [Condat, 2014] defined by

$$\begin{aligned} (\forall z \in \mathbb{R}^N) \quad \psi_t(z) &= \eta \ \mathrm{sltv}(z), \\ &= \sum_{\ell \in \Omega} \chi_2 \left(Dz - V_\ell Dz \right). \end{aligned} \tag{5.5}$$

where $\eta \geqslant 0$, $D \in \mathbb{R}^{2N \times N}$ is the concatenation of the horizontal and vertical gradient operators:

$$D = \begin{bmatrix} \nabla_{\mathsf{H}} \\ \nabla_{\mathsf{V}} \end{bmatrix}, \quad \text{with} \quad \nabla_{\mathsf{H}} \in \mathbb{R}^{N \times N}, \quad \nabla_{\mathsf{V}} \in \mathbb{R}^{N \times N}, \tag{5.6}$$

$\Omega = \{1, \ldots, 6\}$ and $(V_\ell)_{\ell \in \{1,\ldots,6\}} \in \mathbb{R}^{2N \times 2N}$ represent shift operators as illustrated in Figure 5.1. Moreover, for every $q \in \mathbb{N}^*$, $\chi_q \colon \mathbb{R}^{qL} \to \mathbb{R}$ is given by

$$\left(\forall (z_1, \ldots, z_q) \in (\mathbb{R}^L)^q\right) \qquad \chi_q(z_1, \ldots, z_q) = \sum_{k=1}^{L} \sqrt{(z_{1,k})^2 + \cdots + (z_{q,k})^2}. \tag{5.7}$$

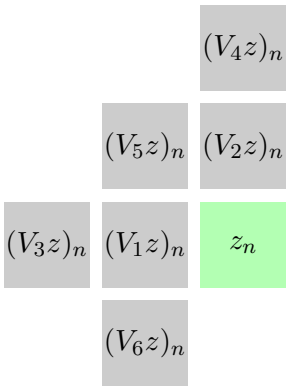


Figure 5.1: *Shift operators* $(V_\ell)_{\ell\in\{1,\dots,6\}}$ *applied to a given pixel position* $n \in \{1,\dots,N\}$.

Note that (5.5) can be rewritten as

$$(\forall z \in \mathbb{R}^N) \quad \mathrm{sltv}(z) = \sum_{\ell\in\Omega} \chi_2\left(L_\ell z\right) \text{ with, for every } \ell \in \Omega, \quad L_\ell = (\mathrm{Id}_{2N} - V_\ell)D. \tag{5.8}$$

5.2.2 Temporal regularization

The temporal regularization function $\mathcal{M}$ in (5.4) is employed in order to take into account temporal redundancies between successive frames. It is defined as follows

$$(\forall \boldsymbol{x} \in \mathbb{R}^{TN}) \quad \mathcal{M}(\boldsymbol{x}) = \sum_{t=1}^{T} \sum_{\ell\in\mathcal{V}_t} \beta_{\ell,t} \|x_t - M_{\ell\to t} x_\ell\|_1, \tag{5.9}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm, in addition, for every t and ℓ, $\beta_{\ell,t}$ are positive weights selected proportionally to the distance $|t-\ell|$ between the frame index of images x_t and x_ℓ, the index set $\mathcal{V}_t$ defines the neighborhood of the current image x_t (i.e., $\ell \in \mathcal{V}_t$ is such that $|\ell - t|$ is small), and $M_{\ell\to t} \in \mathbb{R}^{N\times N}$ is a linear operator modelling the motion fields between the current image x_t and the neighboring image x_ℓ. The matrices $M_{\ell\to t}$ are related to some vertical and horizontal shift matrices $u_{\ell\to t} \in \mathbb{R}^{N_1\times N_2}$ and $v_{\ell\to t} \in \mathbb{R}^{N_1\times N_2}$ respectively, with N_1 (resp. N_2) corresponding to the height (resp. width) of the images (i.e., $N_1N_2 = N$), in such a way that, $(\forall i \in \{1,\dots,N_1\})$ $(\forall j \in \{1,\dots,N_2\})$:

$$M_{\ell\to t} x_\ell(i,j) \approx x_\ell\left(i - u_{\ell\to t}(i,j)\,,\, j - v_{\ell\to t}(i,j)\right). \tag{5.10}$$

More precisely, we set

$$u_{\ell\to t} = \overline{u}_{\ell\to t} + \delta^u_{\ell\to t} \quad \text{and} \quad v_{\ell\to t} = \overline{v}_{\ell\to t} + \delta^v_{\ell\to t}, \tag{5.11}$$

where $\overline{u}_{\ell\to t}$ and $\overline{v}_{\ell\to t}$ represent the integer part of $u_{\ell\to t}$ and $v_{\ell\to t}$ respectively, and $\delta^u_{\ell\to t}$, $\delta^v_{\ell\to t}$ are their decimal part. We propose to resort to the following bilinear interpolation in order to approximate (5.10): $(\forall i \in \{1,\dots,N_1\})$ $(\forall j \in \{1,\dots,N_2\})$

$$\begin{aligned} M_{\ell\to t}x_\ell(i,j) &= (1-\delta^u_{\ell\to t}(i,j))\,(1-\delta^v_{\ell\to t}(i,j))\,x_\ell\,(i-\overline{u}_{\ell\to t}(i,j)\,,\,j-\overline{v}_{\ell\to t}(i,j)) \\ &+ (1-\delta^u_{\ell\to t}(i,j))\,\delta^v_{\ell\to t}(i,j)x_\ell\,(i-\overline{u}_{\ell\to t}(i,j)\,,\,j-\overline{v}_{\ell\to t}(i,j)-1) \\ &+ \delta^u_{\ell\to t}(i,j)\,(1-\delta^v_{\ell\to t}(i,j))\,x_\ell\,(i-\overline{u}_{\ell\to t}(i,j)-1\,,\,j-\overline{v}_{\ell\to t}(i,j)) \\ &+ \delta^u_{\ell\to t}(i,j)\delta^v_{\ell\to t}(i,j)\,x_\ell\,(i-\overline{u}_{\ell\to t}(i,j)-1\,,\,j-\overline{v}_{\ell\to t}(i,j)-1)\,. \end{aligned} \tag{5.12}$$

Thus

$$M_{\ell\to t} = D_{1,\ell\to t}\,M_{1,\ell\to t} + D_{2,\ell\to t}\,M_{2,\ell\to t} + D_{3,\ell\to t}\,M_{3,\ell\to t} + D_{4,\ell\to t}\,M_{4,\ell\to t}, \tag{5.13}$$

where $D_{k,\ell\to t} \in \mathbb{R}^{N\times N}$ with $k \in \{1,\dots,4\}$ are diagonal matrices such that, for every $y \in \mathbb{R}^N$, for every $i \in \{1,\dots,N_1\}$ and for every $j \in \{1,\dots,N_2\}$:

$$\begin{aligned} D_{1,\ell\to t}y(i,j) &= (1-\delta^u_{\ell\to t}(i,j))\;(1-\delta^v_{\ell\to t}(i,j))\;\;y(i,j), \\ D_{2,\ell\to t}y(i,j) &= (1-\delta^u_{\ell\to t}(i,j))\;\delta^v_{\ell\to t}(i,j)\;y(i,j), \\ D_{3,\ell\to t}y(i,j) &= \delta^u_{\ell\to t}(i,j)\;(1-\delta^v_{\ell\to t}(i,j))\;\;y(i,j), \\ D_{4,\ell\to t}y(i,j) &= \delta^u_{\ell\to t}(i,j)\;\delta^v_{\ell\to t}(i,j)\;y(i,j), \end{aligned}$$

and $M_{k,\ell\to t} \in \{0,1\}^{N\times N}$, $k \in \{1,\dots,4\}$, are defined as

$$\begin{aligned} M_{1,\ell\to t}y(i,j) &= y\,(i-\overline{u}_{\ell\to t}(i,j)\,,\,j-\overline{v}_{\ell\to t}(i,j))\,, \\ M_{2,\ell\to t}y(i,j) &= y\,(i-\overline{u}_{\ell\to t}(i,j)\,,\,j-\overline{v}_{\ell\to t}(i,j)-1)\,, \\ M_{3,\ell\to t}y(i,j) &= y\,(i-\overline{u}_{\ell\to t}(i,j)-1\,,\,j-\overline{v}_{\ell\to t}(i,j))\,, \\ M_{4,\ell\to t}y(i,j) &= y\,(i-\overline{u}_{\ell\to t}(i,j)-1\,,\,j-\overline{v}_{\ell\to t}(i,j)-1)\,. \end{aligned}$$

The adjoint operator $(M_{k,\ell\to t})^\top$ is such that, for every $n \in \{1,\dots,N\}$ and $k \in \{1,\dots,4\}$, the n'-th component of $\left((M_{k,\ell\to t})^\top y\right)$ with $y \in \mathbb{R}^N$, corresponds to the sum of all the pixels located at $n \in \{1,\dots,N\}$ in the image y to which the pixel of index n' has been displaced in the resulting image $(M_{k,\ell\to t}y)$. Thereby, for every $n' \in \{1,\dots,N\}$

$$\left((M_{k,\ell\to t})^\top\,(D_{k,\ell\to t})^\top y\right)_{n'} = \sum_{n\in\mathcal{S}^k_{n',\ell\to t}} \left(D_{k,\ell\to t}\,y\right)_n, \tag{5.14}$$

where, for every $i \in \{1,\dots,N_1\}$ and for every $j \in \{1,\dots,N_2\}$,

$$\begin{aligned} \mathcal{S}^1_{n',\ell\to t} &= \{n \mid i = i' + \overline{u}_{\ell\to t}(i,j)\;;\; j = j' + \overline{v}_{\ell\to t}(i,j)\}, \\ \mathcal{S}^2_{n',\ell\to t} &= \{n \mid i = i' + \overline{u}_{\ell\to t}(i,j)\;;\; j = j' + \overline{v}_{\ell\to t}(i,j) + 1\}, \\ \mathcal{S}^3_{n',\ell\to t} &= \{n \mid i = i' + \overline{u}_{\ell\to t}(i,j) + 1\;;\; j = j' + \overline{v}_{\ell\to t}(i,j)\}, \\ \mathcal{S}^4_{n',\ell\to t} &= \{n \mid i = i' + \overline{u}_{\ell\to t}(i,j) + 1; j = j' + \overline{v}_{\ell\to t}(i,j) + 1\}, \end{aligned}$$

and n (resp. n') is the index of the pixel located at (i, j) (resp. (i', j')) in the corresponding image.

According to (5.13), the norm of the motion compensation operator $M_{\ell\to t}$ reads

$$\|M_{\ell\to t}\| = \|\sum_{k=1}^{4} D_{k,\ell\to t} M_{k,\ell\to t}\| \leqslant \sum_{k=1}^{4} \|D_{k,\ell\to t} M_{k,\ell\to t}\|. \tag{5.15}$$

Note that, for every $k \in \{1, \ldots, 4\}$, $M_{k,\ell\to t}$ is an $N \times N$ binary matrix. By definition, for every $n' \in \{1, \ldots, N\}$, the n'-th column of this matrix has nonzero entries at the row indices $n \in \mathcal{S}^k_{n',\ell\to t}$. Therefore, since $D_{k,\ell\to t}$ is a diagonal matrix,

$$(M_{k,\ell\to t})^\top (D_{k,\ell\to t})^\top D_{k,\ell\to t} M_{k,\ell\to t}$$

is also diagonal with n'-th diagonal entry equals

$$\sum_{n\in\mathcal{S}^k_{n',\ell\to t}} D_{k,\ell\to t}(n,n)^2.$$

Thus, $\|D_{k,\ell\to t} M_{k,\ell\to t}\|$ can be easily computed according to

$$\|D_{k,\ell\to t} M_{k,\ell\to t}\| = \max_{n'\in\{1,\ldots,N\}} \left(\sqrt{\sum_{n\in\mathcal{S}^k_{n',\ell\to t}} D_{k,\ell\to t}(n,n)^2} \right). \tag{5.16}$$

5.3 Minimization strategy

A solution to Problem (5.2) can be obtained by making use of PALM Algorithm 19 proposed in [Bolte et al., 2014] (see also Algorithm 8 for recent extensions) which provides an asymptotically exact solution to (5.2). The images $(x_t)_{1\leqslant t\leqslant T}$ are processed sequentially, where at each iteration, an image x_t is updated with a forward-backward iteration that consists of a gradient step on Φ with respect to x_t, and a proximal step on Ψ_t which represents the restriction of Ψ to the t-th image defined as: for every $\boldsymbol{x} \in \mathbb{R}^{TN}$,

$$\begin{aligned}(\forall z \in \mathbb{R}^N) \quad \Psi_t(z|\boldsymbol{x}) = \eta \sum_{\ell\in\Omega} \chi_2\left(L_\ell z\right) \; + \; \iota_{[x_{\min},x_{\max}]^N}(z) \\ + \sum_{\ell\in\mathcal{V}_t} \beta_{\ell,t} \|z - M_{\ell\to t} x_\ell\|_1 + \sum_{\ell\in\mathcal{V}_t} \beta_{t,\ell} \|x_\ell - M_{t\to\ell} z\|_1,\end{aligned} \tag{5.17}$$

thus, the number of terms in (5.17) is equal to

$$J = |\Omega| + 2\,|\mathcal{V}_t| + 1, \tag{5.18}$$

where $|\mathcal{Z}|$ denotes the cardinality of a set $\mathcal{Z}$. PALM algorithm reads:

Algorithm 19 PALM algorithm for solving Problem (5.2)

Initialization:

Let $(x_t^0)_{1\leqslant t\leqslant T} \in \mathbb{R}^{TN}$

For every $k \in \mathbb{N}$ and $t \in \{1, \ldots, T\}$, $\sigma_t^k \in]0, +\infty[$

for $k = 0, 1, \ldots$ **do**

  for $t = 1, \ldots, T$ **do**

$$\breve{x}^{t,k} = \left(x_1^{k+1}, \ldots, x_{t-1}^{k+1},\ x_t^k\ , x_{t+1}^k \ldots, x_T^k\right)$$

$$\widetilde{x}_t^k = x_t^k - \sigma_t^k \left(\nabla_{x_t} \Phi(\breve{x}^{t,k})\right)$$

$$x_t^{k+1} = \mathrm{prox}_{(\sigma_t^k)^{-1} I_N, \Psi_t(\cdot|\breve{x}^{t,k})} \left(\widetilde{x}_t^k\right)$$

  end for

end for

where $\nabla_{x_t}\Phi$ denotes the gradient of Φ with respect to x_t and

$$0 < \sigma_t^k < 2\theta_t^{-1},$$

with θ_t the Lipschitz constant of $\nabla_{x_t}\Phi$ (i.e., $\theta_t = \|S_t H\|^2$ with $H \in \mathbb{R}^{N\times N}$ the Hankel-block Hankel matrix form of the convolution kernel h). Since F is semi-algebraic and Φ is Lipschitz differentiable, the sequence $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ generated by PALM algorithm is guaranteed to converge to a solution to Problem (5.2) [Bolte et al., 2014].

As the proximity operator of the function (5.17) does not have a closed form expression and involves several linear operators, we resort to an inner iteration to estimate it by means of Algorithms 14, 15, and 17. Note that, when implementing Algorithm 14, function f in (4.13) is chosen equal to $\iota_{[x_{\min}, x_{\max}]^N}$ since it does not involve any linear operator, whereas, in Algorithms 15 and 17, the latter function is regarded as some of the g_j functions, the corresponding A_j being the identity matrix. Let us emphasize that PALM algorithm is robust to computational errors in the proximal step [Chouzenoux et al., 2016], assuming that a sufficient decrease condition is satisfied. In practice, a rough stopping criterion on the inner loop will be used in order to avoid numerical instabilities.

5.4 Experimental results

5.4.1 Data-set Benchmark

We evaluate the performance of our method using a benchmark of four sequences of images:

- Two synthetic video sequences **Foreman** and **Claire** of size $N = 352 \times 288$ (resp. $N = 360 \times 288$) composed of $T = 50$ frames. These video sequences were blurred with the horizontal convolution kernel shown in Figure 5.6 which corresponds to a realistic model of the observed degradations in the context of old television archives, then interlaced and finally corrupted with a white Gaussian noise. The process results in a degraded video sequence with spatial dimension $Q = 352 \times 144$ (resp. $Q = 360 \times 144$). The videos are sourced from http://media.xiph.org/video/derf/.

- Two real interlaced sequences of size $Q = 720 \times 288$ supplied by INA from French broadcast archive programmes **Au théâtre ce soir** and **Tachan**. We extract $T = 50$ fields from each sequence and apply our method to recover progressive sharp video sequences with resolution $N = 720 \times 576$.
  For the deconvolution task, we use spatial convolution kernels shown in Figure 5.9a and Figure 5.9b, that are obtained using blind identification methods in [Krishnan et al., 2011] and [Abboud et al., 2014].

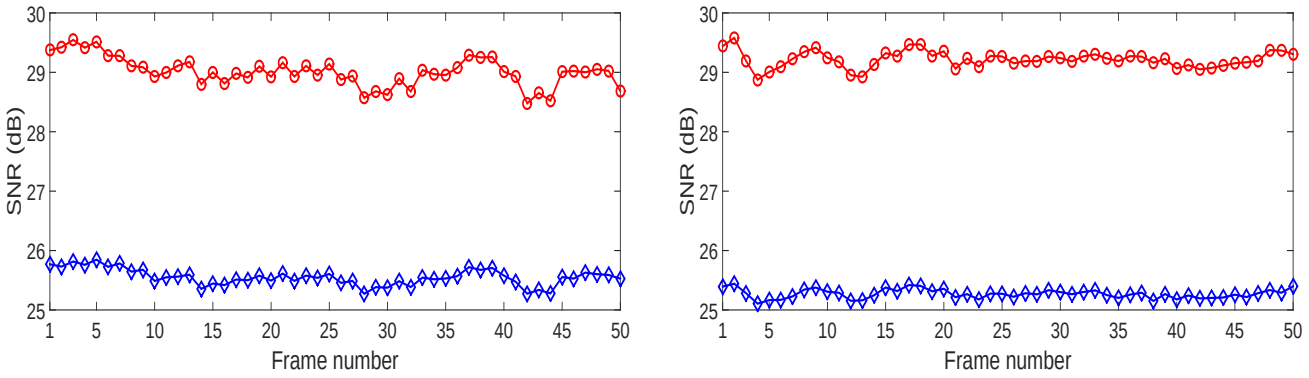
These sequences are provided as RGB videos. We apply our method on their luminance component only, which represents a grayscale version of the original images, while the two chrominance components are processed with a median filter of size 3×3 on each component separately, in order to reduce the residual persistent noise. The motion matrices $(u_{\ell \to t}, v_{\ell \to t})$ involved in the temporal regularization term are computed from the luminance component of the degraded sequences using the method described in [Liu et al., 2008], and then spatially interpolated to reach the final resolution. The neighborhood $\mathcal{V}_t$ includes the previous and next frames of the image x_t. Moreover the set Ω involved in the semi-local total variation term is of cardinality 6, so that the number of terms in (5.17) is equal to $J = 10$ or 11, namely:

- $(\forall j \in \{1, \dots, 6\}) \quad g_j = \chi_2 \;\text{ and }\; (A_j)_{1 \leqslant j \leqslant 6} = (L_\ell)_{\ell \in \Omega}$,
- $(\forall j \in \{7, 8\}) \quad g_j = \|\cdot\|_1 \;\text{ and }\; A_j = \mathrm{Id}_N$,
- $(\forall j \in \{9, 10\}) \quad g_j = \|\cdot\|_1 \;\text{ and }\; (A_j)_{9 \leqslant j \leqslant 10} = (M_{t \to \ell})_{\ell \in \mathcal{V}_t}$,
- For Algorithm (15) or (17), $\quad g_{11} = \iota_{[x_{\min}, x_{\max}]^N} \;\text{ and }\; A_{11} = \mathrm{Id}_N$.

5.4.2 Numerical performance

Restoration quality

Table 5.1 presents the performance of our restoration method in terms of SNR, averaged SSIM [Wang et al., 2004], and MOVIE [Seshadrinathan and Bovik, 2010]. The latter is a video quality assessor, that takes into account both spatial and temporal aspects in the quality measurement. Moreover, the results in terms of SNR per frame are displayed in Figures 5.2a and 5.2b. The simulations are run



(a) ***Foreman*** *sequence* **(b)** ***Claire*** *sequence*

Figure 5.2: *SNR values per frame : Degraded (blue diamond), restored (red circle).*

using 100 iterations of PALM algorithm, which appears to be sufficient to reach the convergence of the method. Note that the values related to the degraded sequences are evaluated on a spatially interpolated version of them, with a final resolution equals to N. In addition, it should be mentioned that, for each test scenario, the restoration results we obtained are similar in terms of visual quality, regardless of the chosen optimization algorithm.

Sequences		SNR (dB)	SSIM	MOVIE
Foreman	Degraded	25.54	0.78	4.34×10^{-4}
	Restored	28.95	0.90	3.73×10^{-4}
Claire	Degraded	25.27	0.85	1.97×10^{-3}
	Restored	29.21	0.96	1.77×10^{-3}

Table 5.1: *Quality of our deinterlacing and deconvolution method.*

Our reconstruction method achieves good quality results for all tested sequences. This can also be assessed by visual inspection on Figures 5.7 and 5.8, and for the real sequences on Figures 5.10 and 5.11, for which no ground truth is available. The motion compensation terms play a central role in the restoration quality, especially in the deinterlacing process. This is emphasized in the case of **Foreman** sequence, where the motion between two successive images is fast, which leads to a rough estimation of motion operators, at the price of a lower improvement of the restoration quality, especially in terms of MOVIE.

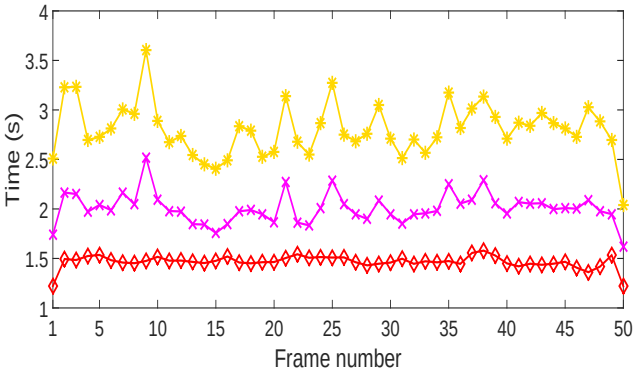
Convergence speed

Let us analyse the convergence speed of the proposed algorithms. First, in order to investigate the impact of preconditioning strategies, we have carried out a number of tests regarding the preconditioning matrices related to the involved linear operators $(A_j)_{1\leqslant j\leqslant J}$. These evaluations are performed on the synthetic sequence **Foreman** using Algorithm 19, combined with Algorithm 15. We work with diagonal preconditioning matrices in order to achieve a good trade-off between the convergence acceleration and the computation time.
The tested preconditioning matrices are

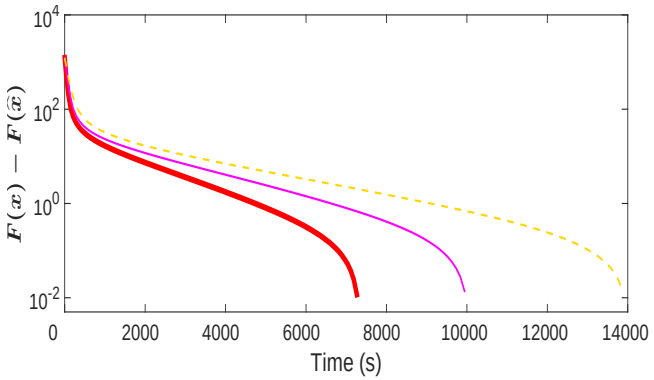
- $\forall j \in \{1,\dots,J\} \quad B_j = \|A_j\|^2 \operatorname{Id}_{M_j}$, (5.19)
  and
- $\forall j \in \{1,\dots,J\} \quad B_j = \operatorname{Diag}\left(|A_j|\,|A_j^\top|\,1_{M_j}\right)$, (5.20)

where 1_{M_j} denotes the ones vector of $\mathbb{R}^{M_j}$.
In the non-preconditioned case (5.19), we need to supply the norms of the operators $(A_j)_{1\leqslant j\leqslant J}$. As far as motion compensation operators are concerned, this norm is either approximated using (5.15), or precomputed using the power iterative method in [Golub and Van Loan, 1996].

Figure 5.3a presents the average execution time needed for computing the proximity operator of Ψ_t per image, by means of Algorithm 15. The latter is stopped when the relative decrease of the criterion gets below 10^{-5} which appears sufficient in practice to ensure the stability of the whole PALM algorithm. A Matlab 7 implementation is used with an Intel(R) Xeon(R) E5-2670 CPU @ 2.3 GHz. We get an acceleration of factor 2 using the preconditioning strategy (5.20) instead of the approximated version of the non-preconditioned case (5.19), while the acceleration is of factor 4/3 if the exact norms of the motion operators are used in (5.19). Note however that the exact computation of these norms is not a realistic strategy when processing long videos at standard or high resolution, since it calls upon an iterative and costly method. Figure 5.3b shows the variation of the cost function $F(x)-F(\widehat{x})$ with respect to the execution time, where F is defined in (5.2), and $F(\widehat{x})$ represents the minimum of F obtained at the end of the corresponding simulations. Figures 5.4a-5.4d illustrate the averaged time spent in computing the proximity operator of Ψ_t for all the images composing the video sequences over 100 iterations of PALM Algorithm 19, using either Algorithm 14, 15, or 17. The preconditioning strategy (5.20) is used for Algorithms 14 and 15. Depending on the video sequence, the best performances in terms of computation time are obtained either with Algorithm 14 or Algorithm 15 with small differences between them. In addition, the dual FB Algorithm 17 from [Combettes et al., 2011] is up to 18 times slower to reach the stopping criterion. We have also compared the convergence speed of our approach (using Algorithm 14 and preconditioning formula (5.20)) with that of the



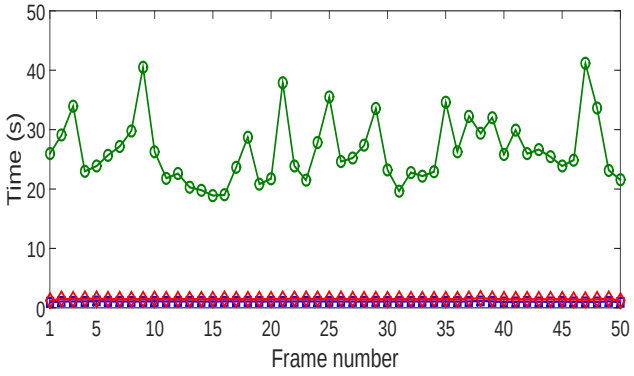
(a) *Average execution time per frame for computing* $\text{prox}_{(\sigma_t^k)^{-1}I_N,\Psi_t}$ *using Algorithm 15 and* ***Foreman*** *sequence: preconditioning strategy* (5.20) *(red diamond), no preconditioning (see* (5.19)*) using exact norms (magenta cross), and no preconditioning strategy* (5.19) *using approximate norms (yellow astrick)).*



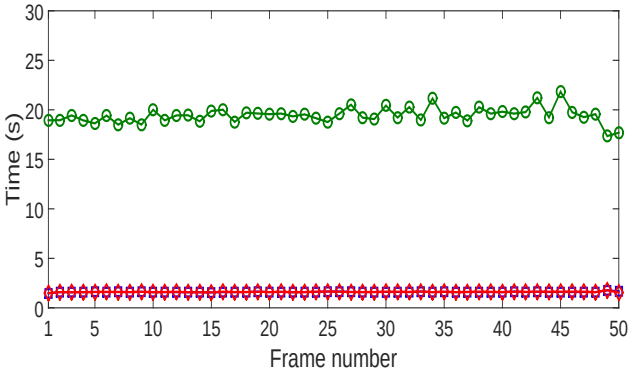
(b) *Comparison of the preconditioning strategies in terms of execution time (s.): preconditioning strategy* (5.20) *(solid tick red), no preconditioning strategy* (5.19) *with exact norms (solid thin magenta) and no preconditioning strategy* (5.19) *with approximated norms (dashed thin yellow).*

Figure 5.3: ***Foreman*** *sequence : Convergence acceleration.*

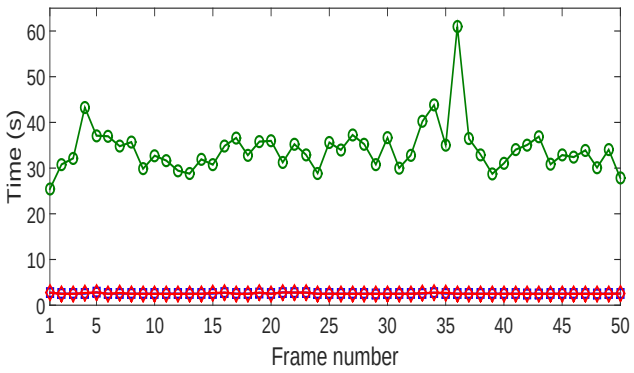
primal-dual algorithm from [Condat, 2013] for solving Problem (5.2). This evaluation is realized using Algorithm 19 combined with Algorithm 14 on the synthetic video **Foreman**. Figure 5.5 shows the variation of the cost function $F(x) - F(\widehat{x})$ with respect to execution time for both methods. We observe that the proposed algorithm reaches the sought solution about 4 times faster than the algorithm in [Condat, 2013]. This emphasizes the gain provided by our algorithms in terms of acceleration.



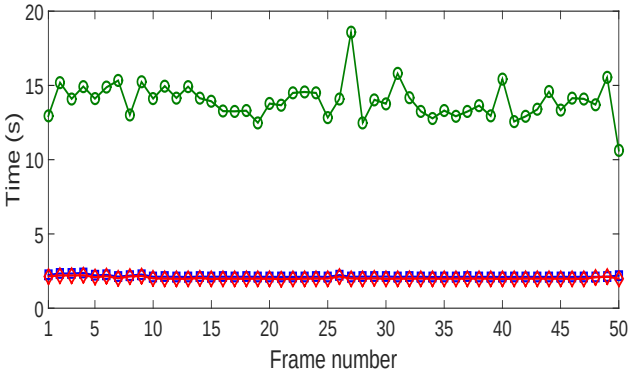
(a) ***Foreman*** *sequence.*



(b) ***Claire*** *sequence.*



(c) ***Tachan*** *sequence.*



(d) ***Au théâtre ce soir*** *sequence.*

Figure 5.4: *Averaged execution time (in s.) per frame: Algorithm 14 (blue square), Algorithm 15 (red diamond) and Algorithm 17 (green circle).*

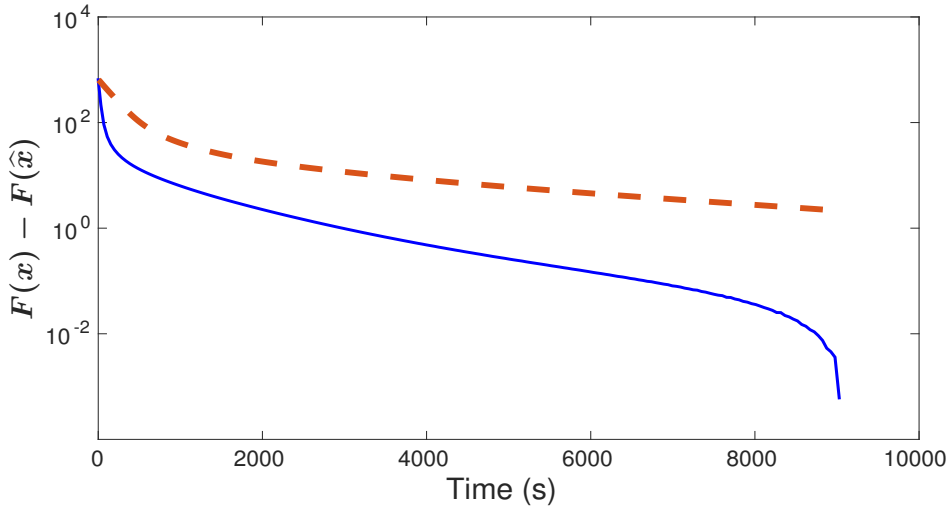


Figure 5.5: *Comparison between the proposed method and the one based on a primal-dual algorithm from [Condat, 2013] in terms of execution time (s.): proposed method with Algorithm 14 (solid thin blue), primal-dual-based method (dashed thick orange).*

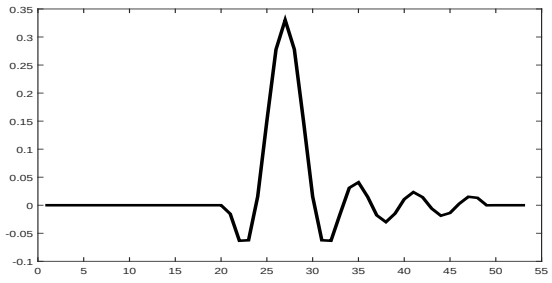


Figure 5.6: *Synthetic spatial convolution kernel, $P = 53$.*

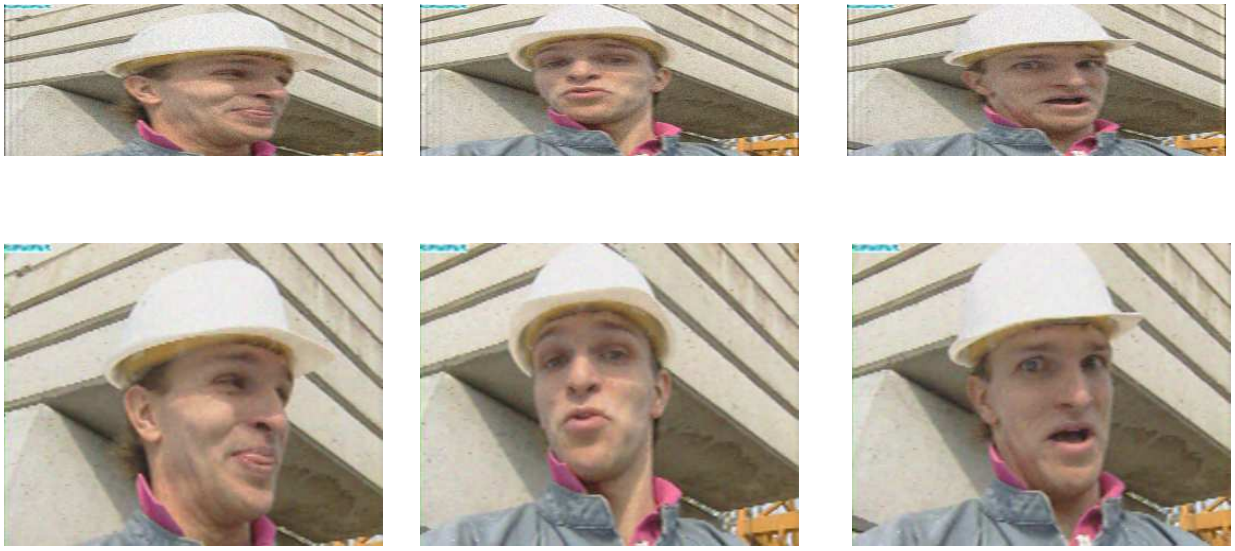
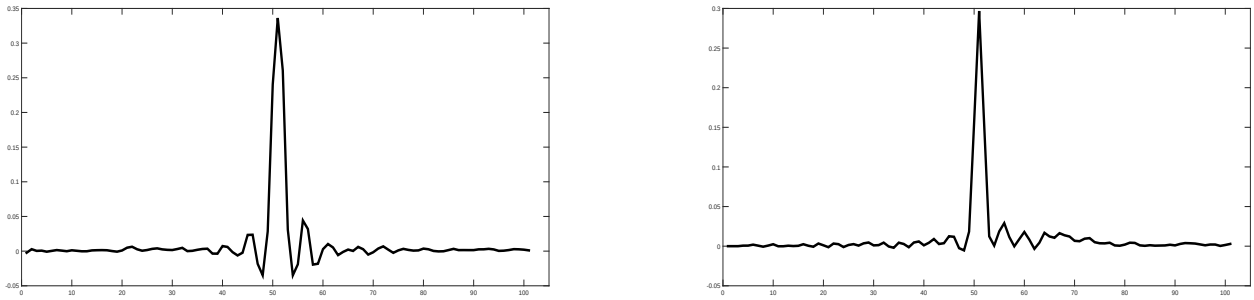


Figure 5.7: ***Foreman*** *sequence: degraded low resolution fields (top), restored high resolution images (bottom).*

Figure 5.8: ***Claire*** *sequence: degraded low resolution fields (top), restored high resolution images (bottom).*



(a) ***Tachan*** *kernel.* **(b)** ***Au théâtre ce soir*** *kernel.*

Figure 5.9: *Spatial convolution kernels for real sequences provided by INA,* $P = 101$.

Figure 5.10: ***Tachan*** *sequence: degraded low resolution fields (top), restored high resolution images (bottom).*

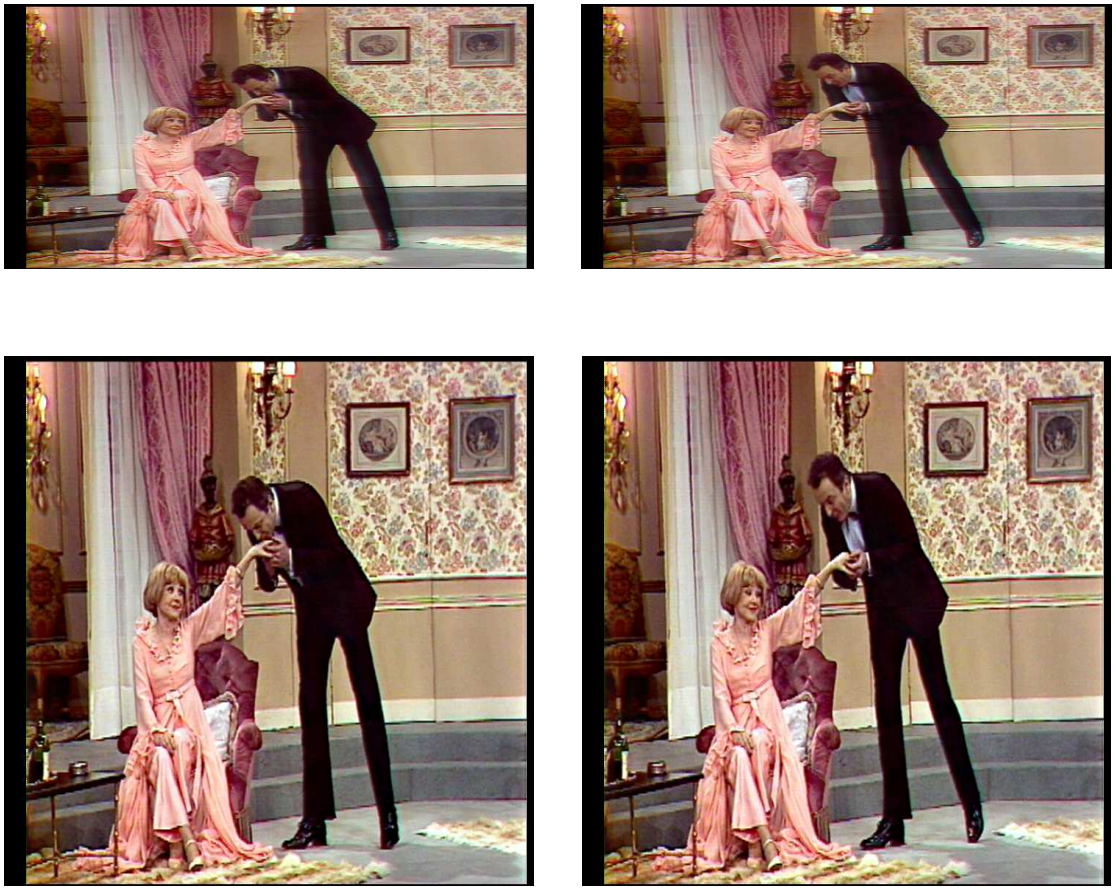


Figure 5.11: ***Au théâtre ce soir*** *sequence: degraded low resolution fields (top), restored high resolution images (bottom).*

5.5 Conclusion

We have addressed in this chapter the challenging problem of joint deconvolution and deinterlacing of blurred video sequences, which amounts to recover the missing even/odd rows of each successive frames, while keeping a good image quality. For this aim, we resorted to a penalized formulation of the problem, and used the block-coordinate algorithm proposed in Chapter 4 to solve it. The proposed algorithm has been compared to similar algorithms in the literature, and our simulation results show the good performance of the proposed method regarding both restoration quality and convergence speed. We propose in the next chapter to extend the algorithm proposed in Chapter 4 to a distributed framework, with the aim to handle the computation of the sought proximity operator in a more efficient manner, by taking advantage of the multi-cores architecture, available in recent computers systems.

- Chapter 6 -

Distributed algorithm for computing proximity operators

Contents

6.1 Introduction

We propose in this chapter a new distributed algorithm for computing the proximity operator of a sum of convex functions involving linear operators. The proposed algorithm extends the dual block preconditioned forward-backward algorithm that was introduced in Chapter 4, to a distributed asynchronous framework [Boyd et al., 2011; Pesquet and Repetti, 2015; Komodakis et al., 2015; Iutzeler et al., 2016; Richtárik and Takác, 2016; Komodakis et al., 2016]. Each involved function is now considered as locally related to a node of a connected hypergraph, where communications are allowed between neighboring nodes that share the same hyperedge. Moreover, our method takes advantage of variable metric techniques that have been shown to be efficient for accelerating the convergence speed of proximal approaches. It also benefits from primal-dual splitting strategies strengths (in particular their ability to handle a finite sum of convex functions without inverting none of the involved linear operators).

The remainder of this chapter is organized as follows: we introduce in Section 6.2 our minimization problem and recall the centralized algorithm for computing the sought proximity operator. In Section 6.3, we present a distributed extension of the algorithm. Afterwards, Section 6.4 proposes a practical version of our asynchronous algorithm and investigates its implementation on a distributed architecture. The performance of the proposed algorithm are evaluated in Section 6.5 regarding the acceleration with respect to the number of used computing units, and finally, some conclusions are drawn in Section 6.6.

6.2 Minimization problem

Let us first recall the considered optimization problem. We are interested, in this chapter, in computing the proximity operator of the following sum of functions at a given point $\widetilde{x}$ of $\mathbb{R}^N$:

$$\big(\forall x \in \mathbb{R}^N\big) \qquad G(x) = \sum_{j=1}^{J} g_j(A_j x), \tag{6.1}$$

where, for every $j \in \{1, \ldots, J\}$, $g_j : \mathbb{R}^{M_j} \to]-\infty, +\infty]$ is a proper lower-semicontinuous convex possibly nonsmooth function and A_j is a linear operator in $\mathbb{R}^{M_j \times N}$. In addition, it is assumed that

$$\bigcap_{j=1}^{J} \operatorname{dom}\left(g_j \circ A_j\right) \neq \varnothing. \tag{6.2}$$

Computing the proximity operator of G amounts to finding the solution to the

following minimization problem:

$$\begin{aligned}\text{Find } \widehat{x} &= \operatorname{prox}_G(\widetilde{x}) \\ &= \operatorname*{argmin}_{x\in\mathbb{R}^N} \sum_{j=1}^{J} g_j(A_j x) + \frac{1}{2}\|x-\widetilde{x}\|^2. \end{aligned} \tag{6.3}$$

A number of primal-dual proximal algorithms [Chambolle and Pock, 2010; Boţ and Hendrich, 2013; Combettes and Pesquet, 2015; Boyd et al., 2011] can be applied to solve Problem (6.3) by making use of its dual formulation given by:

$$\text{Find } \quad \widehat{y} = \operatorname*{argmin}_{y=(y^j)_{1\leqslant j\leqslant J}\in\mathbb{R}^M} \frac{1}{2}\Big\|\widetilde{x} - \sum_{j=1}^{J} A_j^\top y^j\Big\|^2 + \sum_{j=1}^{J} g_j^*(y^j), \tag{6.4}$$

where $(g_j^*)_{1\leqslant j\leqslant J}$ are the Fenchel-Legendre conjugate functions of $(g_j)_{1\leqslant j\leqslant J}$.
Among existing efficient primal-dual approaches, one can mention the dual block preconditioned forward-backward Algorithm 15, which benefits from the acceleration provided by variable metric methods through the introduction of preconditioning matrices. Moreover, convergence guaranties on both generated primal sequence $(x_n)_{n\in\mathbb{N}}$ and dual sequences $(y_n^j)_{n\in\mathbb{N}^*}$ with $j\in\{1,\ldots,J\}$ have been established for this algorithm in Section 4.4 under a quasi-cyclic rule on the block selection.

6.3 Distributed algorithm

Let us ground on Algorithm 15 in order to design a distributed (i.e., decentralized) solution to Problem (6.3). This can be achieved by resorting to a global consensus technique [Pustelnik et al., 2011; Boyd et al., 2011; Pesquet and Repetti, 2015; Komodakis and Pesquet, 2015] and rewriting the problem under the following form:

$$\text{Find } \widehat{\boldsymbol{x}} = \operatorname*{argmin}_{\boldsymbol{x}=(x^j)_{1\leqslant j\leqslant J}\in\Lambda} \sum_{j=1}^{J} g_j(A_j x^j) + \frac{1}{2}\sum_{j=1}^{J}\|x^j-\widetilde{x}\|_{\Omega_j}^2, \tag{6.5}$$

where $(\Omega_j)_{1\leqslant j\leqslant J}$ are diagonal $N\times N$ matrices with positive diagonal elements and Λ is the vector subspace of $\mathbb{R}^{NJ}$ defined so as to introduce suitable coupling constraints on the vectors $(x^j)_{1\leqslant j\leqslant J}$. The most standard choice for such constraint set is

$$\Lambda = \left\{ \begin{bmatrix} x^1 \\ \vdots \\ x^J \end{bmatrix} \in \mathbb{R}^{NJ} \mid x^1 = \ldots = x^J \right\}. \tag{6.6}$$

One can indeed notice that, provided that

$$\sum_{j=1}^{J} \Omega_j = \mathrm{Id}_N, \tag{6.7}$$

the solution to Problem (6.5) yields the solution to Problem (6.3) when the variables $(x^j)_{1\leqslant j\leqslant J}$ are all equal to the solution $\widehat{x}$ to Problem (6.3).

6.3.1 Local form of consensus

We propose instead to split the constraint set Λ into L local linear constraints Λ_ℓ. For every $\ell \in \{1,\dots,L\}$, each constraint set Λ_ℓ handles a nonempty subset $\mathbb{V}_\ell$ of $\{1,\dots,J\}$ with cardinality κ_ℓ such that, for every $\boldsymbol{x} = [(x^1)^\top,\dots,(x^J)^\top]^\top \in \mathbb{R}^{NJ}$,

$$\boldsymbol{x} \in \Lambda \quad \Leftrightarrow \quad (\forall \ell \in \{1,\dots,L\}) \quad (x^j)_{j\in\mathbb{V}_\ell} \in \Lambda_\ell. \tag{6.8}$$

Examples of vector subspaces $(\Lambda_\ell)_{1\leqslant\ell\leqslant L}$ allowing this condition to be satisfied will be discussed in Section 6.3.3. Each node $j \in \{1,\dots,J\}$ is associated with function g_j, which is considered as local and processes its own private data. Moreover, each node j is allowed to communicate with nodes that belong to the same set $\mathbb{V}_\ell$. The sets $(\mathbb{V}_\ell)_{1\leqslant\ell\leqslant L}$ can thus be viewed as the hyperedges of a hypergraph having J nodes. It is worth noticing that the case of a graph topology is encompassed by this formulation when setting for every $\ell \in \{1,\dots,L\}$ the cardinality of the set $\mathbb{V}_\ell$ equals to $\kappa_\ell = 2$. Figure 6.1 shows an illustrative example, where the hypergraph is composed of $J = 7$ nodes associated with functions $(g_j)_{1\leqslant j\leqslant 7}$ and $L = 4$ hyperedges represented by the sets $(\mathbb{V}_\ell)_{1\leqslant\ell\leqslant 4}$ with cardinalities $\kappa_1 = 3, \kappa_2 = 2, \kappa_3 = 2$, and $\kappa_4 = 3$, respectively. Node 4 belonging to the set $\mathbb{V}_2$ can communicate with node 5. Besides, node 3 belongs to $\mathbb{V}_1$ and $\mathbb{V}_4$, hence it is allowed to communicate with nodes $\{1,2,5,7\}$.

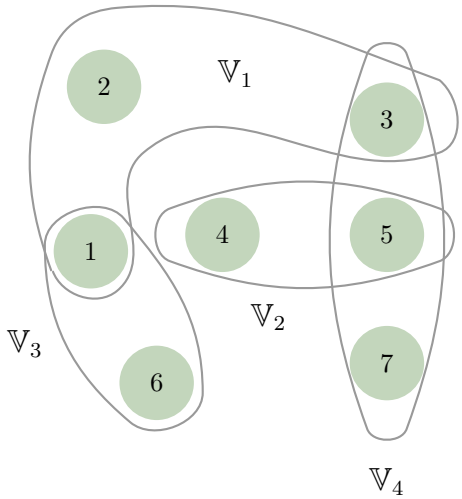


Figure 6.1: *Connected hypergraph of $J = 7$ nodes and $L = 4$ hyperedges.*

Let us define, for every $\ell \in \{1,\dots,L\}$, the matrix $\boldsymbol{S}_\ell \in \mathbb{R}^{N\kappa_\ell\times NJ}$ associated with constraint set Λ_ℓ, which extracts the vector $(x^j)_{j\in\mathbb{V}_\ell}$ from the concatenated vector $\boldsymbol{x} = [(x^1)^\top,\dots,(x^J)^\top]^\top \in \mathbb{R}^{NJ}$:

$$(x^j)_{j\in\mathbb{V}_\ell} = [(x^{\mathrm{i}(\ell,1)})^\top,\dots,(x^{\mathrm{i}(\ell,\kappa_\ell)})^\top]^\top = \boldsymbol{S}_\ell\,\boldsymbol{x}, \tag{6.9}$$

where $\mathsf{i}(\ell,1),\ldots,\mathsf{i}(\ell,\kappa_\ell)$ denote the elements of $\mathbb{V}_\ell$ ordered in an increasing manner. The transpose matrix of $(\boldsymbol{S}_\ell)_{1\leqslant\ell\leqslant L}$ is such that, for every $v^\ell=(v^{\ell,k})_{1\leqslant k\leqslant\kappa_\ell}\in\mathbb{R}^{N\kappa_\ell}$,

$$\boldsymbol{x}=[(x^1)^\top,\ldots,(x^J)^\top]=\boldsymbol{S}_\ell^\top v^\ell, \tag{6.10}$$

where

$$x^j=\begin{cases} v^{\ell,k} & \text{if } j=\mathsf{i}(\ell,k) \text{ with } k\in\{1,\ldots,\kappa_\ell\}\\ 0 & \text{otherwise.}\end{cases} \tag{6.11}$$

From a signal processing standpoint, the matrix $\boldsymbol{S}_\ell$ can be viewed as a decimation operator while its transpose is the associated interpolator.

The above definitions allow us to propose the following equivalent formulation of Problem (6.5):

$$\text{Find } \widehat{\boldsymbol{x}}=\underset{\boldsymbol{x}=(x^j)_{1\leqslant j\leqslant J}\in\mathbb{R}^{NJ}}{\operatorname{argmin}} \sum_{j=1}^{J} g_j(A_jx^j)+\sum_{\ell=1}^{L}\iota_{\Lambda_\ell}(\boldsymbol{S}_\ell\,\boldsymbol{x})+\frac{1}{2}\sum_{j=1}^{J}\|x^j-\widetilde{x}\|^2_{\Omega_j}. \tag{6.12}$$

The main difference between formulations (6.5) and (6.12) is the introduction of the term $\sum_{\ell=1}^{L}\iota_{\Lambda_\ell}(\boldsymbol{S}_\ell\,\boldsymbol{x})$. This latter formulation makes the link with Problem (6.3) more explicit. More precisely, in order to solve Problem (6.12) using Algorithm 15, it is necessary to set:

- $J'=J+L$,
- $(\forall\ell\in\{1,\ldots,L\})\quad M_{J+\ell}=N\kappa_\ell$,
- $M=\sum_{j=1}^{J'}M_j$,
- $(\forall j\in\{1,\ldots,J\})\quad \boldsymbol{A}_j=[\underbrace{0\ldots0}_{N(j-1)\times}\ A_j\Omega_j^{-1/2}\ \underbrace{0\ldots0}_{N(J-j)\times}]$,
- $\boldsymbol{D}=\operatorname{Diag}(\Omega_1^{-1/2},\ldots,\Omega_J^{-1/2})$,
- $(\forall\ell\in\{1,\ldots,L\})\ g_{J+\ell}=\iota_{\Lambda_\ell}$ and $\boldsymbol{A}_{J+\ell}=\boldsymbol{S}_\ell\boldsymbol{D}$.

Then, Problem (6.12) is recast as:

$$\text{Find } \widehat{\boldsymbol{x}}=\boldsymbol{D}\widehat{\boldsymbol{x}}' \quad \text{such that}$$
$$\widehat{\boldsymbol{x}}'=\underset{\boldsymbol{x}'\in\mathbb{R}^{NJ}}{\operatorname{argmin}} \sum_{j=1}^{J'} g_j(\boldsymbol{A}_j\boldsymbol{x}')+\frac{1}{2}\|\boldsymbol{x}'-\widetilde{\boldsymbol{x}}'\|^2, \tag{6.13}$$

where $\widetilde{\boldsymbol{x}}'=[\Omega_1^{1/2}\widetilde{x}^\top,\ldots,\Omega_J^{1/2}\widetilde{x}^\top]^\top\in\mathbb{R}^{NJ}$.

6.3.2 Derivation of the proposed algorithm

The application of Algorithm 15 for the resolution of Problem (6.13) yields:

$$
\begin{array}{l}
\left\lfloor \begin{array}{l}
B_j \in \mathbb{R}^{M_j \times M_j} \text{ with } B_j \succeq \boldsymbol{A}_j \boldsymbol{A}_j^\top, \quad j \in \{1, \dots, J'\} \\
\epsilon \in]0, 1] \\
(y_0^j)_{1 \leqslant j \leqslant J'} \in \mathbb{R}^M \\
\boldsymbol{x}'_0 = \widetilde{\boldsymbol{x}}' - \sum_{j=1}^{J'} \boldsymbol{A}_j^\top y_0^j.
\end{array} \right. \\
\text{For } n = 0, 1, \dots \\
\left\lfloor \begin{array}{l}
\gamma_n \in [\epsilon, 2 - \epsilon] \\
j_n \in \{1, \dots, J'\} \\
\widetilde{y}_n^{j_n} = y_n^{j_n} + \gamma_n (B_{j_n})^{-1} \boldsymbol{A}_{j_n} \boldsymbol{x}'_n \\
y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n (B_{j_n})^{-1} \mathrm{prox}_{\gamma_n (B_{j_n})^{-1}, g_{j_n}} \left(\gamma_n^{-1} B_{j_n} \widetilde{y}_n^{j_n} \right) \\
y_{n+1}^{j} = y_n^{j}, \quad j \in \{1, \dots, J'\} \setminus \{j_n\} \\
\boldsymbol{x}'_{n+1} = \boldsymbol{x}'_n - \boldsymbol{A}_{j_n}^\top (y_{n+1}^{j_n} - y_n^{j_n}).
\end{array} \right.
\end{array}
\tag{6.14}
$$

Let us now show that the above algorithm can be simplified.
First, note that $(\forall j \in \{1, \dots, J\})\ \boldsymbol{A}_j \boldsymbol{A}_j^\top = A_j \Omega_j^{-1} A_j^\top$ and $(\forall \ell \in \{1, \dots, L\})$ $\|\boldsymbol{A}_{J+\ell}\| = \|\boldsymbol{S}_\ell \boldsymbol{D}\| = \max_{j \in \mathbb{V}_\ell} \|\Omega_j^{-1/2}\|$. It can also be observed that

$$
(\forall \ell \in \{1, \dots, L\})\ (\forall \gamma \in\ (0, +\infty)) \qquad \mathrm{prox}_{\gamma^{-1} g_{J+\ell}}(\gamma^{-1} \cdot) = \gamma^{-1} \Pi_{\Lambda_\ell}, \tag{6.15}
$$

where Π_{Λ_ℓ} is the linear projector onto the vector space Λ_ℓ.
Hence, by setting

$$
(\forall \ell \in \{1, \dots, L\}) \qquad B_{J+\ell} = \vartheta_\ell^{-1} \mathrm{Id}_{N \kappa_\ell} \tag{6.16}
$$

with $\vartheta_\ell = \min_{j \in \mathbb{V}_\ell} \|\Omega_j\|$, and

$$
(\forall j \in \{1, \dots, J\}) \quad \mathbb{V}_j^* = \big\{ (\ell, k) \;\big|\; \ell \in \{1, \dots, L\},\, k \in \{1, \dots, \kappa_\ell\} \text{ and } \mathsf{i}(\ell, k) = j \big\}, \tag{6.17}
$$

Algorithm (6.14) can be re-expressed as

$$
\begin{array}{l}
\left\lfloor \begin{array}{l}
B_j \in \mathbb{R}^{M_j \times M_j} \text{ with } B_j \succeq A_j \Omega_j^{-1} A_j^\top, \quad j \in \{1, \dots, J\} \\
\vartheta_\ell = \min\limits_{j \in \mathbb{V}_\ell} \|\Omega_j\|, \quad \ell \in \{1, \dots, L\} \\
\epsilon \in]0, 1] \\
z_0^\ell \in \mathbb{R}^{N\kappa_\ell}, \quad \ell \in \{1, \dots, L\} \\
y_0^j \in \mathbb{R}^{M_j}, x_0^j = \widetilde{x} - \Omega_j^{-1} \Big(A_j^\top y_0^j + \sum\limits_{(\ell,k) \in \mathbb{V}_j^*} z_0^{\ell,k} \Big), \quad j \in \{1, \dots, J\}.
\end{array} \right. \\
\text{For } n = 0, 1, \dots \\
\left\lfloor \begin{array}{l}
\gamma_n \in [\epsilon, 2 - \epsilon] \\
j_n \in \{1, \dots, J + L\} \\
\text{If } j_n \leqslant J \\
\left\lfloor \begin{array}{l}
\widetilde{y}_n^{j_n} = y_n^{j_n} + \gamma_n (B_{j_n})^{-1} A_{j_n} x_n^{j_n} \\
y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n (B_{j_n})^{-1} \mathrm{prox}_{\gamma_n (B_{j_n})^{-1}, g_{j_n}} \big(\gamma_n^{-1} B_{j_n} \widetilde{y}_n^{j_n} \big) \\
y_{n+1}^j = y_n^j, \quad j \in \{1, \dots, J\} \setminus \{j_n\} \\
z_{n+1}^\ell = z_n^\ell, \quad \ell \in \{1, \dots, L\} \\
x_{n+1}^{j_n} = x_n^{j_n} - \Omega_{j_n}^{-1} A_{j_n}^\top (y_{n+1}^{j_n} - y_n^{j_n}) \\
x_{n+1}^j = x_n^j, \quad j \in \{1, \dots, J\} \setminus \{j_n\}
\end{array} \right. \\
\text{else} \\
\left\lfloor \begin{array}{l}
\ell_n = j_n - J \\
\widetilde{z}_n^{\ell_n} = z_n^{\ell_n} + \gamma_n \vartheta_{\ell_n} (x_n^j)_{j \in \mathbb{V}_{\ell_n}} \\
z_{n+1}^{\ell_n} = \widetilde{z}_n^{\ell_n} - \Pi_{\Lambda_{\ell_n}} (\widetilde{z}_n^{\ell_n}) \\
z_{n+1}^\ell = z_n^\ell, \quad \ell \in \{1, \dots, L\} \setminus \{\ell_n\} \\
y_{n+1}^j = y_n^j, \quad j \in \{1, \dots, J\} \\
\text{For } k = 1, \dots, \kappa_{\ell_n} \\
\left\lfloor \; x_{n+1}^{\mathrm{i}(\ell_n,k)} = x_n^{\mathrm{i}(\ell_n,k)} - \Omega_{\mathrm{i}(\ell_n,k)}^{-1} (z_{n+1}^{\ell_n,k} - z_n^{\ell_n,k}) \right. \\
x_{n+1}^j = x_n^j, \quad j \notin \mathbb{V}_{\ell_n}.
\end{array} \right.
\end{array} \tag{6.18}
$$

Hereabove, for more readibility, we have set, for every $n \in \mathbb{N}$,

$$
\begin{aligned}
\boldsymbol{x}_n &= [(x_n^1)^\top, \dots, (x_n^J)^\top]^\top = \boldsymbol{D} \boldsymbol{x}'_n, \\
z_n^\ell &= y_n^{J+\ell} \in \mathbb{R}^{N\kappa_\ell}, \\
\widetilde{z}_n^\ell &= \widetilde{y}_n^{J+\ell} \in \mathbb{R}^{N\kappa_\ell}.
\end{aligned}
$$

Furthermore, it can be noticed that, for every $n \in \mathbb{N}$ such that $j_n = J + \ell_n > J$,

$$
\Pi_{\Lambda_{\ell_n}} (z_{n+1}^{\ell_n}) = \Pi_{\Lambda_{\ell_n}} \big(\widetilde{z}_n^{\ell_n} - \Pi_{\Lambda_{\ell_n}} (\widetilde{z}_n^{\ell_n}) \big) = \Pi_{\Lambda_{\ell_n}} \big(\widetilde{z}_n^{\ell_n} \big) - \Pi_{\Lambda_{\ell_n}} \big(\Pi_{\Lambda_{\ell_n}} (\widetilde{z}_n^{\ell_n}) \big) = 0. \tag{6.19}
$$

Since, for every $\ell \in \{1, \dots, L\} \setminus \{\ell_n\}$,

$$
z_{n+1}^\ell = z_n^\ell, \tag{6.20}
$$

Property (6.19) can be extended by induction to

$$(\forall n \in \mathbb{N})(\forall \ell \in \{1, \dots, L\}) \quad \Pi_{\Lambda_\ell}(z_n^\ell) = 0 \tag{6.21}$$

by an appropriate initialization of the algorithm (e.g., by choosing $(\forall \ell \in \{1, \dots, L\})$ $z_0^\ell = 0$). Hence, for every $n \in \mathbb{N}$ such that $j_n = J + \ell_n > J$,

$$\Pi_{\Lambda_{\ell_n}}(\widetilde{z}_n^\ell) = \gamma_n \vartheta_{\ell_n} \Pi_{\Lambda_{\ell_n}}\big((x_n^j)_{j \in \mathbb{V}_{\ell_n}}\big), \tag{6.22}$$

which implies that

$$z_{n+1}^{\ell_n} - z_n^{\ell_n} = \gamma_n \vartheta_{\ell_n}\big((x_n^j)_{j \in \mathbb{V}_{\ell_n}} - \Pi_{\Lambda_{\ell_n}}\big((x_n^j)_{j \in \mathbb{V}_{\ell_n}}\big)\big). \tag{6.23}$$

The second part of Algorithm (6.18) dealing with the case when $j_n > J$ can then be reexpressed as described in lines 18 to 24 of Algorithm 20. It can be observed that, in the resulting algorithm, we were able to drop the variables $(z_n^\ell)_{1 \leqslant \ell \leqslant L}$, for every $n \in \mathbb{N}$.

Algorithm 20 Distributed Preconditioned Dual Forward-Backward

1: **Initialization:**
2: $\mathbb{V}_\ell \equiv$ index set of nodes in hyperedge $\ell \in \{1, \dots, L\}$
3: $B_j \in \mathbb{R}^{M_j \times M_j}$ with $B_j \succeq A_j \Omega_j^{-1} A_j^\top, \quad j \in \{1, \dots, J\}$
4: $\vartheta_\ell = \min_{j \in \mathbb{V}_\ell} \|\Omega_j\|, \quad \ell \in \{1, \dots, L\}$
5: $\epsilon \in]0, 1]$
6: $y_0^j \in \mathbb{R}^{M_j}, x_0^j = \widetilde{x} - \Omega_j^{-1} A_j^\top y_0^j, \quad j \in \{1, \dots, J\}$.
7: **Main loop:**
8: **for** $n = 0, 1, \dots$ **do**
9: $\quad \gamma_n \in [\epsilon, 2 - \epsilon]$
10: $\quad j_n \in \{1, \dots, J + L\}$
11: $\quad$ **if** $j_n \leqslant J$ **then** *Local optimization*
12: $\quad\quad \widetilde{y}_n^{j_n} = y_n^{j_n} + \gamma_n (B_{j_n})^{-1} A_{j_n} x_n^{j_n}$
13: $\quad\quad y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n (B_{j_n})^{-1} \mathrm{prox}_{\gamma_n (B_{j_n})^{-1}, g_{j_n}}\big(\gamma_n^{-1} B_{j_n} \widetilde{y}_n^{j_n}\big)$
14: $\quad\quad y_{n+1}^j = y_n^j, \quad j \in \{1, \dots, J\} \setminus \{j_n\}$
15: $\quad\quad x_{n+1}^{j_n} = x_n^{j_n} - \Omega_{j_n}^{-1} A_{j_n}^\top (y_{n+1}^{j_n} - y_n^{j_n})$
16: $\quad\quad x_{n+1}^j = x_n^j, \quad j \in \{1, \dots, J\} \setminus \{j_n\}$
17: $\quad$ **else** *Projection*
18: $\quad\quad \ell_n = j_n - J$
19: $\quad\quad y_{n+1}^j = y_n^j, \quad j \in \{1, \dots, J\}$
20: $\quad\quad p_n^{\ell_n} = \Pi_{\Lambda_{\ell_n}}\big((x_n^j)_{j \in \mathbb{V}_{\ell_n}}\big)$
21: $\quad\quad$ **for** $k = 1, \dots, \kappa_{\ell_n}$ **do**
22: $\quad\quad\quad x_{n+1}^{\mathsf{i}(\ell_n, k)} = x_n^{\mathsf{i}(\ell_n, k)} + \gamma_n \vartheta_{\ell_n} \Omega_{\mathsf{i}(\ell_n, k)}^{-1} (p_n^{\ell_n, k} - x_n^{\mathsf{i}(\ell_n, k)})$
23: $\quad\quad$ **end for**
24: $\quad\quad x_{n+1}^j = x_n^j, \quad j \notin \mathbb{V}_{\ell_n}$.
25: $\quad$ **end if**
26: **end for**

One can notice that the body of Algorithm 20 is composed of two main parts:

- First a local optimization part (lines 12 to 16) which is reminiscent of the Dual Block Forward-Backward algorithm where, at each iteration, a block j_n is selected and the associated dual and primal variables $y_n^{j_n}$ (line 13) and $x_n^{j_n}$ (line 15) are updated, respectively. Note that the main difference between the proposed algorithm and Algorithm 15 lies in the fact that each block j_n is now associated with a local primal variable $x_n^{j_n}$ whereas, in Algorithm 15, x_n was a shared variable.
- The second part of Algorithm 20 is a projection step (lines 18 to 24) in which a set $\mathbb{V}_{\ell_n}$ is selected and all the variables $(x^{j_n})_{j_n \in \mathbb{V}_{\ell_n}}$ are updated by means of a projection operating over the selected set $\mathbb{V}_{\ell_n}$.

In Algorithm 20, all computation steps only involve local variables, which is beneficial for parallel processing. Moreover, a high flexibility is allowed by adopting a quasi-cyclic rule for choosing the indices j_n and ℓ_n at each iteration n. The distributed Algorithm 20 inherits all the advantages of primal-dual methods, in particular it requires no inversion of the matrices $(A_j)_{1 \leqslant j \leqslant J}$, which is of main interest when these matrices do not have a simple structure and are of very large size. Note that the proposed approach is quite different from the ones developed in [Pesquet and Repetti, 2015; Richtárik and Takác, 2016] since we do not assume a random sweeping rule for the block index selection, and our convergence analysis does not rely on nonexpansiveness properties of the involved operators.

6.3.3 Consensus choice

Generic case

When the operators $(A_j)_{1 \leqslant j \leqslant J}$ have no specific structure, a natural choice for the vector spaces $(\Lambda_\ell)_{1 \leqslant \ell \leqslant L}$ is to adopt a form similar to that of Λ in (6.6):

$$(\forall \ell \in \{1, \ldots, L\}) \qquad \Lambda_\ell = \left\{ \begin{bmatrix} v^{\ell,1} \\ \vdots \\ v^{\ell,\kappa_\ell} \end{bmatrix} \in \mathbb{R}^{N\kappa_\ell} \mid v^{\ell,1} = \ldots = v^{\ell,\kappa_\ell} \right\}. \tag{6.24}$$

Note that (6.8) and (6.24) imply that the hypergraph induced by the hyperedges $(\mathbb{V}_\ell)_{1 \leqslant \ell \leqslant L}$ is connected (Figure 6.1 is an example of such a connected hypergraph). In this context, the connectivity of the hypergraph is essential in order to allow the global consensus solution to be reached.

For every $\ell \in \{1, \ldots, L\}$, the projection onto Λ_ℓ is then simply expressed as

$$\big(\forall (v^{\ell,k})_{1 \leqslant k \leqslant \kappa_\ell} \in \mathbb{R}^{N\kappa_\ell}\big) \qquad \Pi_{\Lambda_\ell}\big((v^{\ell,k})_{1 \leqslant k \leqslant \kappa_\ell}\big) = [(\overline{v}^\ell)^\top, \ldots, (\overline{v}^\ell)^\top]^\top, \tag{6.25}$$

where

$$\overline{v}^{\ell} = \mathrm{mean}\left((v^{\ell,k})_{1\leqslant k\leqslant \kappa_\ell}\right) \tag{6.26}$$

and $\mathrm{mean}(\cdot)$ designates the arithmetic mean operation (i.e., $\mathrm{mean}\left((v^{\ell,k})_{1\leqslant k\leqslant \kappa_\ell}\right) = \kappa_\ell^{-1}\sum_{k=1}^{\kappa_\ell} v^{\ell,k}$).

In addition, Condition (6.7) is met by simply choosing $(\forall j \in \{1,\ldots,J\})\ \Omega_j = \omega_j \mathrm{Id}_N$, where $(\omega_j)_{1\leqslant j\leqslant J} \in]0,1]^J$ are such that $\sum_{j=1}^J \omega_j = 1$

These simplification lead to the following modifications of lines 20-23 in Algorithm 20:

$$\begin{array}{l} \overline{x}_n^{\ell_n} = \mathrm{mean}\left((x_n^j)_{j\in\mathbb{V}_{\ell_n}}\right) \\ \text{For } k = 1,\ldots,\kappa_{\ell_n} \\ \left\lfloor\ x_{n+1}^{\mathrm{i}(\ell_n,k)} = x_n^{\mathrm{i}(\ell_n,k)} + \gamma_n \vartheta_{\ell_n} \omega_{\mathrm{i}(\ell_n,k)}^{-1} (\overline{x}_n^{\ell_n} - x_n^{\mathrm{i}(\ell_n,k)}). \end{array} \tag{6.27}$$

Dimension reduction

Under its previous form, Algorithm 20 requires each node of the hypergraph to handle a local copy of several variables. In particular, for the j-th node, a vector x_n^j of dimension N has to be stored, which may be prohibitive for highly dimensional problems. However, very often in image processing problems, the operator $(A_j)_{1\leqslant j\leqslant J}$ has a sparse block structure, which makes it possible to alleviate this problem. More specifically, it will be assumed subsequently that

$$(\forall j \in \{1,\ldots,J\})\big(\forall x^j = ([x^j]_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^N\big) \qquad A_j x^j = \sum_{t\in\mathbb{T}_j} \mathcal{A}_{j,t}[x^j]_t, \tag{6.28}$$

where, for every $j \in \{1,\ldots,J\}$, $[x^j]_t$ is a vector corresponding to a block of data of dimension L, T is the overall number of blocks (i.e., $N = TL$), and $\mathbb{T}_j \subset \{1,\ldots,T\}$ defines the reduced index subset of the components of vector x^j acting on the operator A_j. In the above equation, $(\mathcal{A}_{j,t})_{t\in\mathbb{T}_j}$ are the associated reduced-size matrices of dimensions $M_j \times L$. Similarly to the way x^j has been block-decomposed, let us split the diagonal matrix Ω_j as

$$\Omega_j = \mathrm{Diag}\left(\Omega_{j,1},\ldots,\Omega_{j,T}\right) \tag{6.29}$$

where, for every $t \in \{1,\ldots,T\}$, $\Omega_{j,t}$ is a diagonal matrix of size $L \times L$. It then obviously holds that $A_j\Omega_j^{-1}A_j^\top = \sum_{t\in\mathbb{T}_j} \mathcal{A}_{j,t}\Omega_{j,t}^{-1}\mathcal{A}_{j,t}^\top$. To avoid degenerate cases, we will subsequently assume that $(\forall j \in \{1,\ldots,J\})\ \mathbb{T}_j \neq \varnothing$ and $\bigcup_{j=1}^J \mathbb{T}_j = \{1,\ldots,T\}$.

In our distributed formulation, the specific form of the operators $(A_j)_{1\leqslant j\leqslant J}$ suggests us to set the vector subspaces $(\Lambda_\ell)_{1\leqslant \ell\leqslant L}$ so as to reach the consensus only for the components $([x^j]_t)_{1\leqslant j\leqslant J, t\in\mathbb{T}_j}$ of vectors $(x^j)_{1\leqslant j\leqslant J}$. This means that the space Λ

(resp. Λ_ℓ with $\ell \in \{1, \dots, L\}$) is defined as

$$(x^j)_{1 \leqslant j \leqslant J} \in \Lambda \quad \Leftrightarrow \quad (\forall (j, j') \in \{1, \dots, J\}^2)(\forall t \in \mathbb{T}_j \cap \mathbb{T}_{j'}) \quad [x^j]_t = [x^{j'}]_t \tag{6.30}$$

$$(\text{resp. } (x^j)_{j \in \mathbb{V}_\ell} \in \Lambda_\ell \quad \Leftrightarrow \quad (\forall (j, j') \in \mathbb{V}_\ell^2)(\forall t \in \mathbb{T}_j \cap \mathbb{T}_{j'}) \quad [x^j]_t = [x^{j'}]_t).$$

It can be noticed that, although the hypergraph must still be built so that (6.8) holds, Λ is no longer given by (6.6), since the components $([x^j]_t)_{1 \leqslant j \leqslant J, t \notin \mathbb{T}_j}$ are unconstrained. The main interest of this choice is that Problem (6.5) then decouples into two optimization problems:

- the minimization of the function

$$([x^j]_t)_{1 \leqslant j \leqslant J, t \in \mathbb{T}_j} \mapsto \sum_{j=1}^{J} g_j\Big(\sum_{t \in \mathbb{T}_j} \mathcal{A}_{j,t}[x^j]_t\Big) + \frac{1}{2} \sum_{j=1}^{J} \sum_{t \in \mathbb{T}_j} \|[x^j]_t - [\widetilde{x}]_t\|^2_{\Omega_{j,t}} \tag{6.31}$$

  subject to Constraint (6.30);

- the unconstrained minimization of the function

$$([x^j]_t)_{1 \leqslant j \leqslant J, t \notin \mathbb{T}_j} \mapsto \sum_{j=1}^{J} \sum_{t \notin \mathbb{T}_j} \|[x^j]_t - [\widetilde{x}]_t\|^2_{\Omega_{j,t}}. \tag{6.32}$$

Since the second problem is trivial, the variables $([x_n^j]_t)_{1 \leqslant j \leqslant J, t \notin \mathbb{T}_j}$ generated at each iteration $n \in \mathbb{N}$ of Algorithm 20 are useless and, consequently, they can be discarded. By doing so, only $|\mathbb{T}_j|$ vectors[1] $([x_n^j]_t)_{t \in \mathbb{T}_j}$ of dimension L need to be stored at the j-th node (instead of T vectors of this size) and the number of computations to be performed during the projection step are also diminished.

This yields Algorithm 21 where, in the synchronization step, averaging operations corresponding to the projection onto Λ_{ℓ_n} have been substituted for lines 20-23 in Algorithm 20. The notation

$$(\forall t \in \{1, \dots, T\}) \qquad \mathbb{T}_t^* = \big\{ j \in \{1, \dots, J\} \;\big|\; t \in \mathbb{T}_j \big\}, \tag{6.33}$$

has been introduced for the computation of the averages. In particular, in line 27 of Algorithm 21, if $\mathbb{V}_\ell \cap \mathbb{T}_t^*$ is a singleton, which means that the t-th block component of the vector x appears only once in the expression of $g_j(A_j x)$ for indices j in the ℓ_n-th hyperedge, then the averaging reduces to setting $[x_{n+1}^j]_t = [x_n^j]_t$. It is also worthwhile to note that, when $(\forall j \in \{1, \dots, J\})\ \mathbb{T}_j = \{1, \dots, T\}$, the consensus solution described in Section 6.3.3 is recovered. It must be however pointed out that, in general, to have the equivalence between the minimization of (6.31) subject

[1] $|S|$ is the cardinality of a set S.

to Constraint (6.30) and the resolution of Problem (6.3), the following condition has to be substituted for (6.7):

$$(\forall t \in \{1,\dots,T\}) \qquad \sum_{j\in\mathbb{T}_t^*} \Omega_{j,t} = \mathrm{Id}_L. \tag{6.34}$$

In Algorithm 21, this has been simply achieved by setting $(\forall j \in \{1,\dots,J\})\ (\forall t \in \mathbb{T}_j)$ $\Omega_{j,t} = \omega_{j,t}\mathrm{Id}_L$, where $(\omega_{j,t})_{1\leqslant j\leqslant J, t\in\mathbb{T}_j}$ are positive real such that $(\forall t \in \{1,\dots,T\})$ $\sum_{j\in\mathbb{T}_t^*} \omega_{j,t} = 1$. In turn, the notation $(\Omega_{j,t})_{1\leqslant j\leqslant J, t\notin\mathbb{T}_j}$ is no longer used in this algorithm.

Although Algorithm 21 can give rise to a variety of distributed implementations, we will focus on a simpler instance of this algorithm in the remainder of this chapter.

6.4 A useful special case

Let us consider the case when $C \leqslant J$ processing units are available. In the remainder of the chapter, to simplify our presentation, we will restrict our attention to a case of practical interest for the video application described in Section 6.5 by making the following assumptions.

Assumption 6.1

1. *The hyperedges* $(\mathbb{V}_\ell)_{1\leqslant\ell\leqslant C}$ *form a partition of* $\{1,\dots,J\}$.
2. *For every* $\ell \in \{1,\dots,C\}$, *let* $\mathbb{T}_{\mathbb{V}_\ell} = \bigcup_{j\in\mathbb{V}_\ell} \mathbb{T}_j$.
   (a) *For every* $(\ell,\ell') \in \{1,\dots,C\}^2$, $\mathbb{T}_{\mathbb{V}_\ell} \cap \mathbb{T}_{\mathbb{V}_{\ell'}} = \varnothing$ *if* $|\ell-\ell'| > 1$.
   (b) *For every* $\ell \in \{2,\dots,C-1\}$, $\mathbb{T}_{\mathbb{V}_{\ell-1}} \cap \mathbb{T}_{\mathbb{V}_\ell} \cap \mathbb{T}_{\mathbb{V}_{\ell+1}} = \varnothing$.

An example of hypergraph satisfying Assumption 6.11 is displayed in Figure 6.2. For every $\ell \in \{1,\dots,C\}$, $\mathbb{T}_{\mathbb{V}_\ell}$ is the set of the block indices t of the components $[x^j]_t$ where j is any node in $\mathbb{V}_\ell$. According to Assumption 6.1-2a, these indices may only be common to hyperedges having preceding or following index values (i.e., $\ell-1$ or $\ell+1$). Finally, Assumption 6.1-2b means that no overlap is allowed between block indices shared with the preceding hyperedge and the following one.

Algorithm 21 Distributed Preconditioned Dual Forward-Backward after Dimension Reduction

Initialization:
$\mathbb{V}_\ell \equiv$ index set of nodes in hyperedge $\ell \in \{1,\dots,L\}$
$\mathbb{T}_j \equiv$ index set of blocks used at node $j \in \{1,\dots,J\}$
$\mathbb{T}_t^* \equiv$ index set of nodes using block $t \in \{1,\dots,T\}$
$\{\omega_{j,t} \mid 1 \leqslant j \leqslant J, t \in \mathbb{T}_j\} \subset]0,1]$ such that $(\forall t \in \{1,\dots,T\}) \sum_{j \in \mathbb{T}_t^*} \omega_{j,t} = 1$
$B_j \in \mathbb{R}^{M_j \times M_j}$ with $B_j \succeq \sum_{t \in \mathbb{T}_j} \omega_{j,t}^{-1} \mathcal{A}_{j,t} \mathcal{A}_{j,t}^\top, \quad j \in \{1,\dots,J\}$
$\vartheta_\ell = \min_{j \in \mathbb{V}_\ell, t \in \mathbb{T}_j} \omega_{j,t}, \quad \ell \in \{1,\dots,L\}$
$\epsilon \in]0,1]$
$y_0^j \in \mathbb{R}^{M_j}, [x_0^j]_t = [\widetilde{x}]_t - \omega_{j,t}^{-1} \mathcal{A}_{j,t}^\top y_0^j, \quad j \in \{1,\dots,J\}, t \in \mathbb{T}_j.$
Main loop:
for $n = 0, 1, \dots$ **do**
  $\gamma_n \in [\epsilon, 2-\epsilon]$
  $j_n \in \{1,\dots,J+L\}$
  if $j_n \leqslant J$ **then** *Local optimization*
    $\widetilde{y}_n^{j_n} = y_n^{j_n} + \gamma_n B_{j_n}^{-1} \sum_{t \in \mathbb{T}_{j_n}} \mathcal{A}_{j_n,t} [x_n^{j_n}]_t$
    $y_{n+1}^{j_n} = \widetilde{y}_n^{j_n} - \gamma_n B_{j_n}^{-1} \mathrm{prox}_{\gamma_n B_{j_n}^{-1}, g_{j_n}} \big(\gamma_n^{-1} B_{j_n} \widetilde{y}_n^{j_n}\big)$
    $y_{n+1}^j = y_n^j, \quad j \in \{1,\dots,J\} \setminus \{j_n\}$
    for $t \in \mathbb{T}_{j_n}$ **do**
      $[x_{n+1}^{j_n}]_t = [x_n^{j_n}]_t - \omega_{j_n,t}^{-1} \mathcal{A}_{j_n,t}^\top (y_{n+1}^{j_n} - y_n^{j_n})$
    end for
    $([x_{n+1}^j]_t)_{t \in \mathbb{T}_j} = ([x_n^j]_t)_{t \in \mathbb{T}_j}, \quad j \in \{1,\dots,J\} \setminus \{j_n\}$
  else *Projection*
    $\ell_n = j_n - J$
    $y_{n+1}^j = y_n^j, \quad j \in \{1,\dots,J\}$
    for $j \in \mathbb{V}_{\ell_n}$ **do**
      for $t \in \mathbb{T}_j$ **do**
        $[x_{n+1}^j]_t = [x_n^j]_t + \gamma_n \vartheta_{\ell_n} \omega_{j,t}^{-1} \big(\mathrm{mean}\, \big(([x_n^{j'}]_t)_{j' \in \mathbb{V}_{\ell_n} \cap \mathbb{T}_t^*} \big) - [x_n^j]_t \big)$
      end for
    end for
    $([x_{n+1}^j]_t)_{t \in \mathbb{T}_j} = ([x_n^j]_t)_{t \in \mathbb{T}_j}, \quad j \notin \mathbb{V}_{\ell_n}.$
  end if
end for

6.4.1 Form of the algorithm

An interesting instance of Algorithm 21 is then obtained by setting $L = C + 1$ and by assuming that each hyperedge $\mathbb{V}_\ell$ with $\ell \in \{1, \ldots, C\}$ corresponds to a given computing unit where the computations are locally synchronized. In addition, hyperedge $\mathbb{V}_L$ is set to $\{1, \ldots, J\}$ in order to model global synchronization steps consisting of an averaging over all the available nodes. At each iteration n, only a subset $\mathbb{J}_{n,\ell}$ of dual variable indices is activated within the ℓ-th hyperedge and their update is followed by either a possible local synchronization or a global one.

Algorithm 22 summarizes the proposed approach. For simplicity, the index L has been dropped in variable ϑ_L. Note that, if the local synchronization step is omitted (by setting $[x^j_{n+1}]_t = [x^j_{n+1/2}]_t$ in line 27), the algorithm still makes sense since it can be easily shown that it actually corresponds to a rewriting of Algorithm 21 in the case when $L = 1$ and $\mathbb{V}_1 = \{1, \ldots, J\}$. Unlikely, the global synchronization is mandatory although it has not to be performed at each iteration but only in a quasi-cyclic manner.

It should be emphasized that even in the case when all the dual variables are updated iteratively (i.e., $(\forall \ell \in \{1, \ldots, L\})$ $(\forall n \in \mathbb{N})$ $\mathbb{J}_{n,\ell} = \mathbb{V}_\ell$), Algorithm 22 exhibits a different structure from the one of the parallel dual forward-backward algorithm in [Combettes et al., 2011].

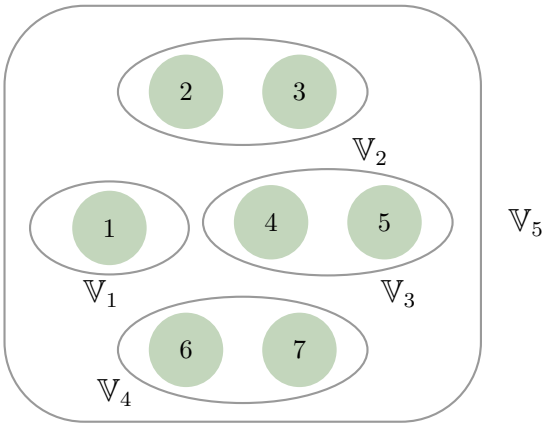


Figure 6.2: *Hypergraph of $J = 7$ nodes, $C = 4$ computing units and $L = 5$ hyperedges.*

6.4.2 Distributed implementation

Let us now look more precisely at the implementation of Algorithm 22 on a distributed architecture having $C \in \mathbb{N}^*$ computing units, each computing unit being indexed by an integer $c \in \{1, \ldots, C\}$. Figure 6.3 shows an illustrative example of $C = 4$ computing units based on the hypergraph defined in Figure 6.2.

Algorithm 22 Special case of distributed Preconditioned Dual Forward-Backward

Initialization:
$\mathbb{V}_\ell \equiv$ index set of nodes associated with computing unit $\ell \in \{1,\dots,C\}$
$\mathbb{T}_j \equiv$ index set of blocks used at node $j \in \{1,\dots,J\}$
$\mathbb{T}_t^* \equiv$ index set of nodes using block $t \in \{1,\dots,T\}$
$\{\omega_{j,t} \mid 1 \leqslant j \leqslant J, t \in \mathbb{T}_j\} \subset]0,1]$ such that $(\forall t \in \{1,\dots,T\}) \sum_{j \in \mathbb{T}_t^*} \omega_{j,t} = 1$
$B_j \in \mathbb{R}^{M_j \times M_j}$ with $B_j \succeq \sum_{t \in \mathbb{T}_j} \omega_{j,t}^{-1} \mathcal{A}_{j,t} \mathcal{A}_{j,t}^\top, \quad j \in \{1,\dots,J\}$
$\vartheta = \min_{1 \leqslant j \leqslant J, 1 \leqslant t \leqslant T} \omega_{j,t}, \quad \vartheta_\ell = \min_{j \in \mathbb{V}_\ell, t \in \mathbb{T}_j} \omega_{j,t}, \ \ \ell \in \{1,\dots,C\}$
$\epsilon \in]0,1]$
$y_0^j \in \mathbb{R}^{M_j}, [x_0^j]_t = [\widetilde{x}]_t - \omega_{j,t}^{-1} \mathcal{A}_{j,t}^\top y_0^j, \quad j \in \{1,\dots,J\}, t \in \mathbb{T}_j.$
for $n = 0, 1, \dots$ **do** *Main loop*
  for $\ell = 1,\dots,C$ **do**
    $\mathbb{J}_{n,\ell} \subset \mathbb{V}_\ell$
    for $j \in \mathbb{J}_{n,\ell}$ **do** *Local optimization*
      $\widetilde{y}_n^j = y_n^j + \gamma_n B_j^{-1} \sum_{t \in \mathbb{T}_j} \mathcal{A}_{j,t} [x_n^j]_t$
      $y_{n+1}^j = \widetilde{y}_n^j - \gamma_n B_j^{-1} \mathrm{prox}_{\gamma_n B_j^{-1}, g_j}\big(\gamma_n^{-1} B_j \widetilde{y}_n^j\big)$
      for $t \in \mathbb{T}_j$ **do**
        $[x_{n+1/2}^j]_t = [x_n^j]_t - \omega_{j,t}^{-1} \mathcal{A}_{j,t}^\top (y_{n+1}^j - y_n^j)$
      end for
    end for
    for $j \in \mathbb{V}_\ell \setminus \mathbb{J}_{n,\ell}$ **do**
      $y_{n+1}^j = y_n^j$
      $([x_{n+1/2}^j]_t)_{t \in \mathbb{T}_j} = ([x_n^j]_t)_{t \in \mathbb{T}_j}$
    end for
    if local synchronization is requested **then**
      for $j \in \mathbb{V}_\ell$ **do**
        for $t \in \mathbb{T}_j$ **do**
          $[x_{n+1}^j]_t = [x_{n+1/2}^j]_t + \gamma_n \vartheta_\ell \omega_{j,t}^{-1} \big(\mathrm{mean}\big(([x_{n+1/2}^{j'}]_t)_{j' \in \mathbb{V}_\ell \cap \mathbb{T}_t^*}\big) - [x_{n+1/2}^j]_t \big)$
        end for
      end for
    end if
  end for
  if global synchronization is requested **then**
    for $t = 1,\dots,T$ **do** $[\overline{x}_n]_t = \mathrm{mean}\big(([x_{n+1/2}^j]_t)_{j \in \mathbb{T}_t^*}\big)$;
    for $j = 1,\dots,J$ **do**
      for $t \in \mathbb{T}_j$ **do**
        $[x_{n+1}^j]_t = [x_{n+1/2}^j]_t + \gamma_n \vartheta \omega_{j,t}^{-1} ([\overline{x}_n]_t - [x_{n+1/2}^j]_t)$
      end for
    end for
  end if
end for

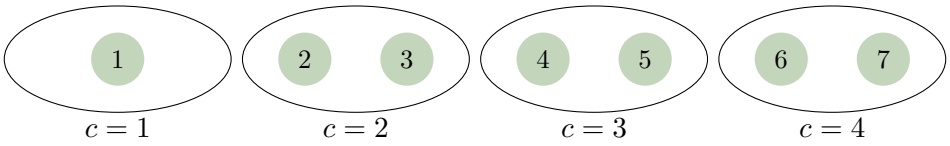


Figure 6.3: *Partitioning of $J = 7$ nodes and $L = 5$ hyperedge on $C = 4$ computing units.*

As we have seen, each computing unit $c \in \{1, \ldots, C\}$ handles κ_c terms corresponding to the nodes in $\mathbb{V}_c$ of the hypergraph, and processes the functions $(g_j)_{j \in \mathbb{V}_c}$ associated with these nodes. Furthermore, a global synchronization step needs to be performed. This task could be assigned to one of the computing unit, say the first one, as modelled in Figure 6.4 by adding a fictitious term corresponding to hyperedge $\mathbb{V}_{C+1}$. This would however lead to a centralized scheme where the computing load between the different units could be unbalanced.

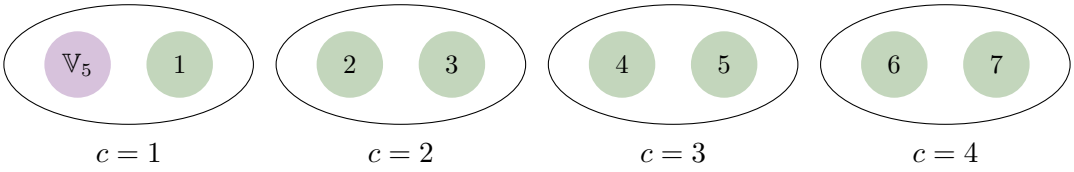


Figure 6.4: *Partitioning of $J = 7$ nodes and $L = 1$ hyperedge on $C = 4$ computing systems.*

A better strategy consists of distributing the operations performed on line 33 of Algorithm 22 over the different computing units. For this purpose, let us first note that at iteration n, the c-th computing unit only needs the block components $([\overline{x}_n]_t)_{t \in \mathbb{T}_{\mathbb{V}_c}}$. In addition, because of Assumption 6.1-2a, some of these variables may be shared with the computing units $c-1$ (if $c \neq 1$) and $c+1$ (if $c \neq C$), where part of the variables $[x^j_{n+1/2}]_t$ necessary to compute the averages are locally available. As a consequence of Assumption 6.1-2b, no other variables than those available in either $\mathbb{T}_{\mathbb{V}_{c-1}} \cap \mathbb{T}_{\mathbb{V}_c}$ or $\mathbb{T}_{\mathbb{V}_c} \cap \mathbb{T}_{\mathbb{V}_{c+1}}$ are necessary . For example, if $c \neq 1$ and $t \in \mathbb{T}_{\mathbb{V}_{c-1}} \cap \mathbb{T}_{\mathbb{V}_c}$, the averaging operation reads

$$\begin{aligned} [\overline{x}_n]_t &= \frac{1}{|\mathbb{T}^*_t|} \sum_{j \in \mathbb{T}^*_t} [x^j_{n+1/2}]_t \\ &= \frac{1}{|\mathbb{T}^*_t|} \big([s_{n,c-1}]_t + [s_{n,c}]_t \big), \end{aligned} \tag{6.35}$$

where

$$[s_{n,c-1}]_t = \sum_{j \in \mathbb{V}_{c-1} \cap \mathbb{T}^*_t} [x^j_{n+1/2}]_t, \tag{6.36}$$

and $[s_{n,c}]_t$ is similarly defined. Since the variables $([x^j_{n+1/2}]_t)_{j\in\mathbb{V}_{c-1}\cap\mathbb{T}^*_t}$ are not available at unit c, the latter summation has to be performed by unit $c-1$ and the result has to be transmitted to unit c. This one will then be able to compute $[\overline{x}_n]_t$, so as to update variables $([x^j_{n+1}]_t)_{j\in\mathbb{V}_c\cap\mathbb{T}^*_t}$. Besides, $[\overline{x}_n]_t$ will be sent to unit $c-1$, which in turn will update their variables $([x^j_{n+1}]_t)_{j\in\mathbb{V}_{c-1}\cap\mathbb{T}^*_t}$. A similar synchronization process can be followed for blocks with indices $t\in\mathbb{T}_{\mathbb{V}_c}\cap\mathbb{T}_{\mathbb{V}_{c+1}}$ with $c\neq C$. Finally, for the block indices t in $\mathbb{T}_{\mathbb{V}_c}$ which do not belong to $\mathbb{T}_{\mathbb{V}_{c-1}}$ or $\mathbb{T}_{\mathbb{V}_{c+1}}$,

$$[\overline{x}_n]_t = \text{mean}\left(([x^j_{n+1/2}]_t)_{j\in\mathbb{V}_c\cap\mathbb{T}^*_t}\right) = \frac{[s_{n,c}]_t}{|\mathbb{T}^*_t|}, \tag{6.37}$$

as we have then $|\mathbb{V}_c\cap\mathbb{T}^*_t| = |\mathbb{T}^*_t|$. This means that locally averaging is only required for these blocks. In Figures 6.5 and 6.6, the synchronization workflow is summarized, while, in Algorithm 23, a more detailed account of the whole process is given.

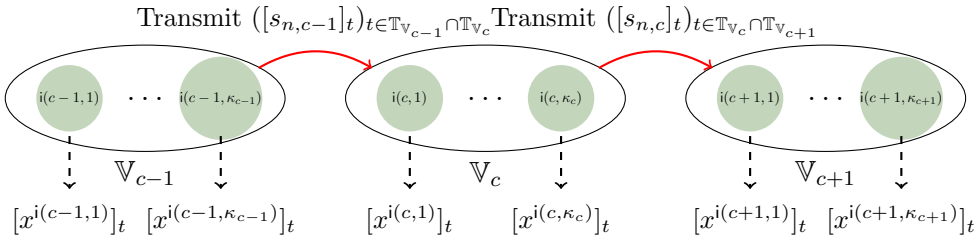


Figure 6.5: *Global synchronisation process: Transmission of local summations to the next computing unit.*

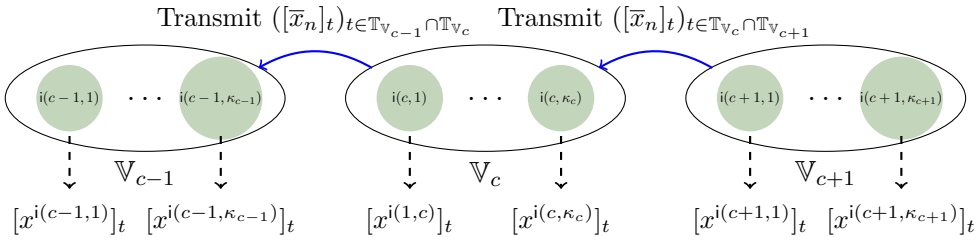


Figure 6.6: *Global synchronisation process: Transmission of averaged blocks to the previous computing unit.*

Remark 6.2

1. It must be emphasized that, in order to facilitate the derivation of our algorithm, a common iteration variable n has been used for each computing unit. However, units have the ability to process data at their own speed. In particular, each unit may perform a different number of local synchronizations before a global one is made. In this sense, our algorithm is asynchronous. To

understand why such behavior is allowed, it suffices to note that if no global synchronization arises and $\mathbb{J}_{n,c} = \varnothing$, $(x_{n+1}^j)_{j\in\mathbb{V}_c} = (x_n^j)_{j\in\mathbb{V}_c}$. This means that such a void iteration can be used to model a time when the c-th computing unit is idle while others are locally updating their variables.

2. When the c-th computing unit will operate a global synchronization, it will suspend its activities until it receives data from units $c-1$ (line 33) and/or $c+1$ (line 37), which happens only when these units also are globally synchronizing their variables. To ensure low latencies, global synchronization steps however have to be scheduled (quasi-)periodically for each computing unit based on their processing speeds (faster ones should schedule less frequent synchronizations than slower ones). Alternatively, when one unit decides to perform a global synchronization, it can broadcast a message to the others to warn them to do the same.

3. Other forms of local consensus could be thought of. For example, another choice would consist in setting $L = 2C - 1$ and $(\forall c \in \{1, \dots, C-1\})$ $\mathbb{V}_{C+c} = \mathbb{V}_c \cup \mathbb{V}_{c+1}$. Then, each node $c \in \{1, \dots, C-1\}$ could be responsible for driving the synchronization with its neighbor of index $c+1$. It appears however more difficult, in this context, to devise an efficient procedure for avoiding deadlocks.

Algorithm 23 Special case of distributed PDFB for the c-th computing unit

Setting of global constants:
$\mathbb{T}_j \equiv$ index set of blocks used at node $j \in \{1,\dots,J\}$
$\mathbb{T}_t^* \equiv$ index set of nodes using block $t \in \{1,\dots,T\}$
$\{\omega_{j,t} \mid 1 \leqslant j \leqslant J, t \in \mathbb{T}_j\} \subset]0,1]$ such that $(\forall t \in \{1,\dots,T\}) \sum_{j \in \mathbb{T}_t^*} \omega_{j,t} = 1$
$\vartheta = \min_{1 \leqslant j \leqslant J, 1 \leqslant t \leqslant T} \omega_{j,t}$, $\epsilon \in]0,1]$, $(\gamma_n)_{n\in\mathbb{N}}$ sequence of $[\epsilon, 2-\epsilon]$
Initialization:
$\mathbb{V}_c \equiv$ index set of nodes associated with computing unit c
$\mathbb{T}_{\mathbb{V}_c} \equiv$ set of block indices used in $\mathbb{V}_c$ (with the convention $\mathbb{T}_{\mathbb{V}_0} = \mathbb{T}_{\mathbb{V}_{C+1}} = \varnothing$)
$B_j \in \mathbb{R}^{M_j \times M_j}$ with $B_j \succeq \sum_{t \in \mathbb{T}_j} \omega_{j,t}^{-1} \mathcal{A}_{j,t} \mathcal{A}_{j,t}^\top, \quad j \in \mathbb{V}_c$
$\vartheta_c = \min_{j \in \mathbb{V}_c, t \in \mathbb{T}_j} \omega_{j,t}, \quad \ell \in \{1,\dots,C\}$
$y_0^j \in \mathbb{R}^{M_j}, [x_0^j]_t = [\widetilde{x}]_t - \omega_{j,t}^{-1} \mathcal{A}_{j,t}^\top y_0^j, \quad j \in \mathbb{V}_c, t \in \mathbb{T}_j.$
for $n = 0, 1, \dots$ **do** *Main loop*
  $\mathbb{J}_{n,c} \subset \mathbb{V}_c$
  for $j \in \mathbb{J}_{n,c}$ **do**
    $\widetilde{y}_n^j = y_n^j + \gamma_n B_j^{-1} \sum_{t \in \mathbb{T}_j} \mathcal{A}_{j,t} [x_n^j]_t$
    $y_{n+1}^j = \widetilde{y}_n^j - \gamma_n B_j^{-1} \mathrm{prox}_{\gamma_n B_j^{-1}, g_j}\big(\gamma_n^{-1} B_j \widetilde{y}_n^j\big)$
    for $t \in \mathbb{T}_j$ **do** $[x_{n+1/2}^j]_t = [x_n^j]_t - \omega_{j,t}^{-1} \mathcal{A}_{j,t}^\top (y_{n+1}^j - y_n^j)$;
  end for
  for $j \in \mathbb{V}_c \setminus \mathbb{J}_{n,c}$ **do**
    $y_{n+1}^j = y_n^j$
    $([x_{n+1/2}^j]_t)_{t \in \mathbb{T}_j} = ([x_n^j]_t)_{t \in \mathbb{T}_j}$
  end for
  for $t \in \mathbb{T}_{\mathbb{V}_c}$ **do** $[s_{n,c}]_t = \sum_{j \in \mathbb{V}_c \cap \mathbb{T}_t^*} [x_{n+1/2}^j]_t$;
  if synchronization is local **then**
    for $j \in \mathbb{V}_c$ **do**
      for $t \in \mathbb{T}_j$ **do**
        $[x_{n+1}^j]_t = [x_{n+1/2}^j]_t + \gamma_n \vartheta_c \, \omega_{j,t}^{-1} \Big(\frac{[s_{n,c}]_t}{|\mathbb{V}_c \cap \mathbb{T}_t^*|} - [x_{n+1/2}^j]_t \Big)$
      end for
    end for
  else *Global synchronization*
    if $c \neq C$ **then send** $([s_{n,c}]_t)_{t \in \mathbb{T}_{\mathbb{V}_c} \cap \mathbb{T}_{\mathbb{V}_{c+1}}}$ to unit $c+1$;
    if $c \neq 1$ **then**
      wait for receiving $([s_{n,c-1}]_t)_{t \in \mathbb{T}_{\mathbb{V}_{c-1}} \cap \mathbb{T}_{\mathbb{V}_c}}$ from unit $c-1$
      for $t \in \mathbb{T}_{\mathbb{V}_{c-1}} \cap \mathbb{T}_{\mathbb{V}_c}$ **do** $[\overline{x}_n]_t = \frac{1}{|\mathbb{T}_t^*|} \big([s_{n,c-1}]_t + [s_{n,c}]_t \big)$;
      send $([\overline{x}_n]_t)_{t \in \mathbb{T}_{\mathbb{V}_{c-1}} \cap \mathbb{T}_{\mathbb{V}_c}}$ to unit $c-1$
    end if
    if $c \neq C$ **then wait for receiving** $([\overline{x}_n]_t)_{t \in \mathbb{T}_{\mathbb{V}_c} \cap \mathbb{T}_{\mathbb{V}_{c+1}}}$ from unit $c+1$;
    for $t \in \mathbb{T}_{\mathbb{V}_c} \setminus (\mathbb{T}_{\mathbb{V}_{c-1}} \cup \mathbb{T}_{\mathbb{V}_{c+1}})$ **do** $[\overline{x}_n]_t = \frac{[s_{n,c}]_t}{|\mathbb{T}_t^*|}$;
    for $j \in \mathbb{V}_c$ **do**
      for $t \in \mathbb{T}_j$ **do**
        $[x_{n+1}^j]_t = [x_{n+1/2}^j]_t + \gamma_n \vartheta \omega_{j,t}^{-1} \big([\overline{x}_n]_t - [x_{n+1/2}^j]_t \big)$
      end for
    end for
  end if
end for

6.5 Application to video denoising

6.5.1 Observation model

In this section, we provide a validation of the proposed distributed algorithm for denoising video sequences. The original sequence $\overline{x} = ([\overline{x}]_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TL}$ is naturally decomposed in T blocks of data, each corresponding to one image containing L pixels. The degradation model relating the observed noisy sequence $y = ([y]_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TL}$ to the sought sequence $\overline{x}$ with $TL = N$ is given by

$$(\forall t \in \{1, \ldots, T\}) \quad [y]_t = [\overline{x}]_t + [w]_t, \tag{6.38}$$

where $([w]_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TL}$ represents an additive zero-mean white Gaussian noise. An estimate of the unknown video can be inferred by solving Problem (6.3) where $J = T$ and $\widetilde{x} = y$. The last quadratic term in (6.3) is a least squares data fidelity term ensuring the compliance with Model (6.38), and functions $(g_j)_{1 \leqslant j \leqslant T}$ stand for regularization functions that incorporate both temporal and spatial prior knowledge on each video frame. The temporal regularization is fulfilled by taking into account motion compensation between the previous and next neighbouring frames. More precisely, at each time $t \in \{2, \ldots, T-1\}$, the linear operator A_t extracts the current frame x_t and its neighbors (x_{t-1}, x_{t+1}) as shown by Figure 6.7. The linear operators $(A_t)_{1 \leqslant t \leqslant T}$ thus have the block sparse structure expressed by (6.28) with

$$(\forall t \in \{1, \ldots, T\}) \quad \mathbb{T}_t = \big\{ \max\{t-1, 1\}, t, \min\{t+1, T\} \big\} \tag{6.39}$$

and

$$\mathcal{A}_{1,1} = \begin{bmatrix} I_L \\ 0 \end{bmatrix}, \quad \mathcal{A}_{1,2} = \begin{bmatrix} 0 \\ I_L \end{bmatrix}, \tag{6.40}$$

$$(\forall t \in \{2, \ldots, T-1\}) \qquad \mathcal{A}_{t,t-1} = \begin{bmatrix} I_L \\ 0 \\ 0 \end{bmatrix}, \quad \mathcal{A}_{t,t} = \begin{bmatrix} 0 \\ I_L \\ 0 \end{bmatrix}, \quad \mathcal{A}_{t,t+1} = \begin{bmatrix} 0 \\ 0 \\ I_L \end{bmatrix}, \tag{6.41}$$

$$\mathcal{A}_{T,T-1} = \begin{bmatrix} I_L \\ 0 \end{bmatrix}, \quad \mathcal{A}_{T,T} = \begin{bmatrix} 0 \\ I_L \end{bmatrix}. \tag{6.42}$$

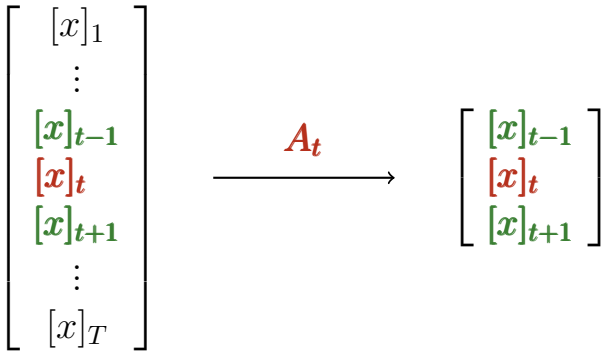


Figure 6.7: *Linear operator A_t that extracts the current frame and its neighbors.*

For every $t \in \{1, \dots T\}$, each regularization function $g_t \colon \mathbb{R}^{M_t} \to [0, +\infty[$ is convex, proper, lower semi-continuous and such that

$$M_t = \begin{cases} 3L & \text{if } t \neq 1 \text{ and } t \neq T \\ 2L & \text{otherwise,} \end{cases} \tag{6.43}$$

and, for every $x = ([x]_t)_{1 \leqslant t \leqslant T}$,

$$g_t\left(([x]_{t'})_{t' \in \mathbb{T}_t}\right) = \eta \ \mathrm{tgv}([x]_t) + \iota_{[x_{\min}, x_{\max}]^L}([x]_t) + h_t\left(([x]_{t'})_{t' \in \mathbb{T}_t}\right), \tag{6.44}$$

where "tgv" denotes the *Total Generalized Variation* regularization from [Bredies et al., 2010], defined as

$$(\forall z \in \mathbb{R}^L) \quad \mathrm{tgv}(z) = \min_{q \in \mathbb{R}^{2L}} \ \alpha_0 \, \chi_2\left(Dz - q\right) + \alpha_1 \, \chi_3\left(\mathsf{G}q\right), \tag{6.45}$$

with $(\alpha_0, \alpha_1) \in]0, +\infty[^2$, $D \in \mathbb{R}^{2L \times L}$ is the concatenation of the horizontal and vertical spatial gradient operators:

$$D = \begin{bmatrix} \nabla_{\mathsf{H}} \\ \nabla_{\mathsf{V}} \end{bmatrix}, \quad \text{with} \quad \nabla_{\mathsf{H}} \in \mathbb{R}^{L \times L}, \quad \nabla_{\mathsf{V}} \in \mathbb{R}^{L \times L}, \tag{6.46}$$

and χ is the sparsity promoting function defined in (5.7). Moreover, $\mathsf{G} \in \mathbb{R}^{3L \times 2L}$ is a second order derivative operator given by

$$\mathsf{G} = \begin{bmatrix} -\nabla_{\mathsf{H}}^\top & 0 \\ -\nabla_{\mathsf{V}}^\top & -\nabla_{\mathsf{H}}^\top \\ 0 & -\nabla_{\mathsf{V}}^\top \end{bmatrix}. \tag{6.47}$$

The indicator function $\iota_{[x_{\min}, x_{\max}]^L}$ in (6.44) imposes a range $[x_{\min}, x_{\max}]$ on the pixel values in each frame. In addition, h_t is a function introducing a temporal regular-

ization of the form

$$h_t\left(([x]_{t'})_{t'\in\mathbb{T}_t}\right) = \begin{cases} \beta_{t-1,t}\,\chi_1([x]_t - M_{t-1\to t}\,[x]_{t-1}) + \beta_{t+1,t}\,\chi_1([x]_t - M_{t+1\to t}\,[x]_{t+1}) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{if } t\neq 1 \text{ and } t\neq T \\ \beta_{2,1}\,\chi_1([x]_1 - M_{2\to 1}\,[x]_2) \qquad\qquad \text{if } t = 1 \\ \beta_{T-1,T}\,\chi_1([x]_T - M_{T-1\to T}\,[x]_{T-1}) \qquad\qquad \text{if } t = T, \end{cases} \tag{6.48}$$

where $M_{t-1\to t}\in\mathbb{R}^{L\times L}$ (resp. $M_{t+1\to t}\in\mathbb{R}^{L\times L}$) is a motion compensation operator between the reference frame x_{t-1} (resp. x_{t+1}) and the current frame x_t, defined as described in Section 5.2.2. Finally, η, $(\beta_{t-1,t})_{2\leqslant t\leqslant T}$ and $(\beta_{t+1,t})_{1\leqslant t\leqslant T-1}$ are positive regularization parameters controlling the importance of the contribution of their associated terms. The values of these parameters have been set empirically so as to achieve the best denoising performance.

6.5.2 Proposed method

We employ our proposed asynchronous distributed framework to address the previous denoising problem. More precisely, we use the practical implementation detailed in Algorithm 23. Functions $(g_t)_{1\leqslant t\leqslant T}$ and their associated primal variables $([x^t]_{t'})_{t'\in\mathbb{T}_t}$ for $t\in\{1,\ldots,T\}$, are spread over C computing units, each of them handling the same number of nodes, i.e., $(\forall c\in\{1,\ldots,C\})\ \kappa_c=\kappa$ (with $T=\kappa C$). The associated hyperedges are then given by

$$(\forall c\in\{1,\ldots,C\}) \qquad \mathbb{V}_c = \{(c-1)\kappa+1,\ldots,c\kappa\}. \tag{6.49}$$

Note that, since

$$(\forall c\in\{1,\ldots,C\}) \qquad \mathbb{T}_{\mathbb{V}_c} = \left\{\max\{(c-1)\kappa,1\},\ldots,\min\{c\kappa+1,T\}\right\} \tag{6.50}$$

$$\Rightarrow\ (\forall c\in\{1,\ldots,C-1\}) \qquad \mathbb{T}_{\mathbb{V}_c}\cap\mathbb{T}_{\mathbb{V}_{c+1}} = \{c\kappa, c\kappa+1\}, \tag{6.51}$$

Assumption 6.1 holds provided that $\kappa>1$.

In the local optimization first performed at the n-th iteration of Algorithm 23, we used, for every $j\in\{1,\ldots,T\}$, $B_j=\sum_{t\in\mathbb{T}_j}\omega_{j,t}^{-1}I_{M_j}$ and $\gamma_n\equiv 1.7$. Then, the local or global synchronization steps are performed as described in Section **??**. In our case, for every $t\in\{1,\ldots,T\}$,

$$\mathbb{T}_t^* = \mathbb{T}_t \tag{6.52}$$

so that, if $t\in\mathbb{T}_{\mathbb{V}_c}$ with $c\in\{1,\ldots,C\}$ corresponds neither to the smallest index nor the largest index in $\mathbb{V}_c$, then 3 values need to be summed to compute $[s_{n,c}]_t$. If t is smallest or largest index in $\mathbb{V}_c$, then the summation involves two terms whereas, if $c>1$ and $t=(c-1)\kappa$ (resp. $c<C$ and $t=c\kappa+1$)), then $[s_{n,c}]_t=[x_{n+1/2}^{t+1}]_t$ (resp. $[s_{n,c}]_t=[x_{n+1/2}^{t-1}]_t$). In global synchronization steps, by virtue of (6.51), only

variables $[s_{n,c}]_{c\kappa}$ and $[s_{n,c}]_{c\kappa+1}$ need to be transmitted from computing unit $c \neq C$ to computing unit $c+1$, which in return sends back the updated averages $[\overline{x}_n]_{c\kappa}$ and $[\overline{x}_n]_{c\kappa+1}$. This workflow is illustrated in Figures 6.8 and 6.9 by an example showing two computing units handling $\kappa = 3$ nodes.

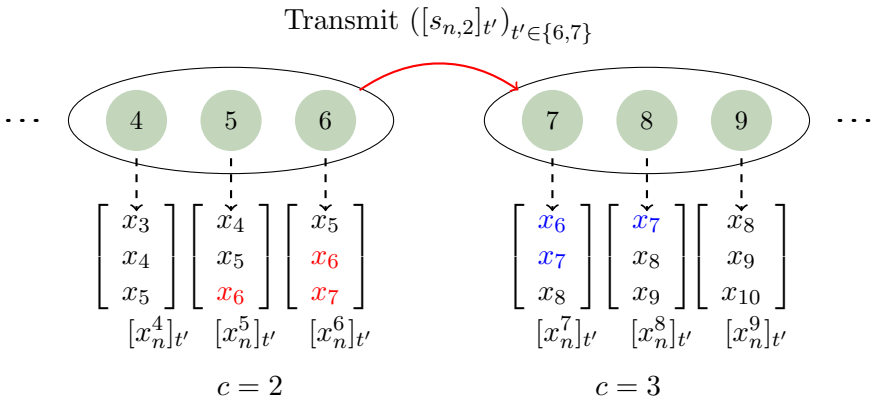


Figure 6.8: *Transmission of local sums* $([s_{n,2}]_{t'})_{t'\in\{6,7\}}$ *shared between* $\mathbb{T}_{\mathbb{V}_2} = \{3,4,5,6,7\}$ *and* $\mathbb{T}_{\mathbb{V}_3} = \{6,7,8,9,10\}$ *from computing unit* $c = 2$ *to computing unit* $c = 3$.

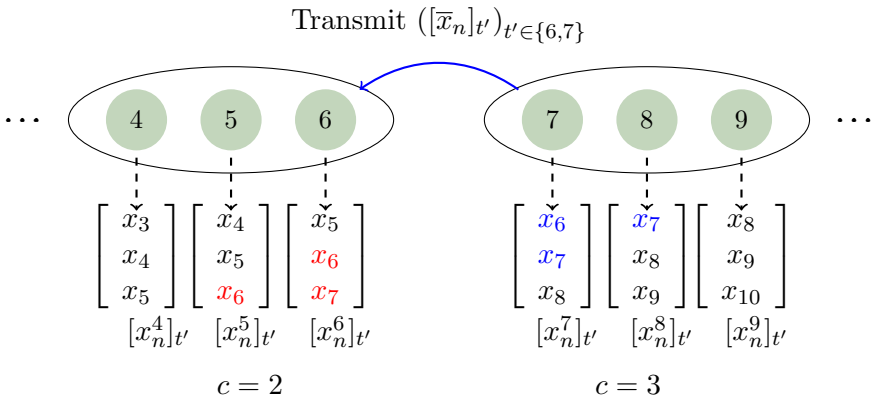


Figure 6.9: *Transmission of averaged images* $([\overline{x}_n]_{t'})_{t'\in\{6,7\}}$ *from computing unit* $c = 3$ *to computing unit* $c = 2$.

It can be noticed that the global synchronizations are activated every 4 iterations. This synchronization frequency has been chosen in order to achieve a good trade-off between the communication overhead and a satisfactory convergence speed of the algorithm. It should be mentioned that the weights $(\omega_{j,t})_{1\leqslant t\leqslant T, j\in\mathbb{T}_t^\star}$ are set to $\frac{1}{|\mathbb{T}_t^\star|}$.

6.5.3 Simulation results

The performance of the proposed denoising method are evaluated on the standard test video sequences **Foreman** and **Claire** with $T = 72$ frames. These frames are of size 348×284 for the first sequence and 300×278 for the second one, hence $N = 7115904$ (resp. $N = 6004800$). The degraded videos are obtained by adding zero-mean white Gaussian noise to the original video sequences, resulting in an initial SNR of 24.41 dB for the first sequence and 24.77 dB for the second one. Our method is implemented with Julia-0.4.6 and a *Message Passing Interface* (MPI) wrapper for managing communication between cores [Forum, 1994; Gropp et al., 1999]. We use a multi-core architecture using 2 Intel(R) Xeon(R) E5-2670 v3 CPU @ 2.3 GHz processors, each of them having 12 cores, hence $C = 24$. The experiments are run using 60 iterations of Algorithm 23, which appears to be sufficient to reach the convergence of our method.

We evaluate the proposed distributed approach in terms of restoration quality and acceleration provided by our algorithm with respect to the number of used computing units. The images composing the video sequences are partitioned in groups of equal size κ processed by the computing units, thereby we consider the cases when $C \in \{1, 2, 3, 4, 6, 8, 9, 12, 18, 24\}$ cores are employed, as shown in Table 6.1.

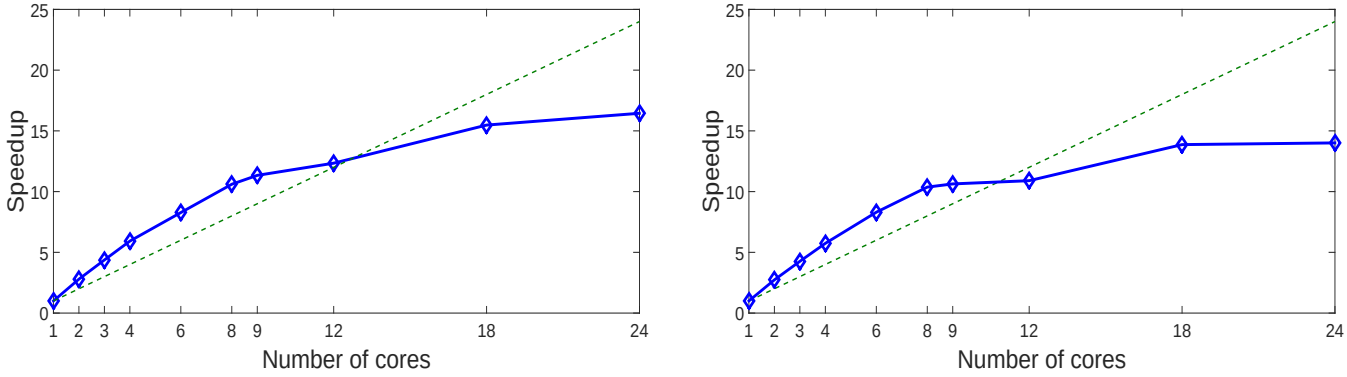
Number of cores C	1	2	3	4	6	8	9	12	18	24
Number of images per core κ	72	36	24	18	12	9	8	6	4	3

Table 6.1: *Investigated simulation scenarios and the number of images per core in each case.*

Figure 6.10 shows the associated speedup in execution time for both sequences with respect to the number of used cores, estimated as follows:

$$\text{Speedup for } C \text{ cores} \quad = \quad \frac{\text{Execution time with 1 core}}{\text{Execution time with } C \text{ cores}}. \tag{6.53}$$

For the first sequence the execution time with 1 core is equal to 107003 s, while it is equal to 84247 s for the second one.

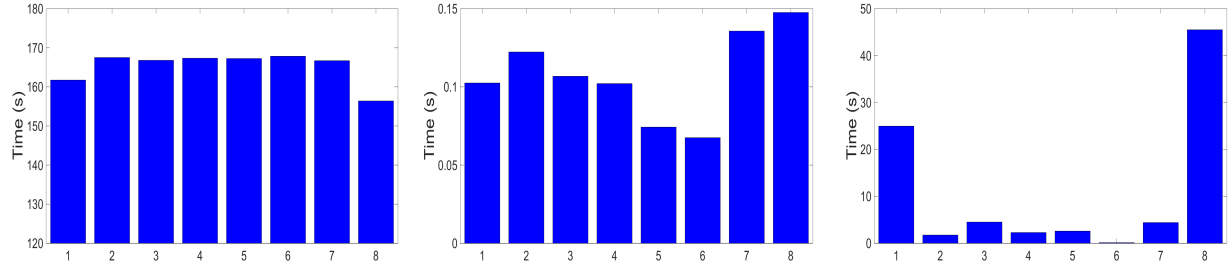


(a) ***Foreman*** *sequence.* **(b)** ***Claire*** *sequence.*

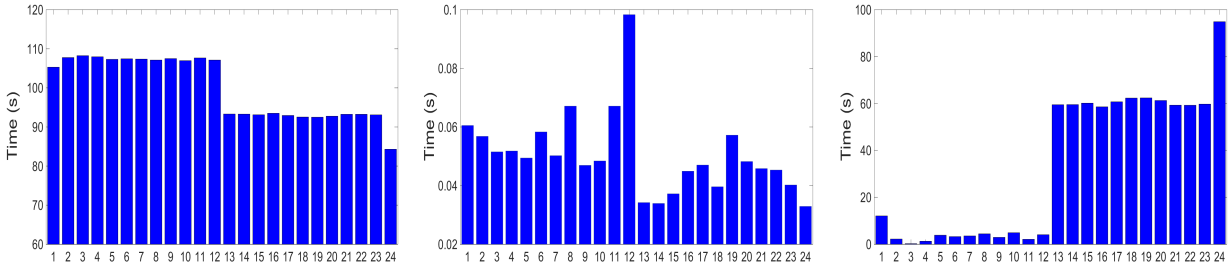
Figure 6.10: *Speedup with respect to the number of used cores: proposed method (solid, blue, diamond), linear speedup (dashed, green).*

Figure 6.10 shows that the speedup increases super linearly as we increase the number of cores from 1 to 9. Indeed, due to the large size of the dataset, it cannot be stored in the cache memory when a small number of cores are used. Hence, a significant amount of time is spent in RAM access [Janakiram et al., 2005]. By increasing the number of cores, the data seem to fit more and more the cache size, which reduces the RAM access time and consequently the global execution time despite the communication overhead. However, this speedup is limited up to some extent, after which its increase becomes slower as the number of core increases, displaying a saturation effect (in agreement with Amdahl's law [Amdahl, 1967]) when the number of computing units is more than 9.

In order to investigate the latter behaviour, we display in Figure 6.11 the execution times per core on **Foreman** sequence, for the three main steps of Algorithm 23 namely, the local optimization, local synchronization, and global synchronization when either $C = 8$ or $C = 24$ cores are used. As expected we observe a significant reduction of the execution time for the local optimization step when going from 8 to 24 cores, but the resulting gain is less than 3, although the computations are then performed independently on each core. The average execution time for the local synchronization step is also reduced as the number of images handled by each core decreases. One can finally observe that the global communication overhead increases as a larger number of cores is used. This behavior appears to be consistent, however it can be noticed on Figure 6.11b that the second half of cores (13 to 24) is much slower than the first one, which is detrimental to the global synchronization process. This fact seems to point out hardware limitations of the Intel-based two-processor computer architecture that we use.



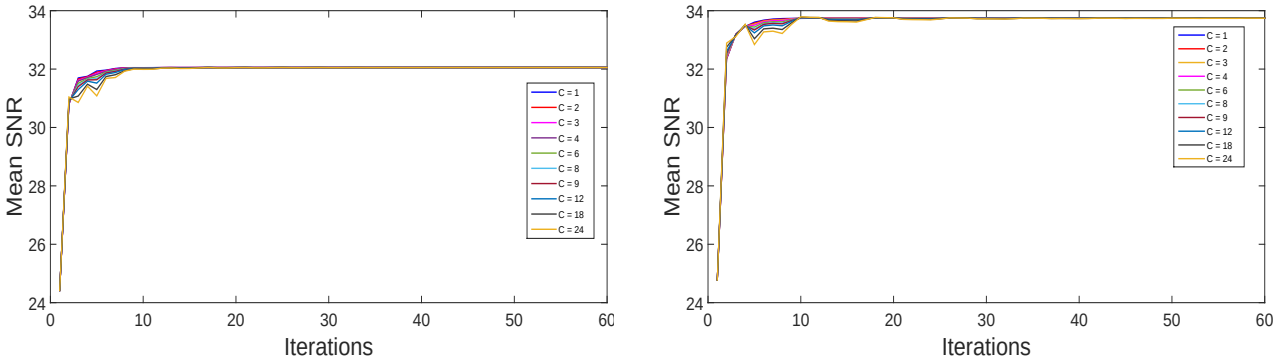
(a) *8 cores.*



(b) *24 cores.*

Figure 6.11: *Execution time of Algorithm 23 steps: local optimization (left), local synchronization (middle), global synchronization (right).*

Figure 6.12 illustrates the performance of our denoising method in terms of restoration quality. One can observe that our method achieves satisfactory restoration results with an improvement of 7.6 dB for **Foreman** and 9 dB for **Claire** with respect to the degraded video. Moreover, the convergence to the sought solution has been reached in each experiment regardless the number of used cores. However, it can be noticed a slight deterioration of the convergence profile when the number of cores is increased, since the global synchronisation plays then a more prominent role.



(a) ***Foreman*** *sequence.* **(b)** ***Claire*** *sequence.*

Figure 6.12: *SNR versus iteration number.*

Finally, Figures 6.13 and 6.14 show some frames extracted from the degraded and restored sequences, which allow us to evaluate the good visual quality of the performed denoising.

Figure 6.13: ***Foreman*** *sequence: Input degraded images (top) initial SNR = 24.41 dB, associated restored images (bottom) final SNR = 32.04 dB.*

6.6 Conclusion

This chapter has introduced a fully parallelized version of the preconditioned dual block-coordinate forward-backward algorithm for computing proximity operators. Our algorithm benefits from all the advantages of primal-dual methods and the acceleration provided by a block-coordinate strategy combined with a variable metric approach. We mainly focused on an instance of the proposed approach for which we proposed an asynchronous implementation, assuming that a given number of computing units is available. Although our distributed algorithm can be applied to a wide range of inverse problems, we have been interested in its application to video sequence denoising. The experimental results we obtained are quite promising and demonstrate the ability of our algorithm to take advantage of multiple cores. An acceleration of about 15 has indeed been reached with a standard two-processor computer configuration.

Figure 6.14: ***Claire*** *sequence: Input degraded images (top) initial SNR = 24.77 dB, associated restored images (bottom) final SNR = 33.74 dB.*

- Chapter 7 -

Alternating proximal method for blind video deconvolution

Contents

7.1 Introduction

Blurring occurs frequently in video sequences captured by consumer devices, as a result of various factors such as lens aberrations, defocus, relative camera-scene motion, and camera shake. When it comes to the contents of archive documents such as old films and television shows, the degradations are even more serious due to several physical phenomena happening during the sensing, transmission, recording, and storing processes [Chenot et al., 1998; Kokaram, 1998; Naranjo and Albiol, 2000; Kokaram and Godsill, 2002]. Most of these degradations can be summarized into two main categories. The first type is of random nature and appears in images as noise, mainly caused by electronic devices. The second one is deterministic and results in blur and oscillations whose common causes are lens imperfections, motion of the scene, diffusion in sensors, and physical or electronic transmission problems.

We provide in this chapter a versatile formulation of the blind video deconvolution problem that seeks to estimate both the sharp unknown video sequence and the underlying blur kernel from an observed video. This inverse problem is severely *ill-posed*, and an appropriate solution can be obtained by modeling it as a nonconvex minimization problem. We propose a novel iterative algorithm to solve it, grounded on the use of recent advances in convex and nonconvex optimization techniques, and having the ability of including numerous well-known regularization strategies [Rudin et al., 1992; Bredies et al., 2010; Repetti et al., 2015; Perrone and Favaro, 2016].

The reminder of this chapter is structured as follows: we introduce in Section 7.2 the formulation of the blind deconvolution problem as a minimization problem and present a number of regularization strategies that can be adopted in the context of image/video processing. Afterwards, we present our minimization approach in Section 7.3, which solves efficiently the resulting nonconvex problem. Section 7.4 illustrates some experimental results on synthetic and real video sequences, and finally, some conclusions are given in Section 7.5.

7.2 Problem statement

7.2.1 Observation model

Blind video deconvolution amounts to inferring an original sharp video sequence $\boldsymbol{x} = (x_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$ and a spatial convolution kernel $h \in \mathbb{R}^P$ from an observed degraded video sequence $\boldsymbol{y} = (y_t)_{1 \leqslant t \leqslant T} \in \mathbb{R}^{TN}$, satisfying the following degradation model

$$(\forall t \in \{1, \dots, T\}) \qquad y_t = h * x_t + w_t, \tag{7.1}$$

where T denotes the number of frames included in the video sequence and $(w_t)_{1\leqslant t\leqslant T} \in \mathbb{R}^{TN}$ represents an additive noise. If no additional information is supplied, this problem is very *ill-conditioned* and its resolution may lead to unstable and unsatisfactory results. Thus, we resort to the following penalized formulation in order to solve it:

$$\underset{\boldsymbol{x}\in\mathbb{R}^{TN},h\in\mathbb{R}^{P}}{\text{minimize}} \quad (F(\boldsymbol{x},h) = \Phi(\boldsymbol{x},h) + \Psi(\boldsymbol{x}) + \Theta(h))\,. \tag{7.2}$$

The cost function F is composed of a least squares data fidelity term Φ which ensures the compliance to Model (7.1), and is given by

$$\Phi(\boldsymbol{x},h) = \frac{1}{2}\sum_{t=1}^{T} \|h * x_t - y_t\|^2, \tag{7.3}$$

and of two regularization functions Ψ and Θ that incorporate prior information on the sought images and kernel, respectively. The objective function F is nonconvex due to the coupling existing in the data fidelity term between the variables $\boldsymbol{x}$ and h. This suggests the use of an optimization method that alternates between the estimation of the images composing the sequence $\boldsymbol{x}$, and the identification of the PSF h in order to reach a critical point of (7.2). Besides, it is worth noticing that the choice of the regularization functions Ψ and Θ plays a prominent role in the quality of the restored video and the identified kernel. A number of regularization strategies has been proposed in the context of image/video processing. Moreover, the adopted optimization method depends heavily on the mathematical properties of the retained penalty functions. One of the contributions of this chapter is to propose a unique and versatile optimization method that can handle a wide class of regularization functions, as detailed in the following.

7.2.2 Video estimation

Let us consider a simpler problem which consists in estimating the video sequence while assuming a known PSF h. The images composing the video sequence can be inferred by solving the following problem

$$\underset{\boldsymbol{x}\in\mathbb{R}^{TN}}{\text{minimize}} \;\; \Phi(\boldsymbol{x},h) + \Psi(\boldsymbol{x}), \tag{7.4}$$

where Ψ is defined as follows

$$(\forall \boldsymbol{x} \in \mathbb{R}^{TN}) \quad \Psi(\boldsymbol{x}) = \sum_{t=1}^{T} \big(\eta\,\psi(x_t) + \iota_{[x_{\min},x_{\max}]^N}(x_t)\big) + \mathcal{M}(\boldsymbol{x}), \tag{7.5}$$

where ψ is a spatial regularization function handling each frame x_t separately, $\iota_{[x_{\min},x_{\max}]^N}$ denotes an indicator function that sets a range on the pixel values of

each image, and finally $\mathcal{M}$ is a smoothed version of the motion term (5.9) introduced in Section 5.2.2:

$$(\forall \boldsymbol{x} \in \mathbb{R}^{TN}) \quad \mathcal{M}(\boldsymbol{x}) = \frac{1}{2} \sum_{t=1}^{T} \sum_{\ell \in \mathcal{V}_t} \beta_{\ell,t} \|x_t - M_{\ell \to t} x_\ell\|^2. \tag{7.6}$$

Various choices of spatial regularization functions can be adopted in the Model (7.5). Let us list some of them herebelow.

- *Total Variation (TV)* is one of the most popular regularization method in image restoration. It has been initially introduced for image denoising and reconstruction problems [Rudin et al., 1992], and reads:

$$(\forall z \in \mathbb{R}^N) \quad \psi(z) = \chi_2 \left(Dz \right), \tag{7.7}$$

  where $D \in \mathbb{R}^{2N \times N}$ is the discrete gradient operator defined in (6.46) as the concatenation of the horizontal and vertical gradient operators $\nabla_{\mathsf{H}} \in \mathbb{R}^{N \times N}, \nabla_{\mathsf{V}} \in \mathbb{R}^{N \times N}$ respectively, and χ_q is the sparsity promoting function in (5.7). The total variation promotes the sparsity of the image derivatives, which has the advantage of reducing the noise and preserving strong edges. However, it may lead to piecewise constant images and induce staicase artefacts [Nikolova, 2009].

- *Total Variation on a Staggered Grid (TVSG)* that has been recently proposed in [Condat, 2016], introduces a new formulation of the total variation with a more accurate adaptation of the continuous definition to the discrete domain, instead of the classical finite differences in (6.46). It resorts to a sophisticated gradient operator which is defined as

$$(\forall z \in \mathbb{R}^N) \quad \psi(z) = \min_{(v_1,v_2,v_3) \in \mathbb{R}^{(2N)^3}} \Big\{ \chi_2 \left(v_1 \right) + \chi_2 \left(v_2 \right) + \chi_2 \left(v_3 \right) \;\Big|\; L_1^\top v_1 + L_2^\top v_2 + L_3^\top v_3 = Dz \Big\}, \tag{7.8}$$

  where $L_1^\top, L_2^\top, L_3^\top$ denote the adjoint operators of L_1, L_2, L_3 respectively, defined as follows. Let $u \in \mathbb{R}^{2N} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$, then :

$$L_1 u \in \mathbb{R}^{2N} = \begin{bmatrix} q_{1,1} \\ q_{1,2} \end{bmatrix} \text{ is such that}$$
$$q_{1,1}(i,j) = u_1(i,j),$$
$$q_{1,2}(i,j) = \left(u_2(i,j) + u_2(i,j-1) + u_2(i+1,j) + u_2(i+1,j-1)\right)/4,$$

$$L_2 u \in \mathbb{R}^{2N} = \begin{bmatrix} q_{2,1} \\ q_{2,2} \end{bmatrix} \text{ is such that}$$
$$q_{2,1}(i,j) = \left(u_1(i,j) + u_1(i-1,j) + u_1(i,j+1) + u_1(i-1,j+1)\right)/4,$$
$$q_{2,2}(i,j) = u_2(i,j),$$

$$L_3 u \in \mathbb{R}^{2N} = \begin{bmatrix} q_{3,1} \\ q_{3,2} \end{bmatrix} \text{ is such that}$$
$$q_{3,1}(i,j) = \left(u_1(i,j) + u_1(i-1,j)\right)/2,$$
$$q_{3,2}(i,j) = \left(u_2(i,j) + u_2(i,j-1)\right)/2,$$

where $i \in \{1,\dots,N_1\}$, $j \in \{1,\dots,N_2\}$ with $N_1 N_2 = N$.
This new definition of gradient fields leads to a regularized approach that improves the sharpness of the edges, and presents a better isotropy compared to the standard total variation.

- *Smoothed One Over Two-Total Variation (SOOT-TV)* is a nonconvex sparsity promoting function combining the ℓ_1/ℓ_2 norm and the total variation operator. ℓ_1/ℓ_2 can be viewed as a more accurate approximation to ℓ_0 compared to the convex ℓ_1 norm, as shown in Figure 7.1.

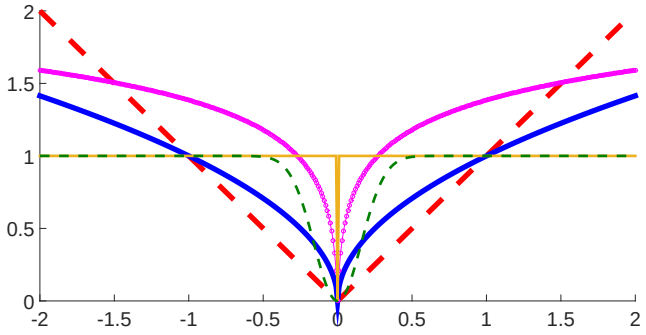


Figure 7.1: *Sparsity promoting norms: ℓ_0 norm (thin solid yellow), ℓ_1 norm (thick dashed red), ℓ_1/ℓ_2 norm (thick solid blue), log-ℓ_1 norm (thin magenta 'o'), Welsch penalty (thin dashed green).*

Here, we will focus on the log-smoothed version of the ℓ_1/ℓ_2 norm called "SOOT" introduced in [Repetti et al., 2015]. The prior then reads:

$$(\forall z \in \mathbb{R}^N) \quad \psi(z) = \log\left(\frac{\ell_{1,\alpha}(Dz) + \beta}{\ell_{2,\lambda}(Dz)}\right), \tag{7.9}$$

where

$$\ell_{1,\alpha}(Dz) = \sum_{n=1}^{2N}\left(\sqrt{(Dz)_n^2 + \alpha^2} - \alpha\right), \quad \ell_{2,\lambda}(Dz) = \sqrt{\sum_{n=1}^{2N}(Dz)_n^2 + \lambda^2}, \tag{7.10}$$

$D \in \mathbb{R}^{2N\times N}$ is the discrete gradient operator defined in (6.46) and α, β, λ are positive parameters.

- *Smoothed log-Total Variation (log-TV)* is a nonconvex smooth sparsity promoting regularization function from [Perrone and Favaro, 2016] defined as follows

$$\begin{aligned}(\forall z \in \mathbb{R}^N) \quad \psi(z) &= \sum_{n=1}^{N}\log\left(\sqrt{(\nabla_{\mathsf{H}} z)_n^2 + (\nabla_{\mathsf{V}} z)_n^2 + \alpha^2}\right), \\ &= \frac{1}{2}\sum_{n=1}^{N}\log\left((\nabla_{\mathsf{H}} z)_n^2 + (\nabla_{\mathsf{V}} z)_n^2 + \alpha^2\right),\end{aligned} \tag{7.11}$$

where $\alpha > 0$ and $\nabla_{\mathsf{H}} \in \mathbb{R}^{N\times N}, \nabla_{\mathsf{V}} \in \mathbb{R}^{N\times N}$ are gradient operators in the horizontal and vertical directions, defined in (2.16) and (2.17) respectively. Similarly to the ℓ_1/ℓ_2 norm, the log-based penalty used in (7.11) can be viewed as a nonconvex approximation to ℓ_0.

- *Welsch-Total Variation (Welsch-TV)* is based on the so-called "Welsch function" [Dennis and Welsch, 1978] defined by

$$\mathbb{R} \to \mathbb{R} \ : \ t \to 1 - \exp\left(-t^2/(2\sigma^2)\right).$$

The Welsch function is bounded and approaches 1 exponentially fast as $|t| \to +\infty$. It is convex near the origin, for $t^2 < \sigma^2$ and nonconvex elsewhere. Its adaptation to the context of image and video deconvolution is realized by applying it to the image gradients in order to improve their sparsity:

$$(\forall z \in \mathbb{R}^N) \quad \psi(z) = \sum_{n=1}^{N}\left(1 - \exp\left(-\left((\nabla_{\mathsf{H}} z)_n^2 + (\nabla_{\mathsf{V}} z)_n^2\right)/(2\sigma^2)\right)\right). \tag{7.12}$$

Finally, we also consider the spatial regularization already defined in Chapters 5 and 6, namely:

- *Semi-Local Total Variation (SLTV)* defined in (5.8)

$$(\forall z \in \mathbb{R}^N) \quad \psi(z) = \sum_{\ell \in \Omega} \chi_2 \left(L_\ell z \right) \text{ with } L_\ell = (\mathrm{Id}_{2N} - V_\ell) D.$$

  where the operators $(V_\ell)_{\ell \in \{1,\dots,6\}} \in \mathbb{R}^{2N \times 2N}$ are the shift operators illustrated in Figure 5.1.

- *Total Generalized Variation (TGV)* defined in (6.45)

$$(\forall z \in \mathbb{R}^N) \quad \psi(z) = \min_{q \in \mathbb{R}^{2N}} \alpha_0 \chi_2 \left(Dz - q \right) + \alpha_1 \chi_3 \left(\mathsf{G} q \right).$$

  where G is a second order derivative operator defined in (6.47).

7.2.3 Kernel identification

The spatial convolution kernel is estimated by solving the minimization problem (7.2) with respect to h while keeping the images $(x_t)_{1 \leqslant t \leqslant T}$ fixed, which reduces to solve

$$\underset{h \in \mathbb{R}^P}{\text{minimize}} \;\; \Phi(\boldsymbol{x}, h) + \Theta(h), \tag{7.13}$$

where Θ accounts for an indicator function of a set $\mathcal{H}$ representing a constrained set, so that *a priori* information on the sought kernel are satisfied. In the proposed method, the following constraints are considered:

$$(\forall h \in \mathbb{R}^P) \quad \Theta(h) = \iota_{\mathcal{H}}(h), \tag{7.14}$$

with

$$\mathcal{H} = \Big\{ h = (h_p)_{1 \leqslant p \leqslant P} \mid \sum_{p=1}^{P} h_p = 1, \tag{7.15}$$

$$(\forall p \in \{1, \dots, P\}) \; h_{\min,p} \leqslant h_p \leqslant h_{\max,p}, \; \Big\}. \tag{7.16}$$

The first constraint (7.15) is used in order to avoid the so-called *scaling ambiguity*. In fact, let $(\widehat{x}, \widehat{h})$ be a solution to (7.2), then each pair $(\alpha \widehat{x}, \frac{1}{\alpha} \widehat{h})$ with $\alpha \neq 0$ is also a solution satisfying Model (7.1). Thus imposing (7.15) ensures the uniqueness and the normalization of the solution. The second constraint (7.16) is adjusted regarding to prior information on the physical properties of the sought convolution kernel. As an example, for old television archive contents, the kernel h may have a narrow spike and small (possibly negative) components.

7.3 Optimization method

7.3.1 Minimization strategy

In order to find estimates of the sharp video sequence $\boldsymbol{x}$ and the convolution kernel h, we alternatively minimize F in (7.2) with respect to each image $(x_t)_{1\leqslant t\leqslant T}$, followed by a minimization with respect to the kernel h.
To do so, we propose an alternating strategy based on the block-coordinate variable metric forward-backward algorithm given by Algorithm 8 in Chapter 3, where at each iteration, a forward-backward iteration is performed with respect to each image x_t, by means of a gradient descent step on the smooth part of the restriction of F to x_t, and a proximal step on the remaining nonsmooth part. Then, we apply a proximal step on the restriction of F to h while all the images of the sequence $\boldsymbol{x}$ are considered as constant.

In order to adapt our alternating minimization strategy to the resolution of Problem (7.2), we propose to rewrite the objective function F as follows

$$\underset{\boldsymbol{x}\in\mathbb{R}^{TN},h\in\mathbb{R}^{P}}{\text{minimize}} \quad \left(F(\boldsymbol{x},h) = f_1(\boldsymbol{x},h) + f_2(\boldsymbol{x}) + \Theta(h)\right), \tag{7.17}$$

where f_1 represents the smooth part of $\Phi(\boldsymbol{x},h) + \Psi(\boldsymbol{x})$ and f_2 its nonsmooth part. Two cases arise depending on the smoothness of the spatial regularization function ψ:

- ψ is nonsmooth, e.g., in case of TV, SLTV, TGV and TVSG, then

$$\begin{aligned}(\forall \boldsymbol{x}\in\mathbb{R}^{TN}),(\forall h\in\mathbb{R}^{P}) \quad & f_1(\boldsymbol{x},h) = \Phi(\boldsymbol{x},h) + \mathcal{M}(\boldsymbol{x}),\\ (\forall \boldsymbol{x}\in\mathbb{R}^{TN}) \quad & f_2(\boldsymbol{x}) = \sum_{t=1}^{T}\left(\eta\,\psi(x_t) + \iota_{[x_{\min},x_{\max}]^N}(x_t)\right).\end{aligned}$$

- ψ is smooth, e.g., it corresponds to the nonconvex regularizations, SOOT-TV, log-TV and Welsch-TV, then

$$\begin{aligned}(\forall \boldsymbol{x}\in\mathbb{R}^{TN}),(\forall h\in\mathbb{R}^{P}) \quad & f_1(\boldsymbol{x},h) = \Phi(\boldsymbol{x},h) + \left(\sum_{t=1}^{T}\eta\,\psi(x_t)\right) + \mathcal{M}(\boldsymbol{x}),\\ (\forall \boldsymbol{x}\in\mathbb{R}^{TN}) \quad & f_2(\boldsymbol{x}) = \sum_{t=1}^{T}\iota_{[x_{\min},x_{\max}]^N}(x_t).\end{aligned}$$

We propose to solve Problem (7.17) with the proximal-based alternating minimization strategy shown in Algorithm 8, which reads in our context:

Algorithm 24 Blind video deconvolution

Initialization:

For every $k \in \mathbb{N}, \gamma_t^k \in]0,2[$ and $\mu^k \in]0,+\infty[$

for $k = 0, 1, \ldots$ **do**

  for $t = 1, \ldots, T$ **do**

    $\check{\boldsymbol{x}}^{t,k} = \left(x_1^{k+1}, \ldots, x_{t-1}^{k+1},\ x_t^k\ , x_{t+1}^k \ldots, x_T^k\right)$

    $\widetilde{x}_t^k = x_t^k - \gamma_t^k A_{t,k}^{-1} \left(\nabla_{x_t} f_1(\check{\boldsymbol{x}}^{t,k}, h^k)\right)$

    $x_t^{k+1} = \mathrm{prox}_{(\gamma_t^k)^{-1} A_{t,k}, f_2}\left(\widetilde{x}_t^k\right)$

  end for

  $h^{k+1} = \mathrm{prox}_{\mu^k\,(\Theta + \Phi(\boldsymbol{x}^{k+1}, \cdot))}(h^k)$

end for

where ∇_{x_t} denotes the gradient of f_1 with respect to the image x_t, and $A_{t,k}$ is a semi-definite positive matrix in $\mathbb{R}^{N\times N}$ satisfying the majorant condition for f_1 at $\check{\boldsymbol{x}}^{t,k}$, i.e., the function $Q(\cdot, \check{\boldsymbol{x}}^{t,k})$ defined by

$$\left(\forall x \in \mathbb{R}^N\right) \quad Q(x, \check{\boldsymbol{x}}^{t,k}) = f_1(\check{\boldsymbol{x}}^{t,k}) + \langle x - x_t^k \mid \nabla_{x_t} f_1(\check{\boldsymbol{x}}^{t,k})\rangle + \frac{1}{2}\|x - x_t^k\|_{A_{t,k}}^2, \tag{7.18}$$

is a majorant function of the restriction of f_1 to the image x_t at $\check{\boldsymbol{x}}^{t,k}$.

7.3.2 Construction of the majorant

The choice of a good majorant function $Q(\cdot, \check{\boldsymbol{x}}^{t,k})$ of the restriction of f_1 to the image x_t at each iteration $k \in \mathbb{N}$ has a strong leverage on the numerical performance of the proposed method. Thereby, one has to favour curvature matrices $(A_{t,k})_{1\leqslant t\leqslant T, k\in\mathbb{N}}$ that are easy to handle.
Depending on the choice of the spatial regularization ψ, $(A_{t,k})_{1\leqslant t\leqslant T,k\in\mathbb{N}}$ is defined as described below.

1. ψ *is nonsmooth* e.g., in case of TV, SLTV, TGV, and TVSG

$$(\forall t \in \{1, \ldots, T\})(\forall k \in \mathbb{N}) \quad A_{t,k} = \upsilon_{t,k}\,\mathrm{Id}_N, \tag{7.19}$$

   where $\upsilon_{t,k}$ is the Lipschitz constant of the gradient of f_1 with respect to x_t^k:

$$\nabla\left(\frac{1}{2}\|h^k * \cdot - y_t\|^2 + \mathcal{M}(x_1^{k+1}, \ldots, x_{t-1}^{k+1}, \cdot, x_{t+1}^k, \ldots, x_T^k)\right). \tag{7.20}$$

   According to (7.6), such a Lipschitz constant is thus expressed as

$$\upsilon_{t,k} = \|H_k\|^2 + \sum_{\ell\in\mathcal{V}_t} \beta_{\ell,t} + \sum_{\ell\in\{1,\ldots,T\}:t\in\mathcal{V}_\ell} \beta_{t,\ell}\|M_{t\to\ell}\|^2, \tag{7.21}$$

where $\|H_k\|$ is maximum magnitude of the frequency response of the blur filter estimate at iteration k and $(\|M_{t\to\ell}\|)_{1\leqslant\ell\leqslant T, t\in\mathcal{V}_\ell}$ denote the spectral norms of the operators used for motion compensation based on x_t^k.

2. *ψ is SOOT-TV*

$$(\forall t \in \{1,\dots,T\})(\forall k\in\mathbb{N})\;\; A_{t,k} = \left(\upsilon_{t,k} + \frac{9\eta\,\|D\|^2}{8\lambda^2}\right)\mathrm{Id}_N + \frac{\eta\, A_{\ell_{1,\alpha}}(\breve{\boldsymbol{x}}^{t,k})}{\ell_{1,\alpha}(D\breve{\boldsymbol{x}}^{t,k})+\beta}, \tag{7.22}$$

where υ is Lipschitz constant of (7.20) and

$$(\forall z\in\mathbb{R}^N)\quad A_{\ell_{1,\alpha}}(z) = \mathrm{Diag}\left(\Omega^\top s(z)\right) + \varepsilon\,\mathrm{Id}_N, \tag{7.23}$$

where, for every $z\in\mathbb{R}^N$, $s(z)\in\mathbb{R}^{2N}$ is such that, for every $i\in\{1,\dots,2N\}$

$$(\forall z\in\mathbb{R}^N)\quad s^{(i)}(z) = ((D^{(i)}z)^2+\alpha^2)^{-1/2},$$

$D^{(i)}\in\mathbb{R}^{1\times N}$ denotes the ith row of D, $\Omega^{(i,j)} = \left|D^{(i,j)}\right|\sum_{k=1}^N\left|D^{(i,k)}\right|$ and $\varepsilon>0$.

3. *ψ is log-TV*

$$(\forall t \in \{1,\dots,T\})(\forall k\in\mathbb{N})\quad A_{t,k} = \upsilon_{t,k}\mathrm{Id}_N + \eta\, A_{\log}(\breve{\boldsymbol{x}}^{t,k}), \tag{7.24}$$

where υ is the Lipschitz constant of (7.20) and

$$(\forall z\in\mathbb{R}^N)\quad A_{\log}(z) = \mathrm{Diag}\left(\Omega^\top s(z)\right) + \varepsilon\,\mathrm{Id}_N, \tag{7.25}$$

where, for every $z\in\mathbb{R}^N$, $s(z)\in\mathbb{R}^N$ is such that, for every $i\in\{1,\dots,N\}$

$$(\forall z\in\mathbb{R}^N)\quad s^{(i)}(z) = ((\nabla_{\mathsf{H}}^{(i)}z)^2 + (\nabla_{\mathsf{V}}^{(i)}z)^2+\alpha^2)^{-1},$$

$\nabla_{\mathsf{H}}^{(i)}\in\mathbb{R}^{1\times N}$ (resp. $\nabla_{\mathsf{V}}^{(i)}\in\mathbb{R}^{1\times N}$) denotes the ith row of ∇_{H} (resp. ∇_{V}) , and $\Omega = \Omega_{\mathsf{H}} + \Omega_{\mathsf{V}}$:

$$\Omega_{\mathsf{H}}^{(i,j)} = \left|\nabla_{\mathsf{H}}^{(i,j)}\right|\sum_{k=1}^N\left|\nabla_{\mathsf{H}}^{(i,k)}\right|\;,\;\Omega_{\mathsf{V}}^{(i,j)} = \left|\nabla_{\mathsf{V}}^{(i,j)}\right|\sum_{k=1}^N\left|\nabla_{\mathsf{V}}^{(i,k)}\right|,\;\varepsilon>0.$$

4. *ψ is Welsch-TV*

$$(\forall t \in \{1,\dots,T\})(\forall k\in\mathbb{N})\quad A_{t,k} = \upsilon_{t,k}\mathrm{Id}_N + \eta\, A_w(\breve{\boldsymbol{x}}^{t,k}), \tag{7.26}$$

where υ is Lipschitz constant of (7.20) and

$$(\forall z \in \mathbb{R}^N) \quad A_w(z) = \sigma^{-2}\mathrm{Diag}\left(\Omega^\top s(z)\right) + \varepsilon\,\mathrm{Id}_N, \tag{7.27}$$

where, for every $z \in \mathbb{R}^N$, $s(z) \in \mathbb{R}^N$ is such that, for every $i \in \{1, \ldots, N\}$

$$(\forall z \in \mathbb{R}^N) \quad s^{(i)}(z) = \exp\left(-((\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2)/(2\sigma^2)\right),$$

$\nabla_{\mathsf{H}}^{(i)} \in \mathbb{R}^{1\times N}$ (resp. $\nabla_{\mathsf{V}}^{(i)} \in \mathbb{R}^{1\times N}$) denotes the ith row of ∇_{H} (resp. ∇_{V}) and $\Omega = \Omega_{\mathsf{H}} + \Omega_{\mathsf{V}}$:

$$\Omega_{\mathsf{H}}^{(i,j)} = \left|\nabla_{\mathsf{H}}^{(i,j)}\right| \sum_{k=1}^{N} \left|\nabla_{\mathsf{H}}^{(i,k)}\right| \ , \ \Omega_{\mathsf{V}}^{(i,j)} = \left|\nabla_{\mathsf{V}}^{(i,j)}\right| \sum_{k=1}^{N} \left|\nabla_{\mathsf{V}}^{(i,k)}\right|, \ \varepsilon > 0.$$

Proof.

SOOT-TV

Let us set

$$\left(\forall z \in \mathbb{R}^N\right) \quad \eta\,\psi(z) = \eta\,\log\left(\frac{\ell_{1,\alpha}(Dz) + \beta}{\ell_{2,\lambda}(Dz)}\right) = \psi_1(z) + \psi_2(z), \tag{7.28}$$

where $\psi_1(z) = \eta\,\log(\ell_{1,\alpha}(Dz)+\beta)$ and $\psi_2 = -\eta\,\log(\ell_{2,\lambda}(Dz))$. We need to prove that

1. $A_{\ell_{1,\alpha}}(z)$ satisfies the majoration condition for $\ell_{1,\alpha}$ at z,
2. $\frac{A_{\ell_{1,\alpha}}(z)}{\ell_{1,\alpha}(Dz)+\beta}$ satisfies the majoration condition for ψ_1 at z,
3. $\frac{9\eta\|D\|^2}{8\lambda^2}$ is a Lipschitz constant for ψ_2.

Proving Statements 2 and 3 is similar to the proof provided in [Repetti et al., 2015]. Let us now consider Statement 1, first let us define

$$\ell_{1,\alpha}(D\,\cdot) = \sum_{i=1}^{2N} \phi\left(D^{(i)}\,\cdot\right), \tag{7.29}$$

$$(\forall v \in \mathbb{R}) \qquad \phi(v) = \sqrt{v^2 + \alpha^2} - \alpha. \tag{7.30}$$

We have [Allain et al., 2006]

$$(\forall u \in \mathbb{R}) \quad \phi\left(u\right) \leqslant \phi\left(v\right) + (u - v)\dot{\phi}\left(v\right) + \frac{\kappa(v)}{2}\left(u - v\right)^2, \tag{7.31}$$

with, for every $v \in \mathbb{R}$,

$$\dot{\phi}(v) = \frac{v}{\sqrt{v^2+\alpha^2}} \quad \text{and} \quad \kappa(v) = \frac{1}{\sqrt{v^2+\alpha^2}}. \tag{7.32}$$

Thus, for every $\omega \in \mathbb{R}^N$,

$$\phi\left(D^{(i)}\omega\right) \leqslant \phi\left(D^{(i)}z\right) + \left\langle \omega - z, {D^{(i)}}^\top \dot{\phi}\left(D^{(i)}z\right)\right\rangle + \frac{\kappa(D^{(i)}z)}{2}\left(D^{(i)}(\omega - z)\right)^2, \tag{7.33}$$

By combining (7.29) and (7.33), we have, for every $\omega \in \mathbb{R}^N$,

$$\begin{aligned}\ell_{1,\alpha}(D\omega) \leqslant \ell_{1,\alpha}(Dz) + \langle \omega - z, \nabla\left(\ell_{1,\alpha} \circ (D\cdot)\right)(z)\rangle \\ + \frac{1}{2}\left(D(\omega - z)\right)^\top \mathrm{Diag}(s(z))\, D(\omega - z).\end{aligned} \tag{7.34}$$

where, for every $z \in \mathbb{R}^N$, $s(z) \in \mathbb{R}^{2N}$ is such that, for every $i \in \{1, \dots 2N\}$,

$$(\forall z \in \mathbb{R}^N) \quad s^{(i)}(z) = ((D^{(i)}z)^2 + \alpha^2)^{-1/2}.$$

Let us define $(\sigma_j)_{1\leqslant j\leqslant N} \in]0, +\infty[^N$ such that $\sum_{j=1}^N \sigma_j = 1$, so that, for every $i \in \{1, \dots, 2N\}$

$$\begin{aligned}\left(D^{(i)}(\omega - z)\right)^2 &= \left(\sum_{j=1}^N D^{(i,j)}(\omega^j - z^j)\right)^2, \\ &= \left(\sum_{j=1}^N D^{(i,j)}(\omega^j - z^j)\right)^2, \\ &= \left(\sum_{j=1}^N \sigma_j \frac{D^{(i,j)}(\omega^j - z^j)}{\sigma_j}\right)^2.\end{aligned} \tag{7.35}$$

According to the Jensen's inequality, we get

$$\begin{aligned}\left(\sum_{j=1}^N \sigma_j \frac{D^{(i,j)}(\omega^j - z^j)}{\sigma_j}\right)^2 &\leqslant \sum_{j=1}^N \sigma_j \left(\frac{D^{(i,j)}(\omega^j - z^j)}{\sigma_j}\right)^2, \\ &= \sum_{j=1}^N \frac{\left(D^{(i,j)}(\omega^j - z^j)\right)^2}{\sigma_j}.\end{aligned} \tag{7.36}$$

Taking for all $j \in \{1, \dots, N\}$, $\sigma_j = \frac{\left|D^{(i,j)}\right|}{\sum_{k=1}^N \left|D^{(i,k)}\right|}$, then

$$\begin{aligned}\left(D^{(i)}(\omega - z)\right)^2 &\leqslant \sum_{j=1}^{N} \left(\frac{\left(D^{(i,j)}(\omega^j - z^j)\right)^2}{\sigma_j} \right), \\ &= \sum_{j=1}^{N} \left(\frac{\left(D^{(i,j)}(\omega^j - z^j)\right)^2}{\frac{\left|D^{(i,j)}\right|}{\sum_{k=1}^{N}\left|D^{(i,k)}\right|}} \right), \\ &= \sum_{j=1}^{N} \left|D^{(i,j)}\right| \sum_{k=1}^{N} \left|D^{(i,k)}\right| (\omega^j - z^j)^2. \end{aligned} \tag{7.37}$$

This yields

$$\|D(\omega - z)\|^2 \leqslant \|\omega - z\|^2_{A_{\ell_{1,\alpha}}(z)}, \tag{7.38}$$

with

$$A_{\ell_{1,\alpha}}(z) = \mathrm{Diag}\left(\Omega^\top s(z)\right) + \varepsilon\, \mathrm{Id}_N, \tag{7.39}$$

where $\Omega^{(i,j)} = \left|D^{(i,j)}\right| \sum_{k=1}^{N} \left|D^{(i,k)}\right|$, and $\varepsilon > 0$.

log-TV

The process of constructing the majorant when ψ stands for log-TV regularization, is similarly to the one of SOOT-TV, by setting [Chouzenoux et al., 2016]

$$\begin{aligned}(\forall z \in \mathbb{R}^N) \quad \psi(z) &= \frac{1}{2} \sum_{i=1}^{N} \log\left((\nabla_{\mathsf{H}} z)_i^2 + (\nabla_{\mathsf{V}} z)_i^2 + \alpha^2\right), \\ &= \sum_{i=1}^{N} \phi\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2}\right), \end{aligned} \tag{7.40}$$

where

$$(\forall v \in \mathbb{R}) \qquad \phi(v) = \frac{1}{2} \log\left(v^2 + \alpha^2\right). \tag{7.41}$$

Using (7.31) with, for every $v \in \mathbb{R}$,

$$\dot{\phi}(v) = \frac{v}{v^2 + \alpha^2} \quad \text{and} \quad \kappa(v) = \frac{1}{v^2 + \alpha^2}, \tag{7.42}$$

we have that, for every $\omega \in \mathbb{R}^N$,

$$\begin{aligned}\phi\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)}\omega)^2 + (\nabla_{\mathsf{V}}^{(i)}\omega)^2}\right) &\leqslant \phi\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2}\right) \\ &\quad + \left\langle \omega - z, \dot{\phi}\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2}\right) \right\rangle \\ &\quad + \frac{\kappa\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2}\right)}{2} \left((\nabla_{\mathsf{H}}^{(i)}(\omega - z))^2 + (\nabla_{\mathsf{V}}^{(i)}(\omega - z))^2\right). \end{aligned} \tag{7.43}$$

By combining (7.40) and (7.43), we obtain, for every $\omega \in \mathbb{R}^N$,

$$\psi(\omega) \leqslant \psi(z) + \langle \omega - z, \nabla\psi(z)\rangle + \frac{1}{2}\,(\omega - z)^\top \operatorname{Diag}(s(z))\ (\omega - z)\,, \tag{7.44}$$

where, for every $z \in \mathbb{R}$, $s(z) \in \mathbb{R}^N$ is such that, for every $i \in \{1, \dots N\}$,

$$(\forall z \in \mathbb{R}^N) \quad s^{(i)}(z) = \left((\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2 + \alpha^2\right)^{-1}.$$

Thereby, for every $\omega \in \mathbb{R}^N$,

$$\psi(\omega) \leqslant \psi(z) + \langle \omega - z, \nabla\psi(z)\rangle + \frac{1}{2}\|\omega - z\|^2_{A_{\log}(z)}, \tag{7.45}$$

where

$$(\forall z \in \mathbb{R}^N) \quad A_{\log}(z) = \operatorname{Diag}\left(\Omega^\top s(z)\right) + \varepsilon \operatorname{Id}_N, \tag{7.46}$$

with $\varepsilon > 0$ and $\Omega = \Omega_{\mathsf{H}} + \Omega_{\mathsf{V}}$ such that

$$\Omega_{\mathsf{H}}^{(i,j)} = \left|\nabla_{\mathsf{H}}^{(i,j)}\right| \sum_{k=1}^{N} \left|\nabla_{\mathsf{H}}^{(i,k)}\right|\,, \quad \Omega_{\mathsf{V}}^{(i,j)} = \left|\nabla_{\mathsf{V}}^{(i,j)}\right| \sum_{k=1}^{N} \left|\nabla_{\mathsf{V}}^{(i,k)}\right|.$$

Welsch-TV

The construction of this majorant is analogous to the one of log-TV regularization, by taking

$$\begin{aligned}(\forall z \in \mathbb{R}^N) \quad \psi(z) &= \sum_{i=1}^{N} \left(1 - \exp\left(-\left((\nabla_{\mathsf{H}} z)_i^2 + (\nabla_{\mathsf{V}} z)_i^2\right)/(2\sigma^2)\right)\right), \\ &= \sum_{i=1}^{N} \phi\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2}\right), \end{aligned} \tag{7.47}$$

with

$$(\forall v \in \mathbb{R}) \qquad \phi(v) = 1 - \exp\left(-v^2/(2\sigma^2)\right). \tag{7.48}$$

Using (7.31) and, for every $v \in \mathbb{R}$,

$$\dot\phi(v) = \frac{v}{\sigma^2}\exp\left(-v^2/(2\sigma^2)\right) \quad \text{and} \quad \kappa(u) = \frac{1}{\sigma^2}\exp\left(-v^2/(2\sigma^2)\right), \tag{7.49}$$

we obtain, for every $\omega \in \mathbb{R}^N$,

$$\begin{aligned}\phi\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)}\omega)^2 + (\nabla_{\mathsf{V}}^{(i)}\omega)^2}\right) &\leqslant \phi\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2}\right) \\ &\quad + \left\langle \omega - z, \dot\phi\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2}\right)\right\rangle \\ &\quad + \frac{\kappa\left(\sqrt{(\nabla_{\mathsf{H}}^{(i)} z)^2 + (\nabla_{\mathsf{V}}^{(i)} z)^2}\right)}{2}\left((\nabla_{\mathsf{H}}^{(i)}(\omega - z))^2 + (\nabla_{\mathsf{V}}^{(i)}(\omega - z))^2\right). \end{aligned} \tag{7.50}$$

By combining (7.47) and (7.50), and following the same idea in the previous proof, we have, for every $\omega \in \mathbb{R}^N$,

$$\psi(\omega) \leqslant \psi(z) + \langle \omega - z, \nabla\psi(z)\rangle + \frac{1}{2}\|\omega - z\|^2_{A_w(z)}, \tag{7.51}$$

where

$$(\forall z \in \mathbb{R}^N) \quad A_w(z) = \operatorname{Diag}\left(\Omega^\top s(z)\right) + \varepsilon \operatorname{Id}_N, \tag{7.52}$$

with, for every $z \in \mathbb{R}$, $s(z) \in \mathbb{R}^N$ is such that, for every $i \in \{1, \dots N\}$,

$$(\forall z \in \mathbb{R}^N) \qquad s^{(i)}(z) = \exp\left(-((\nabla^{(i)}_{\mathsf{H}} z)^2 + (\nabla^{(i)}_{\mathsf{V}} z)^2)/(2\sigma^2)\right),$$

$\varepsilon > 0$ and $\Omega = \Omega_{\mathsf{H}} + \Omega_{\mathsf{V}}$ such that

$$\Omega^{(i,j)}_{\mathsf{H}} = \left|\nabla^{(i,j)}_{\mathsf{H}}\right| \sum_{k=1}^{N} \left|\nabla^{(i,k)}_{\mathsf{H}}\right| , \quad \Omega^{(i,j)}_{\mathsf{V}} = \left|\nabla^{(i,j)}_{\mathsf{V}}\right| \sum_{k=1}^{N} \left|\nabla^{(i,k)}_{\mathsf{V}}\right| .$$

□

7.3.3 Implementation of the proximity operator of f_2

The retained metric matrices $(A_{t,k})_{1\leqslant t\leqslant T, k\in\mathbb{N}}$ being diagonal, the proximity operator involved in Algorithm 24 may have a closed form expression when f_2 is a "simple" function. However, when the latter is more sophisticated, for example when it represents a sum of functions possibly composed with linear operators, we have to resort to some iterative strategies in order to evaluate it. In our framework, the computation of the proximity operator of f_2 at each image x_t (i.e., $\operatorname{prox}_{(\gamma^k_t)^{-1}A_{t,k},f_2}$) depends also on the choice of the spatial regularization function ψ. In some instances, it has an explicit form while in others, we must use specific algorithms to evaluate it, namely

- for smooth nonconvex regularization functions ψ (SOOT-TV, log-TV and Welsch-TV), we have

  $$(\forall z \in \mathbb{R}^N) \quad f_2(z) = \iota_{[x_{\min}, x_{\max}]^N}(z),$$

  so that the proximity operator has an explicit expression, since it reduces to the projection into $[x_{\min}, x_{\max}]$.

- for nonsmooth convex regularization functions ψ (TV, SLTV, TGV, TVSG), we have

  $$(\forall z \in \mathbb{R}^N) \quad f_2(z) = \psi(z) + \iota_{[x_{\min}, x_{\max}]^N}(z),$$

and the proximity operator is evaluated using the following algorithms:

TV	SLTV	TGV	TVSG
Dual forward-backward [Combettes et al., 2011]	Algorithm 15	Primal-dual splitting [Condat, 2013]	Alternating proximal gradient [Ma, 2016]

Table 7.1: *List of optimization algorithms used for computing the proximity operator with respect to the different convex regularization functions.*

7.3.4 Implementation of the proximity operator for kernel estimation

The blur kernel h is estimated in Algorithm 24 by computing the proximity operator of the sum of the data fidelity term and regularization function Θ (i.e., $\mathrm{prox}_{\mu^k\,(\Theta+\Phi(\boldsymbol{x}^{k+1},\cdot))}$). Since there is no closed form expression for the latter proximity operator, we resort to the parallel proximal algorithm (PPXA) in [Combettes and Pesquet, 2008] to evaluate it.

7.3.5 Convergence analysis

The convergence properties of Algorithm 24 depends on the settings of parameters $(\gamma_t^k,\, \mu^k)_{t\in\{1,\dots,T\},k\in\mathbb{N}}$ and the preconditioning matrices $(A_{t,k})_{1\leqslant t\leqslant T,k\in\mathbb{N}}$. First, let us state the following proposition related to the quadratic form of the data fidelity term Φ.

Proposition 7.1 *Let us define the symmetric definite positive matrix*

$$B = \mu X^\top X + \mathrm{Id}_P, \tag{7.53}$$

where Id_P *is the identity matrix of* $\mathbb{R}^P$, $\mu \in (0,+\infty)$ *and* $X \in \mathbb{R}^{TN\times P}$ *is such that* $Xh = (h * x_t)_{1\leqslant t\leqslant T}$. *Then, for every* $h \in \mathbb{R}^P$ *and* $\boldsymbol{x} \in \mathbb{R}^{TN}$,

$$\mathrm{prox}_{\mu\,(\Theta+\Phi(\boldsymbol{x}^{k+1},\cdot))}\,(h) = \mathrm{prox}_{\mu^{-1}B,\Theta}(h - \mu B^{-1}\nabla_h\Phi(\boldsymbol{x},h)), \tag{7.54}$$

and, for every $\boldsymbol{x} \in \mathbb{R}^{TN}$ *and* $h' \in \mathbb{R}^P$,

$$\Phi(\boldsymbol{x},h') + \nabla_{h'}\Phi(\boldsymbol{x}',h')^\top(h-h') + \frac{1}{2}\|h-h'\|^2_{\mu^{-1}B} \geqslant \Phi(\boldsymbol{x},h). \tag{7.55}$$

Proof. Let $q \in \mathbb{R}^P$ be the value of the proximity operator of $\Phi(\boldsymbol{x}, \cdot) + \Theta$ at h, i.e., $q = \operatorname{prox}_{\mu(\Theta+\Phi(\boldsymbol{x},\cdot))}(h)$. We have the following subdifferential inclusion:

$$\begin{aligned} & h - q \in \mu\,(\partial\,\Theta(q) + \nabla_q \Phi(\boldsymbol{x}, q)) \\ \Leftrightarrow\ & h - q \in \mu\,\partial\,\Theta(q) + \mu\,X^\top(Xq - \boldsymbol{y}) \\ \Leftrightarrow\ & h - (\mathrm{Id}_P + \mu\,X^\top X)q + \mu\,X^\top \boldsymbol{y} \in \mu\,\partial\,\Theta(q) \\ \Leftrightarrow\ & (\mathrm{Id}_P + \mu\,X^\top X)^{-1}(h + \mu\,X^\top \boldsymbol{y}) - q \in \mu\,(\mathrm{Id}_P + \mu\,X^\top X)^{-1}\partial\,\Theta(q) \end{aligned} \tag{7.56}$$

Thus, by setting $B = \mathrm{Id}_P + \mu X^\top X$,

$$\begin{aligned} & q = \operatorname{prox}_{\mu^{-1}\,B,\Theta}\,(B^{-1}(h + \mu\,X^\top \boldsymbol{y})) \\ \Leftrightarrow\ & q = \operatorname{prox}_{\mu^{-1}\,B,\Theta}\,(h - \mu\,B^{-1}X^\top(Xh - \boldsymbol{y})) \\ \Leftrightarrow\ & q = \operatorname{prox}_{\mu^{-1}\,B,\Theta}\,(h - \mu\,B^{-1}\nabla_h \Phi(x, h)). \end{aligned} \tag{7.57}$$

Because of the quadratic form of Φ,

$$\Phi(\boldsymbol{x}, h') + \nabla_{h'}\Phi(\boldsymbol{x}', h')^\top(h - h') + \frac{1}{2}(h - h')^\top \nabla^2_{h'}\Phi(\boldsymbol{x}', h')(h - h') = \Phi(\boldsymbol{x}, h), \tag{7.58}$$

where the Hessian of $\Phi(\boldsymbol{x}', \cdot)$ is

$$\nabla^2_{h'}\Phi(\boldsymbol{x}', h') = X^\top X \preceq X^\top X + \mu^{-1}\mathrm{Id}_P = \mu^{-1}B \tag{7.59}$$

($\preceq$ stands for the Loewner order). This yields (7.55). □

This allows us to derive the following convergence result:

Theorem 7.2 *Let us consider Algorithm 24. Assume that*

$$(\forall t \in \{1, \ldots, T\}) \qquad 0 < \inf_{k\in\mathbb{N}} \gamma_t^k \quad \text{and} \quad \sup_{k\in\mathbb{N}} \gamma_t^k < 2, \tag{7.60}$$

$$0 < \inf_{k\in\mathbb{N}} \mu^k \quad \text{and} \quad \sup_{k\in\mathbb{N}} \mu^k < +\infty. \tag{7.61}$$

Then, the sequence $(\boldsymbol{x}^k, h^k)_{k\in\mathbb{N}}$ converges to a critical point $(\widehat{\boldsymbol{x}}, \widehat{h})$ of F. Moreover, $\big(F(\boldsymbol{x}^k, h^k)\big)_{k\in\mathbb{N}}$ is a nonincreasing sequence converging to $F(\widehat{\boldsymbol{x}}, \widehat{h})$.

Proof. It follows from Proposition 7.1 that the proximal step for kernel estimation in Algorithm 24 at iteration $k \in \mathbb{N}$ reduces to a preconditioned forward-backward iteration with the preconditioning matrix $(\mu^k)^{-1}B_k$ where $B_k = \mu^k X_k^\top X_k + \mathrm{Id}_P$ and $X_k \in \mathbb{R}^{TN\times P}$ is such that, for every $h \in \mathbb{R}^P$, $X_k h = (h * x_t^k)_{1\leqslant t\leqslant T}$. Algorithm 24 thus appears as a special case of the block-coordinate variable metric forward-backward algorithm studied in [Chouzenoux et al., 2016] where $T + 1$ blocks of variables are involved (corresponding to the T frames and the kernel to be estimated). Indeed, the cost function in (7.17) satisfies the assumptions required in [Chouzenoux et al., 2016]:

- it is a coercive function (since both variables $\boldsymbol{x}$ and h are constrained to belong to compact sets) and it satisfies Kurdyka-Łojasiewicz inequality;
- f_1 is a function with a Lipschitz continuous gradient;
- the function $(\boldsymbol{x}, h) \mapsto f_2(\boldsymbol{x}) + \Theta(h)$ is a proper convex lower-semicontinuous function which is separable with respect to the blocks of variables;
- according to Section (7.3.2) and Equation (7.55), the curvature matrices $(A_{t,k})_{1\leqslant t\leqslant T}$ and $(\mu^k)^{-1}B_k$ used at each iteration $k \in \mathbb{N}$ provide quadratic majorant approximations to the restriction of f_1 to the current activated block.

In addition, since $(x_t^k)_{1\leqslant t\leqslant T, k\in\mathbb{N}}$ and $(h^k)_{k\in\mathbb{N}}$ are constrained to belong to bounded sets, it follows from the expressions derived in Section 7.3.2 and the positive lower bound already exhibited on $(\upsilon_{t,k})_{1\leqslant t\leqslant T,k\in\mathbb{N}}$ that, for every $t \in \{1,\dots,T\}$, there exists $(a_{t,\min}, a_{t,\max}) \in]0,+\infty[^2$ such that

$$(\forall k \in \mathbb{N}) \qquad a_{t,\min}\mathrm{Id}_N \preceq A_{t,k} \preceq a_{t,\max}\mathrm{Id}_N. \tag{7.62}$$

According to (7.61), there also exists $(b_{\min}, b_{\max}) \in]0,+\infty[^2$ such that

$$(\forall k \in \mathbb{N}) \qquad b_{\min}\mathrm{Id}_P \preceq (\mu^k)^{-1}B_k = X_k^\top X_k + (\mu^k)^{-1}\mathrm{Id}_P \preceq b_{\max}\mathrm{Id}_P. \tag{7.63}$$

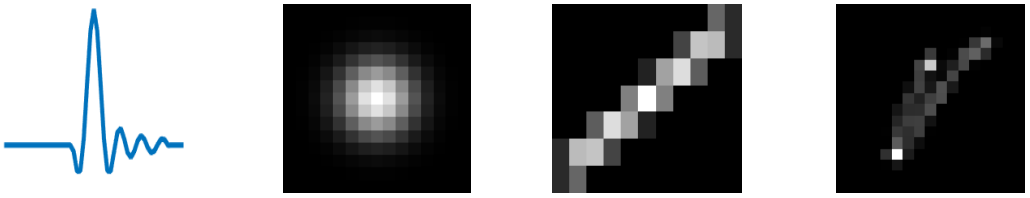
The convergence result then follows from [Chouzenoux et al., 2016, Theorem 3.1]. □

7.4 Experimental results

We assess in this section the performance of the proposed approach on artificially and naturally degraded video sequences. We first begin with an evaluation on the synthetic video sequences **Foreman** and **Claire** presented in Chapter 5, that have been blurred using the four convolution kernels displayed in Figure 7.2, and to which a Gaussian noise with zero mean, and variance equal to 2 have been added. Since it is usually challenging to develop a blind deconvolution method that achieves satisfactory results for both blur kernel and unknown video in a single step, we proceed successively in our method with a blind, and then a non-blind stages.

Our evaluation is composed of two parts, the first part is dedicated to the *blind deconvolution step* in-which we aim at identifying the blur kernel from the input degraded sequence using Algorithm 24. Then, we continue with the so-called *non-blind deconvolution step* where the observed degraded sequence and the identified kernel are both employed to estimate the unknown sharp video sequence.
We also apply the proposed method to the real video sequences **Tachan** and **Au**

théâtre ce soir supplied by INA, that have been introduced in Chapter 5. Neighboring frames such that $|\ell - t| = 1$ have been taken into account in the temporal regularization term $\mathcal{M}$. The motion matrices $(M_{\ell \to t})_{\ell,t}$ have been estimated from the degraded sequence $\boldsymbol{y}$, using the optical flow estimation algorithm from [Liu et al., 2008]. It should be noted that all the experiments are initialized with the Dirac delta function for the kernel identification step, and the input degraded video sequence for both blind and non-blind steps.



(a) $P = 1 \times 53$ **(b)** $P = 15 \times 15$ **(c)** $P = 7 \times 11$ **(d)** $P = 17 \times 17$

Figure 7.2: *Synthetic convolution kernels.*

7.4.1 Blind video deconvolution step

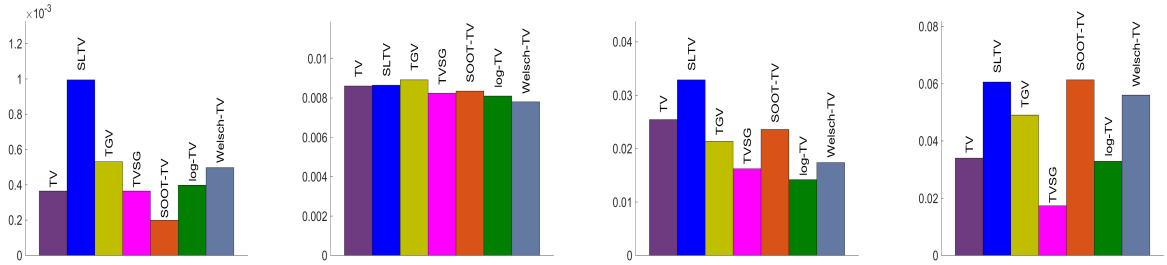
Figure 7.3 shows the quadratic of error on kernel identification for the seven regularization approaches presented in Section 7.2.2, and the four tested convolution kernels. This error is evaluated as follows

$$\text{Error} = \|h - \widehat{h}\|^2, \tag{7.64}$$

where $\widehat{h} \in \mathbb{R}^P$ denotes the ground truth kernel and $h \in \mathbb{R}^P$ is the estimated one. Note that the parameters involved in the regularization approaches have been adjusted in each experiment in order to obtain the lowest possible error.

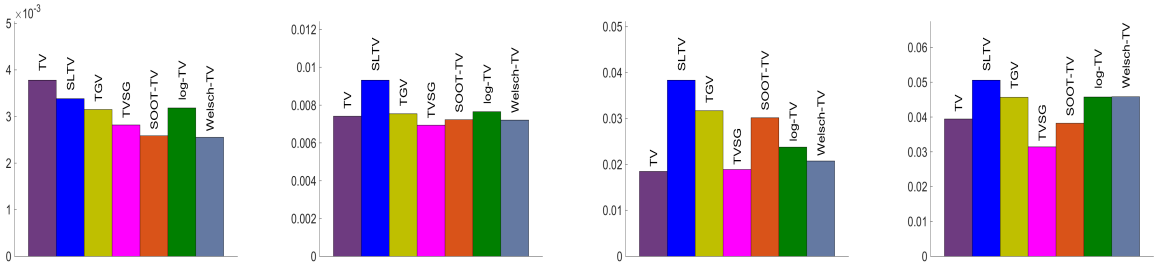
We observe in Figure 7.3 that the results vary slightly depending on the kernels and video sequences. We can notice that the TVSG achieves low errors regardless of the kernel and the video sequence. The nonconvex regularizations achieve also low errors on kernel identification in certain cases (e.g., kernels (a) and (c) with **Foreman** sequence, and kernel (a) with **Claire** sequence). Nevertheless, they can also fail in identifying the correct kernel possibly because of the existence of numerous local minima, in addition, they are harder to adjust since they involve multiple parameters, that may be quite different from a degraded sequence to another (see Table 7.3). Note that the worst results in terms of kernel identification are usually achieved by the SLTV regularization. This may be due to the fact that the latter relies on second order derivative operators, that tend to over-smooth the images and particularly the edges, resulting in an unsatisfactory kernel identification.

It should be noted that the convex regularizations SLTV and TVSG are approximately 5 times slower than TV while TGV is about 10 times slower. Besides, the nonconvex regularizations are comparable with TV in terms of computational cost by iteration. Moreover, SOOT-TV leads to a slower convergence compared to the other nonconvex penalties.



Kernel (a) *Kernel (b)* *Kernel (c)* *Kernel (d)*

(a) ***Foreman*** *sequence*



Kernel (a) *Kernel (b)* *Kernel (c)* *Kernel (d)*

(b) ***Claire*** *sequence*

Figure 7.3: *Performance in terms of error on kernel identification with respect to the different regularizations and blur kernels, from left to right: TV, SLTV, TGV, TVSG, SOOT-TV, log-TV, Welsch-TV.*

Table 7.2 illustrates the gap between the best and worst identification quality scores with respect to the two synthetic sequences and the four convolution kernels.

Sequences		Kernel (a)	Kernel (b)	Kernel (c)	Kernel (d)
Foreman	Worst	9.94×10^{-4} *(SLTV)*	8.92×10^{-3} *(TGV)*	3.28×10^{-2} *(SLTV)*	6.13×10^{-2} *(SOOT-TV)*
	Best	1.98×10^{-4} *(SOOT-TV)*	7.8×10^{-3} *(Welsch-TV)*	1.41×10^{-2} *(log-TV)*	1.73×10^{-2} *(TVSG)*
Claire	Worst	3.78×10^{-3} *(TV)*	9.32×10^{-3} *(SLTV)*	3.83×10^{-2} *(SLTV)*	5.05×10^{-2} *(SLTV)*
	Best	2.55×10^{-3} *(Welsch-TV)*	6.93×10^{-3} *(TVSG)*	1.84×10^{-2} *(TV)*	3.14×10^{-2} *(TVSG)*

Table 7.2: *Gap between the best and worst kernel identification scores.*

7.4.2 Non-blind video deconvolution step

Let us now investigate the performance of the previous regularizations in the context of non-blind video deconvolution. For each kernel, the identified one with the lowest error is selected for performing non-blind deconvolution on the input degraded sequences. To this aim, we resort to the same optimization strategy given by Algorithm 24, where we omit the kernel identification step since it is assumed to be known, as shown by Algorithm 25. For these experiments, the parameters are again adjusted so that the best SNR is obtained. Figures 7.4 and 7.5 illustrate the

Algorithm 25 Non-blind video deconvolution

Initialization:

Let $h \in \mathbb{R}^P$ be the identified kernel

For every $k \in \mathbb{N}, \gamma_t^k \in]0, 2[$

for $k = 0, 1, \ldots$ **do**

  for $t = 1, \ldots, T$ **do**

    $\check{\boldsymbol{x}}^{t,k} = \left(x_1^{k+1}, \ldots, x_{t-1}^{k+1},\ x_t^k\ , x_{t+1}^k \ldots, x_T^k\right)$

    $\widetilde{x}_t^k = x_t^k - \gamma_t^k A_k^{-1} \left(\nabla_{x_t} f_1(\check{\boldsymbol{x}}^{t,k}, h)\right)$

    $x_t^{k+1} = \mathrm{prox}_{(\gamma_t^k)^{-1} A_k, f_2}\left(\widetilde{x}_t^k\right)$

  end for

end for

restoration quality in terms of SNR and MOVIE [Seshadrinathan and Bovik, 2010],

	Foreman				Claire			
	Kernel (a)	Kernel (b)	Kernel (c)	Kernel (d)	Kernel (a)	Kernel (b)	Kernel (c)	Kernel (d)
TV	$\eta = 6.24 \times 10^{-4}$	$\eta = 9.2 \times 10^{-3}$	$\eta = 1.52 \times 10^{-2}$	$\eta = 2.82 \times 10^{-2}$	$\eta = 7.53 \times 10^{-3}$	$\eta = 1.60 \times 10^{-2}$	$\eta = 1.52 \times 10^{-2}$	$\eta = 2.90 \times 10^{-1}$
	$\beta_{\ell\to t} = 6.76 \times 10^{-4}$	$\beta_{\ell\to t} = 1.53 \times 10^{-3}$	$\beta_{\ell\to t} = 1.33 \times 10^{-3}$	$\beta_{\ell\to t} = 8.05 \times 10^{-4}$	$\beta_{\ell\to t} = 3.13 \times 10^{-3}$	$\beta_{\ell\to t} = 7.02 \times 10^{-3}$	$\beta_{\ell\to t} = 1.10 \times 10^{-3}$	$\beta_{\ell\to t} = 9.32 \times 10^{-2}$
SLTV	$\eta = 6.17 \times 10^{-5}$	$\eta = 3.99 \times 10^{-4}$	$\eta = 2.83 \times 10^{-3}$	$\eta = 2.08 \times 10^{-3}$	$\eta = 3.51 \times 10^{-4}$	$\eta = 2.34 \times 10^{-3}$	$\eta = 1.44 \times 10^{-3}$	$\eta = 2.34 \times 10^{-3}$
	$\beta_{\ell\to t} = 2.17 \times 10^{-3}$	$\beta_{\ell\to t} = 2.01 \times 10^{-2}$	$\beta_{\ell\to t} = 6.97 \times 10^{-3}$	$\beta_{\ell\to t} = 7.48 \times 10^{-3}$	$\beta_{\ell\to t} = 1.57 \times 10^{-3}$	$\beta_{\ell\to t} = 7.59 \times 10^{-3}$	$\beta_{\ell\to t} = 3.89 \times 10^{-2}$	$\beta_{\ell\to t} = 1.04 \times 10^{-2}$
TGV	$\eta = 1.99 \times 10^{-3}, \alpha_1 = 0.63$	$\eta = 5.50 \times 10^{-3}, \alpha_1 = 0.77$	$\eta = 3.32 \times 10^{-1}, \alpha_1 = 0.68$	$\eta = 1.34, \alpha_1 = 0.51$	$\eta = 9.36 \times 10^{-3}, \alpha_1 = 0.79$	$\eta = 5.46 \times 10^{-2}, \alpha_1 = 0.92$	$\eta = 8.13 \times 10^{-2}, \alpha_1 = 0.32$	$\eta = 2.44 \times 10^{-2}, \alpha_1 = 0.86$
$\alpha_0 = 1 - \alpha_1$	$\beta_{\ell\to t} = 4.95 \times 10^{-3}$	$\beta_{\ell\to t} = 2.08 \times 10^{-2}$	$\beta_{\ell\to t} = 2.12 \times 10^{-3}$	$\beta_{\ell\to t} = 8.65 \times 10^{-4}$	$\beta_{\ell\to t} = 1.10 \times 10^{-2}$	$\beta_{\ell\to t} = 1.32 \times 10^{-2}$	$\beta_{\ell\to t} = 1.70 \times 10^{-3}$	$\beta_{\ell\to t} = 2.16 \times 10^{-2}$
TVSG	$\eta = 2.52 \times 10^{-3}$	$\eta = 3.30 \times 10^{-2}$	$\eta = 4.36 \times 10^{-2}$	$\eta = 2.55 \times 10^{-2}$	$\eta = 3.16 \times 10^{-3}$	$\eta = 2.93 \times 10^{-2}$	$\eta = 2.73 \times 10^{-2}$	$\eta = 1.84 \times 10^{-2}$
	$\beta_{\ell\to t} = 1.18 \times 10^{-4}$	$\beta_{\ell\to t} = 6.53 \times 10^{-4}$	$\beta_{\ell\to t} = 3.03 \times 10^{-4}$	$\beta_{\ell\to t} = 8.26 \times 10^{-4}$	$\beta_{\ell\to t} = 1.78 \times 10^{-9}$	$\beta_{\ell\to t} = 5.58 \times 10^{-4}$	$\beta_{\ell\to t} = 4.44 \times 10^{-4}$	$\beta_{\ell\to t} = 6.20 \times 10^{-4}$
SOOT-TV	$\eta = 1.48, \lambda = 541.43$	$\eta = 13.65, \lambda = 13$	$\eta = 23.48, \lambda = 7.94$	$\eta = 30.63, \lambda = 11.79$	$\eta = 4.61, \lambda = 11.40$	$\eta = 9.80, \lambda = 14.97$	$\eta = 29.78, \lambda = 297.78$	$\eta = 6.48, \lambda = 9.50$
$\beta = 1.26 \times 10^{-5}$	$\beta_{\ell\to t} = 3.55 \times 10^{-4}$	$\beta_{\ell\to t} = 5.58 \times 10^{-3}$	$\beta_{\ell\to t} = 8.94 \times 10^{-6}$	$\beta_{\ell\to t} = 1.15 \times 10^{-5}$	$\beta_{\ell\to t} = 2.31 \times 10^{-7}$	$\beta_{\ell\to t} = 1.66 \times 10^{-4}$	$\beta_{\ell\to t} = 2.09 \times 10^{-4}$	$\beta_{\ell\to t} = 5.18 \times 10^{-3}$
$\alpha = 1.2 \times 10^{-3}$								
log-TV	$\eta = 2.59 \times 10^{-5}$	$\eta = 4.94 \times 10^{-5}$	$\eta = 7.38 \times 10^{-4}$	$\eta = 2.41 \times 10^{-4}$	$\eta = 5.61 \times 10^{-5}$	$\eta = 7.24 \times 10^{-5}$	$\eta = 4.17 \times 10^{-4}$	$\eta = 5.63 \times 10^{-5}$
	$\alpha = 1.32 \times 10^{-2}$	$\alpha = 1.32 \times 10^{-2}$	$\alpha = 2.79 \times 10^{-3}$	$\alpha = 3.35 \times 10^{-3}$	$\alpha = 1.03 \times 10^{-2}$	$\alpha = 8.44 \times 10^{-3}$	$\alpha = 3.04 \times 10^{-3}$	$\alpha = 1.03 \times 10^{-2}$
	$\beta_{\ell\to t} = 1.50 \times 10^{-3}$	$\beta_{\ell\to t} = 1.12 \times 10^{-3}$	$\beta_{\ell\to t} = 1.59 \times 10^{-5}$	$\beta_{\ell\to t} = 3.26 \times 10^{-4}$	$\beta_{\ell\to t} = 1.35 \times 10^{-3}$	$\beta_{\ell\to t} = 1.08 \times 10^{-3}$	$\beta_{\ell\to t} = 2.33 \times 10^{-4}$	$\beta_{\ell\to t} = 1.47 \times 10^{-3}$
Welsch-TV	$\eta = 7.21 \times 10^{-5}$	$\eta = 1.69 \times 10^{-4}$	$\eta = 5.05 \times 10^{-4}$	$\eta = 5.52 \times 10^{-4}$	$\eta = 2.21 \times 10^{-4}$	$\eta = 2.04 \times 10^{-4}$	$\eta = 4.21 \times 10^{-4}$	$\eta = 2.04 \times 10^{-4}$
	$\sigma = 5.66 \times 10^{-2}$	$\sigma = 1.61 \times 10^{-2}$	$\sigma = 3.66 \times 10^{-2}$	$\sigma = 3.80 \times 10^{-2}$	$\sigma = 1.92 \times 10^{-2}$	$\sigma = 2.16 \times 10^{-2}$	$\sigma = 1.19 \times 10^{-2}$	$\sigma = 2.46 \times 10^{-2}$
	$\beta_{\ell\to t} = 3.59 \times 10^{-4}$	$\beta_{\ell\to t} = 3.11 \times 10^{-4}$	$\beta_{\ell\to t} = 1.64 \times 10^{-4}$	$\beta_{\ell\to t} = 1.61 \times 10^{-4}$	$\beta_{\ell\to t} = 2.92 \times 10^{-4}$	$\beta_{\ell\to t} = 3.10 \times 10^{-4}$	$\beta_{\ell\to t} = 4.34 \times 10^{-4}$	$\beta_{\ell\to t} = 3.20 \times 10^{-4}$

Table 7.3: *Regularization parameters used in the blind deconvolution step.*

for the different regularization approaches.

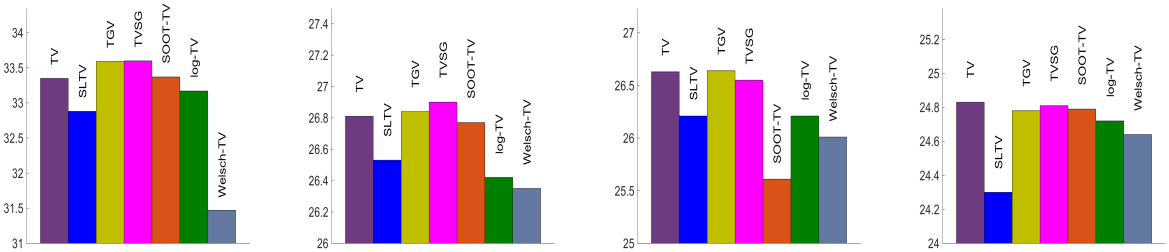
The performance in terms of images quality are much more stable compared to the errors on kernel identification. High SNR scores are usually obtained using the TGV and TVSG regularizations, and in some cases by TV and log-TV. Moreover, the lowest SNR are usually obtained with the SLTV regularization and by some nonconvex regularization such as the Welsch-TV (e.g., kernel (a) for **Foreman** and **Claire** sequences). The regularizations that have achieved the best scores in terms of restoration quality with respect to SNR and MOVIE are displayed in Tables 7.4 and 7.5, respectively. The latter emphasizes the good performance of some nonconvex regularizations. Figures 7.7 and 7.8 show images taken from **Foreman** and **Claire** sequences respectively. We present some frames from the degraded sequences and the corresponding restored frames with the best regularizations in terms of SNR. We can observe from the above figures and tables that the proposed method achieves good performance in terms of video restoration quality, where a gain up to 7 dB is obtained.

Sequences		Kernel (a)	Kernel (b)	Kernel (c)	Kernel (d)
Foreman	Degraded	28.72 dB	22.72 dB	21.58 dB	19.88 dB
	Restored	33.60 dB	26.90 dB	26.64 dB	24.83 dB
		(TVSG)	*(TVSG)*	*(TGV)*	*(TV)*
Claire	Degraded	26.99 dB	22.40 dB	20.49 dB	20.04 dB
	Restored	30.84 dB	28.35 dB	27.67 dB	27.17 dB
		(TVSG)	*(log-TV)*	*(log-TV)*	*(TVSG)*

Table 7.4: *Performance of the best non-blind deconvolution methods in terms of SNR.*

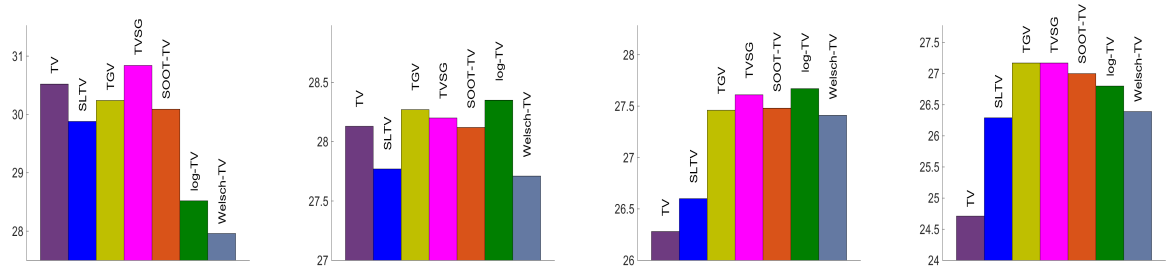
Sequences		Kernel (a)	Kernel (b)	Kernel (c)	Kernel (d)
Foreman	Degraded	2.03×10^{-4}	1.58×10^{-3}	2.41×10^{-3}	3.12×10^{-3}
	Restored	7.7×10^{-5}	4.82×10^{-4}	1.17×10^{-3}	1.83×10^{-3}
		(SOOT-TV)	*(SOOT-TV)*	*(TV)*	*(SOOT-TV)*
Claire	Degraded	2.04×10^{-3}	9.52×10^{-3}	1.18×10^{-2}	1.19×10^{-2}
	Restored	5.28×10^{-4}	2.9×10^{-3}	3.32×10^{-3}	3.67×10^{-3}
		(TVSG)	*(Welsch-TV)*	*(log-TV)*	*(TGV)*

Table 7.5: *Performance of the best non-blind deconvolution methods in terms of MOVIE.*



Kernel (a) *Kernel (b)* *Kernel (c)* *Kernel (d)*

(a) ***Foreman*** *sequence*



Kernel (a) *Kernel (b)* *Kernel (c)* *Kernel (d)*

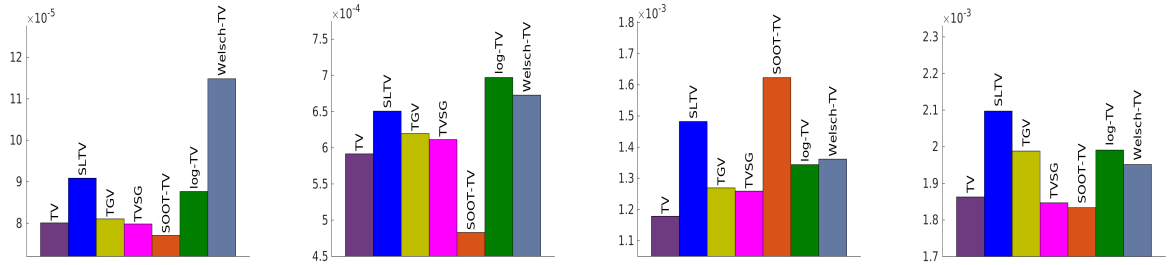
(b) ***Claire*** *sequence*

Figure 7.4: *Performance in terms of SNR with respect to the different regularizations and blur kernels, from left to right: TV, SLTV, TGV, TVSG, SOOT-TV, log-TV, and Welsch-TV.*

7.4.3 Real Data

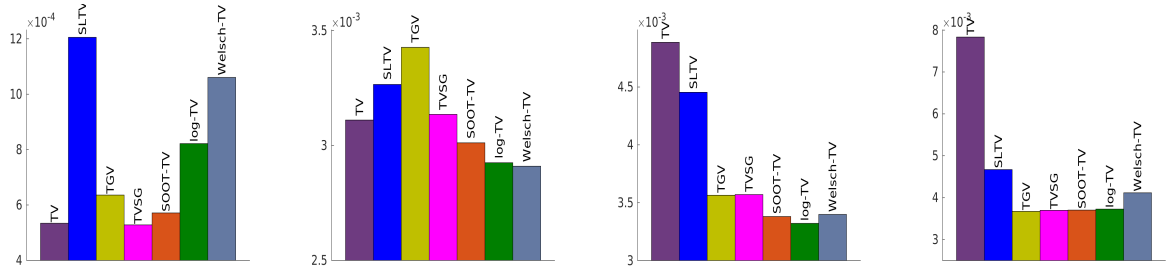
We have applied our blind deconvolution method to the interlaced real sequences **Tachan** and **Au théâtre ce soir** provided by INA. The odd and even fields of each frame are extracted and both blind and non-blind deconvolution stages are performed on them. Once the restored fields are obtained, they are merged in order to reconstruct a deblurred interlaced sequence. The estimated kernels with respect to the different regularization approaches are displayed in Figure 7.6.

Since no ground truth is available for the real sequences, and based on visual inspection on the videos from the blind deconvolution step and the comparison of the regularizations on the synthetic sequences, we select the kernels estimated with SOOT-TV and log-TV regularizations for **Tachan** and **Au théâtre ce soir** sequences respectively, for the non-blind deconvolution step. Moreover, we employ in the second step the spatial regularizations that have achieved the best performance on synthetic data, namely TGV, TVSG and log-TV. Figures 7.9, 7.10, 7.11 and 7.12



Kernel (a) *Kernel (b)* *Kernel (c)* *Kernel (d)*

(a) ***Foreman*** *sequence*



Kernel (a) *Kernel (b)* *Kernel (c)* *Kernel (d)*

(b) ***Claire*** *sequence*

Figure 7.5: *Performance in terms of MOVIE with respect to the different regularizations and blur kernels, from left to right: TV, SLTV, TGV, TVSG, SOOT-TV, log-TV, and Welsch-TV.*

illustrate images taken from the input degraded sequences and the restored ones with the above-listed regularizations. One can notice the enhancement of the sharpness and the visual quality of the restored images, and the attenuation of several artifacts such as the ghost effect in **Au théâtre ce soir** sequence.

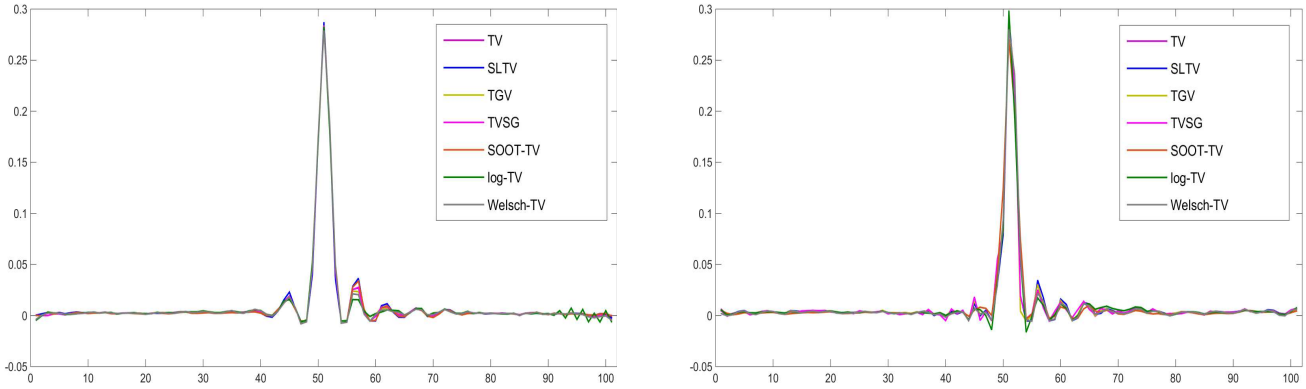
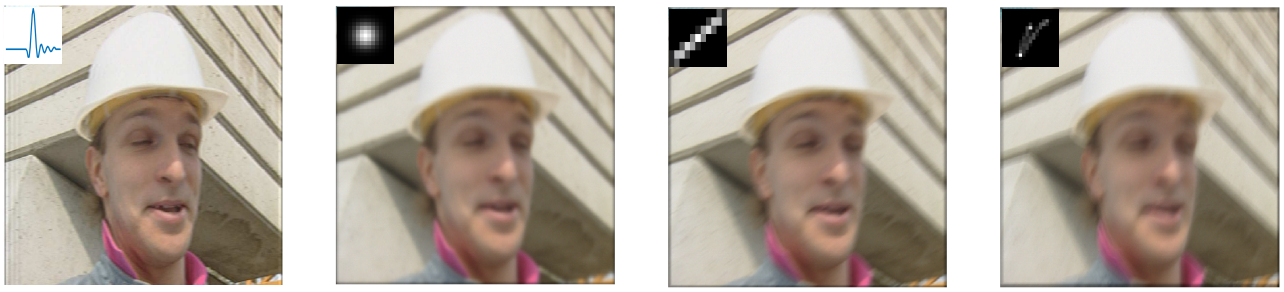


Figure 7.6: *Identified blur kernels ($P = 101$) with the different regularization approaches:* ***Tachan*** *(left),* ***Au théâtre ce soir*** *(right).*



SNR = 28.72 dB *SNR = 22.72 dB* *SNR = 21.58 dB* *SNR = 19.88 dB*

SNR = 33.60 dB *SNR = 26.90 dB* *SNR = 26.64 dB* *SNR = 24.83 dB*

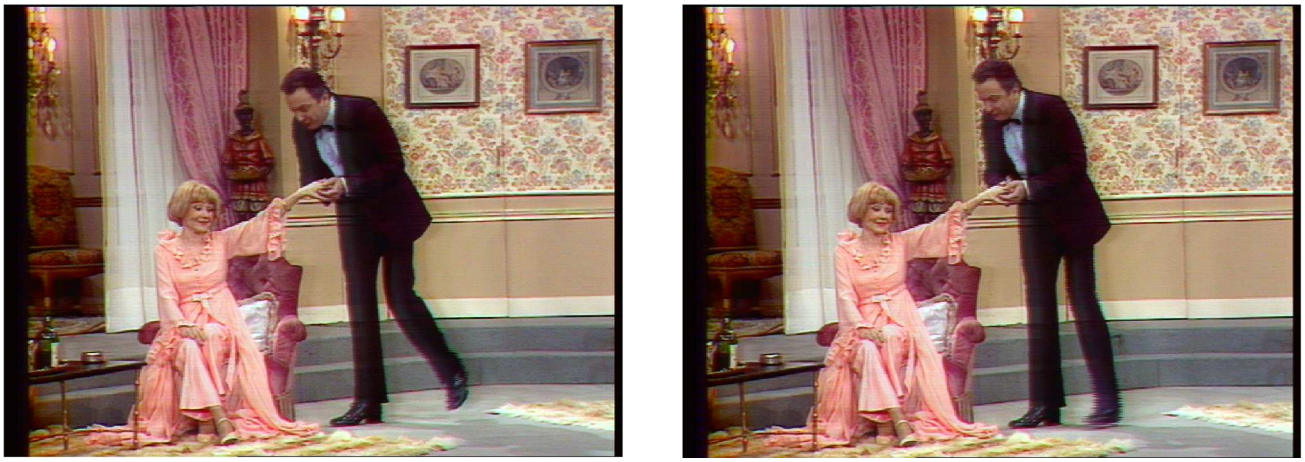
Figure 7.7: ***Foreman*** *sequence: images from the degraded sequence (top), corresponding restored images with the best choice of spatial regularizations in terms of SNR (bottom).*

Figure 7.8: ***Claire*** *sequence: images from the degraded sequence (top), corresponding restored images with the best choice of spatial regularizations in terms of SNR (bottom).*

7.5 Conclusion

We have presented in this chapter a new variational method for blind video deconvolution. Our approach relies on the minimization of a penalized criterion to enhance the restoration quality. Our iterative algorithm alternates between two steps, namely a video estimation step followed by a kernel identification stage, which makes the proposed algorithm well adapted to both blind and non-blind schemes. The versatility of the proposed method allows us to consider a temporal regularization associated with a large number of convex and nonconvex spatial regularization strategies that are usually employed for solving image and video restoration problems. The experimental results on both synthetic and real data revealed that our method achieves good results in both blind and non-blind video deconvolution problems, depending on the chosen spatial regularization and the considered problem (blind/non-blind deconvolution). This approach could be further accelerated through preconditioning and/or an adaptation for an implementation on parallel computing architectures, as seen in the previous chapters.

(a) ***Tachan*** *sequence*



(b) ***Au théâtre ce soir*** *sequence*

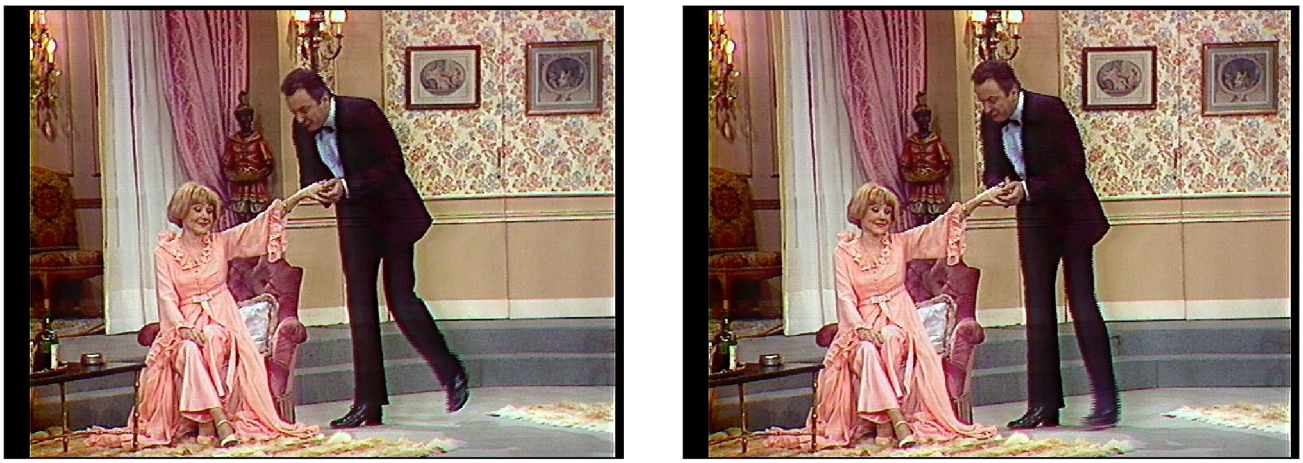
Figure 7.9: *4-th and 9-th frames from the input degraded and interlaced sequences.*

(a) *TGV regularization*

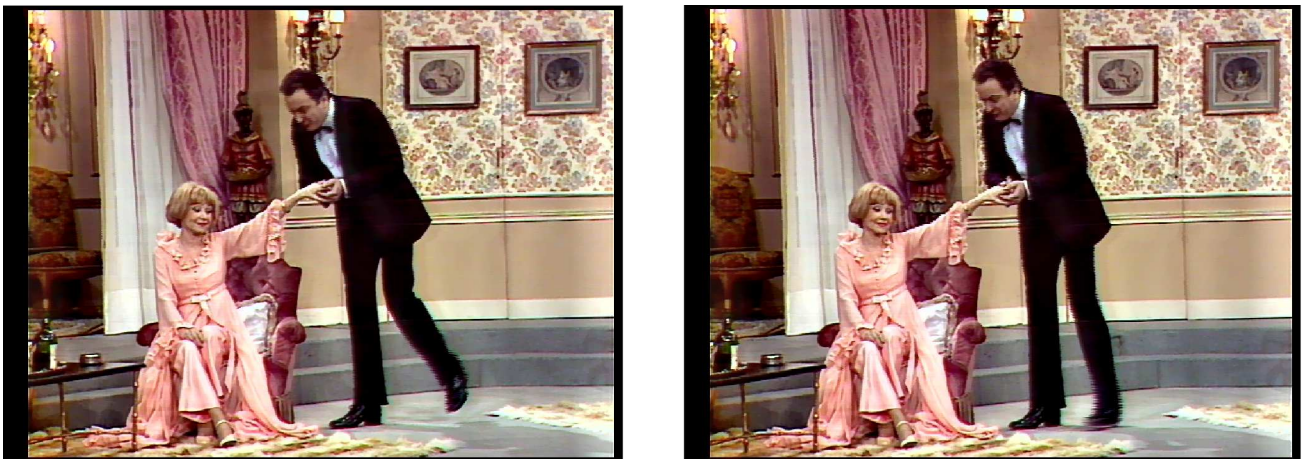
(b) *TVSG regularization*

(c) *log-TV regularization*

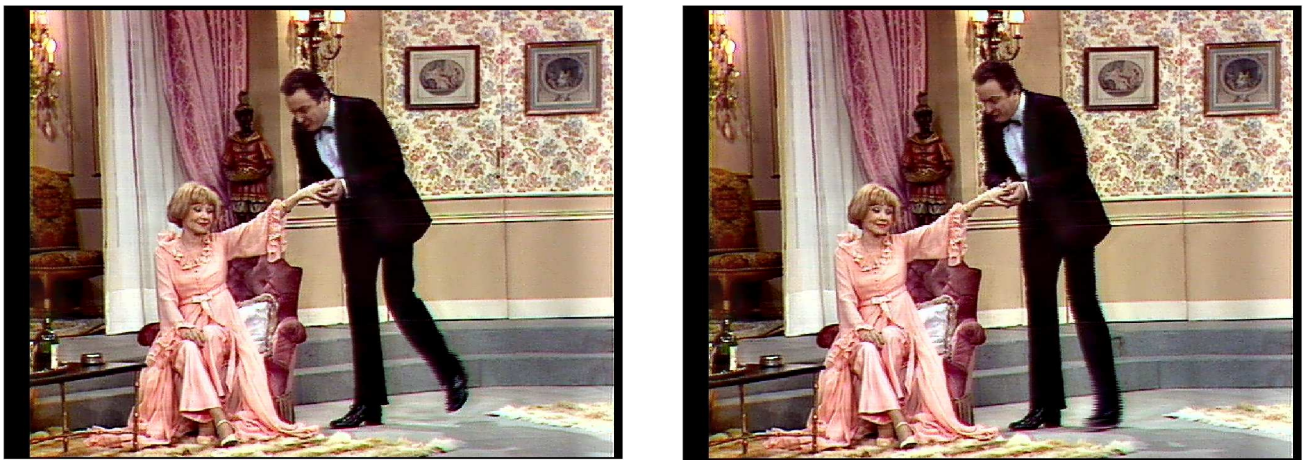
Figure 7.10: ***Tachan*** *sequence: 4-th and 9-th frames from the restored sequences with the best spatial regularizations in non-blind deconvolution.*



(a) *TGV regularization*



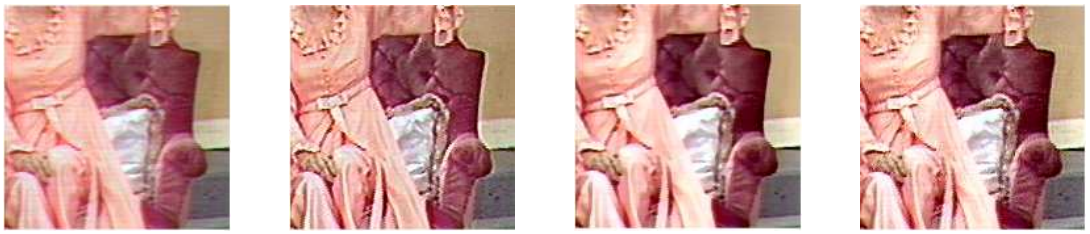
(b) *TVSG regularization*



(c) *log-TV regularization*

Figure 7.11: ***Au théâtre ce soir*** *sequence: 4-th and 9-th frames from the restored sequences with the best spatial regularizations in non-blind deconvolution.*

(a) ***Tachan*** *sequence*



(b) ***Au théâtre ce soir*** *sequence*

Figure 7.12: *Zoom on part of images, from left to right: degraded sequence, restored sequence with TGV, restored sequence with TVSG, restored sequence with log-TV.*

- Chapter 8 -

Conclusion

Contributions

This thesis was dedicated to the proposal of new video restoration methods with a particular emphasis on the contents of old television archives. As we have seen, video restoration issues can be expressed as inverse problems whose solution is obtained by minimizing an objective function composed of several terms with different mathematical properties. Thereby, we proposed in this thesis different optimization methods that can handle efficiently a broad range of video restoration problems.

When the optimization problem is nonsmooth, the proximity operator appears as the appropriate tool. Hence, we have first considered the problem of computing the proximity operator of a sum of convex functions composed with arbitrary linear operators. We have developed a new primal dual splitting algorithm, called dual block-coordinate forward-backward, that resorts to the dual formulation of the problem, to which forward-backward iterations are applied by performing a gradient step on the smooth part of the criterion, and a proximal step on its nonsmooth part. Our algorithm takes advantage of primal-dual methods since it does not require any linear operator inversion, which turns to be of great interest when the latter are of high dimension. Moreover, the proposed algorithm relies on a block-coordinate strategy where at each iteration, only a subset or a "block" of the overall data is processed according to a quasi-cyclic rule, which is very useful from an implementation point of view since it reduces the memory requirements of the algorithm. The convergence properties of the proposed algorithm have been analyzed and established for both generated primal and dual iterates.

The application to a problem of jointly deinterlacing and deblurring video sequences demonstrates the good performance of the proposed algorithm. In the

interlaced television transmission standard, only the odd/even rows of two successive frames are preserved and then merged to reconstruct a single frame. Hence, we aim at recovering the discarded even/odd rows of each frame, while preforming a deconvolution operation. To do so, we have proposed a variational formulation of the problem that relies on both spatial and temporal regularizations. We have compared several variants of the dual block-coordinate algorithm to alternative algorithms in the literature and we have shown that the proposed method achieves satisfactory performance in terms of restoration quality on both synthetic and real videos. In terms of convergence speed, thanks to the block-coordinate approach and the use of preconditioning matrices, a speed up of factor 18 was reached.

In the second part of this thesis, we focused on the development of an asynchronous algorithm based on the dual block-coordinate forward-backward algorithm. The asynchronous algorithm is implemented on a distributed architecture having a given number of independent computing units. In the asynchronous framework, each function composing our objective function is considered as locally related to a single computing unit. However, a consensus constraint has to be imposed in order to ensure the convergence toward an aggregate solution. This consensus is obtained by designing a suitable communication strategy between the computing units, defined by a connected hypergraph. Moreover, in order to maintain good performance of the proposed distributed algorithm, a close attention has been paid to the frequency and the amounts of data to transfer. This requirement is mandatory to achieve a good acceleration factor.
We have implemented our distributed algorithm on a multicore architecture using the *Message-Passing Interface "MPI"* for the management of inter-processors communication. We have designed a communication strategy in-which the images composing the video sequence are equally partitioned over the computing units. These units run the proposed asynchronous algorithm in an independent manner and communicate with neighbouring units only. Numerical experiments on the problem of video sequences denoising show that our algorithm achieves satisfactory and promising results in terms of acceleration gain regarding the number of available computing units.

Finally, we have considered the problem of blind video deconvolution, that consists in estimating both the blur filter and sharp video sequence from an observed and degraded sequence. In this context, the associated inverse problem is blind and severely *ill-posed* so that regularization on both video sequence and convolution kernel are required. We have proposed a formulation as a nonconvex optimization problem and developed an efficient algorithm that alternates between a video estimation step and a kernel identification step until reaching a suitable solution. The major advantage of our method is that it has the ability of handling various recent regularization approaches having different mathematical properties. We have pro-

vided numerical experiments and comparison on synthetic and real video sequences with multiple convolution kernels. We have considered several convex and nonconvex spatial regularization functions and compared their performance on both synthetic and real video sequences.

Perspectives

The work carried out during this thesis opens the door to many perspectives. We will cite a few of them in the following.

Joint blind deconvolution and super-resolution method

We have proposed in Chapter 5 a joint deconvolution and super-resolution method for video sequences, while we have addressed the problem of blind video deconvolution in Chapter 7. An interesting perspective of this thesis would be to develop a combined blind video deconvolution and super-resolution method, that seeks to identify both the blur filter and the high resolution video sequence from the input degraded low resolution video. Most of the works in image and video super-resolution are non-blind, i.e., they assume that the blur kernel is known or can be neglected. This assumption is clearly non realistic since blur always degrades the images during the different images acquisition stages.
The design of such an approach could be based on an extension of our blind deconvolution method, by introducing the downsampling operator S of Chapter 5 to the objective function of Chapter 7, as follows

$$\underset{\boldsymbol{x}\in\mathbb{R}^{TN},h\in\mathbb{R}^{P}}{\text{minimize}} \quad \left(F(\boldsymbol{x},h) = \frac{1}{2}\sum_{t=1}^{T} \|S(h * x_t) - y_t\|^2 + \Psi(\boldsymbol{x}) + \Theta(h) \right),$$

and solving the resulting problem with Algorithm 24 of Chapter 7.

Full color video processing

A classical way to perform deconvolution and/or super-resolution on RGB images or videos is to either process the luminance component only, or to apply the method to the three channels in an independent manner. The luminance component is obtained by converting the RGB video to the YCbCr model, afterwards, the chrominance components can be omitted or softly processed, as we have seen in the thesis where we have applied a small median filter of size 3×3 on the "Cb" and "Cr" channels. Both of these methods are sub-optimal since they do not fully exploit the correlation across the color bands. Thereby, another approach that could improve the restoration results would be to process the three RGB channels in a single step [Chierchia et al., 2014; Mousavi and Monga, 2017]. The underlying model should

include in this case a new regularization term that maintains the correlation between the three channels in the spatial or gradient domains, e.g., using ℓ_1 or ℓ_2 norm.

Motion estimation

We have assumed in this thesis, that the motion fields relating neighbour frames are pre-computed over the input degraded images, for all our numerical experiments on video restoration. This strategy represented a good trade-off between the computation overhead (cost) and the restoration quality of the estimated sequences. However, this remains a rough approximation of the displacements between the frames, hence, we propose as a prospect to include the estimation and the refinement of the motion operators $(M_{\ell \to t})_{(\ell,t)\in\{1,\dots,T\}^2}$ within our method.
Motion fields estimation has been extensively studied in the literature, especially for multi-frames super-resolution [Sroubek and Milanfar, 2012; Héas et al., 2016]. This can be achieved using a block matching technique [Kanj et al., 2015], or by enforcing some non-negative function of $(M_{\ell \to t})_{(\ell,t)\in\{1,\dots,T\}^2}$ to be small, More specially, it assumed that the sought motion operators are sparse in some representation, such as the gradient domain. It should be noted that the underlying optimization problem is nonconvex, and only a convergence towards a local minimum can be guaranteed, because of the coupling term relating the frames and motion operators. The underlying problem could be solved using an alternating strategy similar to the one of Algorithm 24, that would alternate between the video restoration step and the motion operators estimation step.

Further investigation and extension of the distributed algorithm

The work that we have presented in Chapter 6 offers news and various prospects in both practical and theoretical aspects. A short-term perspective would be to evaluate our algorithm on more powerful architectures that are dedicated to parallel computing with a larger volume of data. This will allow to further investigate the acceleration gain provided by our method and its saturation threshold. We may also further optimize our Julia implementation, or rewrite our code with other programming languages such as C++ combined with OpenMP and OpenMPI. Moreover, it would be interesting to employ such a framework in the blind deconvolution scheme. Each computing unit would handle its own set of images and the associated convolution kernel. However, the latter is assumed to be the same for all the images composing the video sequence. Hence, the work would involve the design and the study of an efficient communication strategy, with a suitable synchronization scheme.

List of Figures

List of Tables

Bibliography

F. Abboud, E. Chouzenoux, J.-C. Pesquet, J.-H. Chenot, and L. Laborelli. A hybrid alternating proximal method for blind video restoration. In *IEEE Eur. Signal Process. Conf. (EUSIPCO 2014)*, pages 1811–1815, Lisbon, Portugal, 1-5 Sep. 2014. 74

M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Trans. Image Process.*, 19(9):2345–2356, Sep. 2010. 18

M. Allain, J. Idier, and Y. Goussard. On global and local convergence of halfquadratic algorithms. *IEEE Trans. Image Process.*, 15(5):1130–1142, May 2006. 123

M. S. C. Almeida and M. A. T. Figueiredo. Frame-based image deblurring with unknown boundary conditions using the alternating direction method of multipliers. In *IEEE Int. Conf. Image Process. (ICIP 2013)*, pages 582–585, Melbourne, Australia, 15-18 Sep. 2013. 15

H. Aly and E. Dubois. Image up-sampling using total-variation regularization with a new observation model. *IEEE Trans. Image Process.*, 14(10):1647–1659, Oct. 2005. 25

G. M. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *American Fed. Inform. Process. Soc. (AFIPS 1967)*, pages 483–485, Atlantic City, USA, 18-20 Apr. 1967. 109

H. Attouch and J. Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Math. Program.*, 116(1):5–16, Jun. 2009. 43

H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems : proximal algorithms, forward-backward splitting, and regularized gauss-seidel methods. *Math. Program.*, 137(1):91–129, Feb. 2011. 43

L. Bar, N. Sochen, and N. Kiryati. Variational pairing of image segmentation and blind restoration. In *IEEE Eur. Conf. Comput. Vis. (ECCV 2004)*, pages 166–177, Prague, Czech Republic, 11-14 May 2004. 18

H. H. Bauschke and P. L. Combettes. A Dykstra-like algorithm for two monotone operators. *Pac. J. Optim.*, 4(3):383–391, Sept. 2008. 54, 57

H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, New York, USA, 2nd edition, 2017. 35, 36, 39, 48, 65, 66

E. M. Bednarczuk, A. Jezierska, and K. E. Rutkowski. Inertial proximal best approximation primal-dual algorithm. Technical report, 2016. 47

M. Ben-Ezra and S. K. Nayar. Motion-based motion deblurring. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(6):689–698, Jun. 2004. 18

M. Benning, C. Brune, M. Burger, and J. Muller. Higher-order TV methods enhancement via bregman iteration. *SIAM J. Sci. Comput.*, 54(2):269–310, Feb. 2013. 14

M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. IOP Publishing Ltd., Bristol, U.K., 1998. 8

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, USA, 1999. 38

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. *SIAM J. Optim.*, 23(4):2541–2565, Dec. 2013. 87

J. Bolte and E. Pauwels. Majorization-minimization procedures and convergence of sqp methods for semi-algebraic and tame programs. *Math. Oper. Res.*, 41(2): 442–465, Jan. 2016. 41

J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.*, 146(1-2):459–494, Aug. 2014. 72, 73

J. Bolte, T. P. Nguyen, J. Peypouquet, and B. Suter. From error bounds to the complexity of first-order descent methods for convex functions. *To appear in Math. Program.*, 2016. 66

S. Bonettini, A. Benfenati, and V. Ruggiero. Primal-dual first order methods for total variation image restoration in presence of Poisson noise. In *IEEE Int. Conf. Image Process. (ICIP 2014)*, pages 4156–4160, Paris, France, 27-30 Oct. 2014. 47

S. Bonettini, I. Loris, F. Porta, M. Prato, and S. Rebegoldi. On the convergence of variable metric line-search based proximal-gradient method under the Kurdyka-łojasiewicz inequality. Technical report, May 2016a. https://arxiv.org/abs/1605.03791. 44

S. Bonettini, F. Porta, and V. Ruggiero. A variable metric forward-backward method with extrapolation. *SIAM J. Sci. Comput.*, 38(4):2558–2584, Aug. 2016b. 44

N. Bose, H. Kim, and B. Zhou. Performance analysis of the TLS algorithm for image reconstruction from a sequence of undersampled noisy and blurred frames. In *IEEE Int. Conf. Image Process.*, page 571575, 13-16 Nov., Austin, Texas, USA 1994. 23

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 Mach. Learn.*, 1(8):1–122, Jan. 2011. 54, 86, 87

K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM J. Imaging Sci.*, 3(3):492–526, Sep. 2010. 14, 105, 114

A. Buades, B. Coll, and J.-M. Morel. The staircasing effect in neighborhood filters and its solution. *IEEE Trans. Image Process.*, 15(6):1499–1505, Jun. 2006. 14

J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring using multiple images. *J. Comput. Phys.*, 228(14):5057–5071, Aug. 2009. 28

P. Campisi and K. Egiazarian. *Blind Image Deconvolution: Theory and Applications.* CRC Press, 2007. 17

D. P. Capel and A. Zisserman. Super-resolution from multiple views using learnt image models. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2001)*, pages 627–634, Kauai, HI, USA, 8-14 Dec. 2001. 25

A. S. Carasso. The APEX method in image sharpening and the use of low exponent lévy stable laws. *SIAM J. Appl. Math.*, 63(2):593–618, Dec. 2002. 18

A. S. Carasso. Direct blind deconvolution. *SIAM J. Appl. Math.*, 61(6):1980–2007, May 2011. 18

A. Chakrabarti, A. Rajagopalan, and R. Chellappa. Super-resolution of face images using kernel-based prior. *IEEE Trans. Multimed.*, 9(4):888–892, Jun. 2007. 24

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imag. Vision*, 40(1):120–145, May 2010. 49, 87

A. Chambolle and T. Pock. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *SMAI J. Comput. Math.*, 1:29–54, 2015. 55

R. H. Chan, M. Tao, and X. Yuan. Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers. *SIAM J. Imaging Sci.*, 6(1):680–697, Mar. 2013. 17, 18

T. Chan and C. Wong. Total variation blind deconvolution. *IEEE Trans. Image Process.*, 7(3):370–375, Mar. 1998. 18

T. Chan, A. Marquina, and P. Mulet. High-order total variation-based image restoration. *SIAM J. Sci. Comput.*, 22(2):503–516, Feb. 2000. 14

M. S. Chang, S. W. Yun, and P. Park. PSF search algorithm for dual-exposure type blurred image. *Int. J. Appl. Sci. Eng. Technol.*, 4(1):7–12, 2007. 18

G. Chantas, N. Galatsanos, A. Likas, and M. Saunders. Variational bayesian image restoration based on a product of t-distributions image prior. *IEEE Trans. Image Process.*, 7(10):1795–1805, Oct. 2008. 16

G. H.-G. Chen and R. T. Rockafellar. Convergence rates in forward-backward splitting. *SIAM J. Optim.*, 7(2):421–444, 1997. 42

J. Chen, L. Yuan, C.-K. Tang, and L. Quan. Robust dual motion deblurring. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2008)*, pages 1–8, Anchorage, Alaska, USA, 23-28 Jun. 2008. 20, 21, 28

J. Chen, J. N. Yanez, and A. Achim. Video super-resolution using generalized Gaussian Markov Random Fields. *IEEE Signal Process. Lett.*, 19(2):63–69, Feb. 2012. 24

J.-H. Chenot, J. O. Drewery, and D. Lyon. Restoration of archived television programmes for digital broadcasting. Technical report, Sep. 1998. http://www.bbc.co.uk/rd/publications/rdreport_1998_05. 114

M. C. Chiang and T. E. Boulte. Efficient super-resolution via image warping. *Image Vis. Comput.*, 18(10):761–771, Jul. 2000. 24

G. Chierchia, N. Pustelnik, B. Pesquet-Popescu, and J-C Pesquet. A nonlocal structure tensor-based approach for multicomponent image recovery problems. *IEEE Trans. Image Process.*, 23(12):5531–5544, Dec. 2014. 147

E. Chouzenoux, A. Jezierska, J.-C. Pesquet, and H. Talbot. A majorize-minimize subspace approach for l2-l0 image regularization. *SIAM J. Imaging Sci.*, 6(1): 563–591, Mar. 2013. 15

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 and Appl.*, 162(1):107–132, Jul. 2014. 43, 44, 63

E. Chouzenoux, J.-C. Pesquet, and A. Repetti. A block coordinate variable metric forward-backward algorithm. *J. Global Optim.*, 66(3):457485, Nov. 2016. 45, 55, 64, 66, 73, 125, 129, 130

P. L. Combettes and J.-C. Pesquet. A proximal decomposition method for solving convex variational inverse problems. *Inverse Problems*, 24(8):065014, Dec. 2008.

P. L. Combettes and J.-C. Pesquet. Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping. *SIAM J. Optim.*, 25(2):1221–1248, Jul. 2015. 55, 87

P. .L. Combettes and B. C. Vũ. Variable metric forward-backward splitting with applications to monotone inclusions in duality. *Optimization*, 63(9):1289–1318, Sept. 2014a. 43, 44

P. L. Combettes and B. C. Vũ. Variable metric forward-backward splitting with applications to monotone inclusions in duality. *Optimization*, 63(9):1289–1318, Sept. 2014b. 61

P .L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4(4):1168–1200, Nov. 2005. 42, 43

P. L. Combettes, D. Dung, and B. C. Vũ. Dualization of signal recovery problems. *Set-Valued Var. Anal.*, 18(3):373–404, Dec. 2010. 56, 57

P. L. Combettes, D. Dũng, and B. C. Vũ. Proximity for sums of composite functions. *J. Math. Anal. Appl.*, 380(2):680–688, Aug. 2011. 50, 54, 62, 76, 98, 128

L. Condat. A primal-dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *J. Optim. Theory App.*, 158(2): 460–479, Aug. 2013. 48, 49, 77, 79, 128, 150

L. Condat. Semi-local total variation for regularization of inverse problems. In *IEEE Eur. Signal Process. Conf. (EUSIPCO 2014)*, pages 1806–1810, Lisbon, Portugal, 1-5 Sep. 2014. 14, 69

L. Condat. Discrete Total Variation: New definition and minimization. Technical report, 2016. https://hal.archives-ouvertes.fr/hal-01309685/document. 116

M. Cristani, D. S. Cheng, V. Murino, and D. Pannullo. Distilling information with super-resolution for video surveillance. In *ACM Int. Workshop Video Surveillance Sensor Netw.*, pages 2–11, New York, USA, 10-16 Oct. 2004. 22

J. Y. B. Cruz and T. T. A. Nghia. On the convergence of the forwardbackward splitting method with linesearches. *Optim. Methods Softw.*, 31(6):1209–1238, Aug. 2016. 43

D. Csiba, Z. Qu, and P. Richtárik. Stochastic dual coordinate ascent with adaptive probabilities. In *Int. Conf. on Mach. Learn. (ICML 2015)*, pages 674–683, Lille, France, 1-6 Jul. 2015. 59

S. Dai, M. Han, W. Xu, Y. Wu, and Y. Gong. Soft edge smoothness prior for alpha channel super resolution. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2007)*, pages 1–8, Minneapolis, Minnesota, USA, 18-23 Jun. 2007. 25

S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos. SoftCuts: a soft edge smoothness prior for color image superresolution. *IEEE Trans. Image Process.*, 18(5):969–981, May 2009. 25

A. Danielyan, V. Katkovnik, and K. Egiazarian. BM3D frames and variational image deblurring. *IEEE Trans. Image Process.*, 21(4):1715–1728, Apr. 2012. 17

G. Demoment. Image reconstruction and restoration : Overview of common estimation structure and problems. *IEEE Trans. Acoust. Speech Signal Process.*, 37 (12):2024–2036, Dec. 1989. 8, 10

J. E. Dennis and R. E. Welsch. Techniques for nonlinear least squares and robust regression. *Comm. Statist. Simulation Comput.*, 7(4):345–359, Jan. 1978. 118

A.M. Dhake. *TV and Video Engineering.* McGraw-Hill Education (India) Pvt Limited, 1999. 29, 30

C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *IEEE Eur. Conf. Comput. Vis. (ECCV 2014)*, pages 184–199, Zurich, Switzerland, 6-12 Sept. 2014. 26

M. Elad and A. Feuer. Super-resolution reconstruction of image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(9):817–834, Sep. 1999. 68

M. Elad and Y. Hel-Or. A fast super-resolution reconstruction algorithm for pure translational motion and common space invariant blur. *IEEE Trans. Image Process.*, 10(8):1187–1193, Aug. 2001. 24, 29

M. Elad, P. Milanfar, and R. Ron. Analysis versus synthesis in signal priors. *Inverse Problems*, 23(3):947–968, Jun. 2007. 14, 15

M. A. Elgharib, F. Pitié, and A. C. Kokaram. Blotch and scratch removal in archived film using a semi-transparent corruption model and a ground-truth generation technique. *EURASIP J. image video Process.*, (33):1–20, Dec. 2013. 29

H. Erdogan and J. A. Fessler. Monotonic algorithms for transmission tomography. *IEEE Trans. Med. Imag.,*, 18(9):801–814, Sept. 1999. 41

W. Fan and D. Y. Yeung. Image hallucination using neighbor embedding over visual primitive manifolds. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2007)*, pages 1–7, Minneapolis, Minnesota, USA, 18-23 Jun. 2007. 25

S. Farsiu, D. Robinson, M. Elad, and P. Milanfar. Fast and robust multi-frame super-resolution. *IEEE Trans. Image Process.*, 13(10):1327–1344, Oct. 2004. 24

S. Farsiu, M. Elad, and P. Milanfar. Video-to-video dynamic super-resolution for grayscale and color sequences. *EURASIP J. Adv. Signal Process.*, 2006:1–15, Dec. 2006. 68

R. Fattal. Image upsampling via imposed edge statistics. *ACM Trans. Graph.*, 26 (3):1–8, Jul. 2007. 22, 25

R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and Freeman W. T. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787794, July 2006. 19, 20

M. A. T. Figueiredo and R. D. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Process.*, 12(8):906–916, Aug. 2003. 17

M. A. T. Figueiredo, J. Bioucas-Dias, and R. Nowak. Majorization-minimization algorithms for wavelet-based image restoration. *IEEE Trans. Image Process.*, 16 (12):2980–2991, Dec. 2007. 14

Message Passing Interface Forum. MPI: A Message-Passing Interface standard. Technical report, 1994. 108

P. Frankel, G. Garrigos, and J. Peypouquet. Splitting methods with variable metric for Kurdyka-łojasiewicz functions and general convergence rates. *J. Optim. Theory Appl.*, 165(3):874–900, Jun. 2015. 44

X. Fu, B. He, X. Wang, and X. Yuan. Block wise alternating direction method of multipliers with Gaussian back substitution for multiple block convex programming. Technical report, 2014. http://www.optimization-online.org/DB_FILE/2014/09/4544.pdf. 54

G. B. Giannakis and R. W. Heath Jr. Blind identification of multichannel FIR blur and perfect image restoration. *IEEE Trans. Image Process.*, 9(11):1877–1896, Nov. 2000. 20

G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Model. Simul.*, 7(3):1005–1028, Nov. 2008. 14

T. Goldstein and O. S. Osher. The split Bregman method for l1-regularized problems. *SIAM J. Imag. Sci.*, 2(2):323–343, Apr. 2009. 18

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3 edition, 1996. 76

W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Paral lel Programming with the Message Passing Interface*. MIT Press, 2 edition, 1999. 108

J. Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902. 9

D. K. Hammond, P. Vandergheynst, and R. Gribonval. Wavelets on graphs via spectral graph theory. *Appl. Comp. Harm. Anal.*, 30(2):129–150, Mar. 2011. 8

G. Harikumar and Y. Bresler. Exact image deconvolution from multiple FIR blurs. *IEEE Trans. Image Process.*, 8(6):846–862, Jun. 1999a. 20, 21

G. Harikumar and Y. Bresler. Perfect blind restoration of images blurred by multiple filters: Theory and efficient algorithms. *IEEE Trans. Image Process.*, 8(2):202–219, Feb. 1999b. 20

K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *IEEE Eur. Conf. Comput. Vis. (ECCV 2014)*, pages 346–361, Zurich, Switzerland, 6-12 Sept. 2014. 26

P. Héas, A. Drémeau, and C. Herzet. An efficient algorithm for video superresolution based on a sequential model. *SIAM J. Imaging Sci.*, 9(2):537–572, Jan. 2016. 68, 148

D. R. Hunter and K. Lange. A tutorial on mm algorithms. *Amer. Stat.*, 58(1):30–37, Feb. 2004. 41

M. Irani and S. Peleg. Super-resolution from image sequences. In *IEEE Conf. Pattern Recognt. (ICPR 1990)*, pages 115–120, Atlantic Citys, USA, 16-21 Jun. 1990. 22

F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem. Explicit convergence rate of a distributed alternating direction method of multipliers. *IEEE Trans. Autom. Control*, 61(4):892–904, Apr. 2016. 86

M. W. Jacobson and J. A. Fessler. An expanded theoretical treatment of iteration-dependent majorize-minimize algorithms. *IEEE Trans. Image Process.*, 16(10): 2411–2422, Oct. 2007. 40

M. Jaggi, V. Smith, M. Takác, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan. Communication-efficient distributed dual coordinate ascent. In *Adv. Neural Inf. Process. Syst. (NIPS 2014)*, pages 3068–3076. Montréal, Canada, 8-13 Dec. 2014. 55

D. Janakiram, M. Venkateswara Reddy, A. Vijay Srinivas, M. A. Maluk Mohamed, and S. Santosh Kumar. GDP: A paradigm for intertask communication in grid computing through distributed pipes. In *Int. Conf. Distrib. Comput. Internet Technol. (ICDCIT 2005)*, pages 235–241, Bhubaneswar, India, 22-24 Dec. 2005. 109

P. Jansson. *Deconvolution of Images and Spectra*. Academic Press, San Diego, California, USA, 2 edition, 1997. 8

K. Jensen and D. Anastassiou. Digitally assisted deinterlacing for EDTV. *IEEE Trans. Circuits Syst. Video Tech.*, 3(2):99–106, Apr. 1993. 68

M. V. Joshi, S. Chaudhuri, and R. Panuganti. Super-resolution imaging: use of zoom as a cue. *Image Vis. Comput.*, 22(14):1185–1196, Dec. 2004. 24

N. Joshi, R. Szeliski, and D. J. Kriegman. Psf estimation using sharp edge prediction. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2008)*, pages 1–8, Anchorage, Alaska, USA, 23-28 Jun. 2008. 19

A. Kanj, H. Talbot, J.-C. Pesquet, and R. Rodriguez Luparello. Color deflickering for high-speed video in the presence of artificial lighting. In *IEEE Int. Conf. Image Process. (ICIP 2015)*, pages 976–980, Quebéc City, Canada, 27-30 Sep. 2015. 148

S. H. Keller. *Video Upscaling Using Variational Methods*. PhD thesis, The Image Group, Department of Computer Science Faculty of Science, University of Copenhagen, 2007. 68

S. H. Keller, F. Lauze, and M. Nielsen. A total variation motion adaptive deinterlacing scheme. In *Int. Conf., Scale-Space 2005*, pages 408–418, Hofgeismar, Germany, 7-9 Apr. 2005. 68

S. Kim, N. Bose, and H. M. Valenzuela. Recursive reconstruction of high resolution image from noisy undersampled multiframes. *IEEE Trans. Acoust. Speech Signal Process.*, 38(6):1013–1027, Jun. 1990. 23

T. H. Kim and K. M. Lee. Generalized video deblurring for dynamic scenes. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2015)*, pages 5426–5434, Boston, Massachusetts, USA, 7-12 Jun. 2015. 28

A. C. Kokaram. *Motion Picture Restoration: Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*. Springer-Verlag, London, UK, UK, 1998. 29, 114

A. C. Kokaram and S. J. Godsill. MCMC for joint noise reduction and missing data treatment in degraded video. *IEEE Transactions on Signal Processing*, 50 (2):189–205, Feb. 2002. 114

N. Komodakis and N. Paragios. MRF-based blind image deconvolution. In *Asian Conf. Comput. Vision (ACCV 2012)*, pages 361–374, Daejeon, North Korea, 5-9 Nov. 2012. 20

N. Komodakis and J.-C. Pesquet. Playing with duality : An overview of recent primal-dual approaches for solving large-scale optimization problems. *IEEE Signal Process. Mag.*, 32(6):31–54, Nov. 2015. 47, 48, 87

N. Komodakis, B. Xiang, and N. Paragios. A framework for efficient structured max-margin learning of high-order mrf models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(7):1425–1441, Jul. 2015. 86

N. Komodakis, M. Pawan Kumar, and N. Paragios. (hyper)-graphs inference through convex relaxations and move making algorithms: Contributions and applications in artificial vision. *Found. Trends Comput. Graph. Vision*, 10(1):1–102, May 2016. 86

J. Kotera, F. Šroubek, and P. Milanfar. Blind deconvolution using alternating maximum a posteriori estimation with heavy-tailed priors. *Comput. Anal. Images Patterns*, 8048(2013):59–66, Aug. 2013. 20

D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2011)*, pages 233–240, Colorado Springs, CO, USA, 21-25 Jun. 2011. 15, 19, 20, 74

A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *Adv. Neural Inf. Process. Syst. (NIPS 2012)*, pages 1097–1105. Curran Associates, Inc., 2012. 26

A. S. Krylov and V. A. Nasonov. Adaptive image deblurring with ringing control. In *IEEE Int. Conf. Image Graph. (ICIG 2009)*, pages 72–75, Xian, Shanxi, China, 20-23 Sept. 2009. 18

K. Lange. *Numerical analysis for statisticians*. Springer, New York, USA, 2010. 41

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, Dec. 1989. 26

I. H. Lee, N. K. Bose, and C. W. Lin. Locally adaptive regularized super-resolution on video with arbitrary motion. In *IEEE Int. Conf. Image Process. (ICIP 2010)*, pages 897–900, Hong Kong, 26-29 Sept. 2010. 24

A. Levin. Blind motion deblurring using image statistics. In *Adv. Neural Inf. Process. Syst. (NIPS 20007)*, pages 841–848. Vancouver, B.C., Canada, 3-6 Dec. 2007. 19

A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 27(3):1–10, Jul. 2007. 18

A. Levin, Y. Weiss, F. Durand, and W. T. . Freeman. Understanding and evaluating blind deconvolution algorithms. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2009)*, pages 1964–1971, Miami, Florida, USA, 20-26 Jun. 2009. 19

B. Li and H. Chang. Aligning coupled manifolds for face hallucination. *IEEE Signal Process. Lett.*, 16(11):957–960, Nov. 2009. 22

F. Li, X. Jia, and D. Fraser. Universal HMT based super resolution for remote sensing images. In *IEEE Int. Conf. Image Process. (ICIP 2008)*, pages 333–336, San Diego, CA, USA, 12-15 Oct. 2008. 22

H. Li, Y. Zhang, J. Sun, and D. Gong. Joint motion deblurring with blurred/noisy image pair. In *IEEE Conf. Pattern Recognt. (ICPR 2014)*, pages 1020–1024, Stockholm, Sweden, 24-28 Aug. 2014. 20

X. Li and M. T. Orchard. New edge-directed interpolation. *IEEE Trans. Image Process.*, 10(10):1521–1527, Oct. 2001. 25

X. Li, K. M. Lam, G. Qiu, L. Shen, and S. Wang. Example-based image super-resolutionwith class-specific predictors. *J. Vis. Commun. Image Represent.*, 20 (5):312–322, Jul. 2009. 26

Y. Li, S. B. Kang, N. Joshi, S. M. Seitz, and D. P. Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2010)*, pages 2424–2431, San Francisco, CA, USA, 13-18 Jun. 2010. 28

R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia. Video super-resolution via deep draft-ensemble learning. In *IEEE Int. Conf. Comput. Vis. (ICCV 2015)*, pages 531–539, Santiago, Chile, 11-18 Dec. 2015. 29

C. Liu and D. Sun. A bayesian approach to adaptive video super resolution. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2011)*, pages 209–216, Colorado Springs, CO, USA, 21-25 Jun. 2011. 29

C. Liu and D. Sun. On bayesian adaptive video super resolution. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(2):346–360, Feb. 2014. 29

C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR 2008)*, pages 1–8, Anchorage, Alaska, USA, 23-28 Jun. 2008. 74, 130

X. Liu and A. Gamal. Simultaneous image formation and motion blur restoration via multiple capture. In *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP 2001)*, pages 1841–1844, Utah, USA, 7-11 May 2001. 18

S. Lojasiewicz. Une propriété topologique des sous-ensembles analytiques réels. *Editions du centre National de la Recherche Scientifique*, pages 87–89, 1963. 44

D. A. Lorenz, S. Wenger, F. Schöpfer, and M. A. Magnor. A sparse Kaczmarz solver and a linearized Bregman method for online compressed sensing. In *IEEE Int. Conf. Image Process. (ICIP 2014)*, pages 1347–1351, Paris, France, 27-30 Oct. 2014. 59

X. Lu, H. Yuan, P. Yan, Y. Yuan, and X. Li. Geometry constrained sparse coding for single image super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2012)*, pages 1648–1655, Providence Rhode Island, 16-21 Jun. 2012. 25

L. Lucy. An iterative technique for the rectification of observed distributions. *The Astron. J.*, 79(6):745–754, Jun. 1974. 17

S. Ma. Alternating proximal gradient method for convex minimization. *J. Sci. Comput.*, 68(2):546–572, Aug. 2016. 128

X. Ma, J. Zhang, and C. Qi. Position-based face hallucination method. In *IEEE Int.Conf. Multimedia Expo. (ICME 2009)*, pages 290–293, New York, USA, 28 Jun. - 03 Jul. 2009. 22

Z. Ma, R. Liao, X. Tao, L. Xu, J. Jia, and E. Wu. Handling motion blur in multi-frame super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2015)*, pages 5224–5232, Boston, Massachusetts, USA, 7-12 Jun. 2015. 29

J. Maintz and M. Viergever. A survey of medical image registration. *Med. Image Anal.*, 2(1):1–36, Mar. 1998. 22

S. Mallat. Super resolution bandlet upconversion for HD TV. Technical report, 2006. http://www.di.ens.fr/∼mallat/papiers/whitepaper.pdf. 30

S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 3rd edition, 2008. 14

C. Mancas-Thillou and M. Mirmehdi. Super-resolution text using the Teager filte. In *Int.Workshop Camera-Based Document Anal. and Recognt.*, page 1016, Seoul, Korea, 29 Aug. 2005. 24

E. Mjolsness. *Neural networks, pattern recognition, and fingerprint hallucination.* PhD thesis, California Institute of Technology, Pasadena, CA, USA, Sept. 1985. 22, 25

J. J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965. 37

H.S. Mousavi and V. Monga. Sparsity-based color image super resolution via exploiting cross channel constraints. *IEEE Trans. Image Process.*, pages 1–13, May 2017. 147

V. Naranjo and A. Albiol. Flicker reduction in old films. In *IEEE Int. Conf. Image Process. (ICIP 2000)*, volume 2, pages 657–659, Vancouver, Canada, 10-13 Sept. 2000. 114

V. Naranjo, S. Goméz, A. Albiol, and J. M. Massi. Grain noise reduction using the human visual system characteristics. In *IEEE Eur. Signal Process. Conf. (EUSIPCO 2014)*, pages 1–5, Florence, Italy, 4-8 Sep. 2006. 29

H. Nasir, V. Stankovic, and S. Marshall. Singular value decomposition based fusion for super-resolution image reconstruction. *Signal Process. Image Commun.*, 27 (2):180–191, Feb. 2012. 24

A. V. Nasonov and A.S. Krylov. Fast super-resolution usingweighted median filtering. In *Int.Conf. on Pattern Recognt. (ICPR 2010)*, pages 2230–2233, Istanbul, Turkey, 23-26 Aug. 2010. 24

N. Nguyen, P. Milanfar, and G. Golub. Blind super-resolution with generalized cross-validation using gauss-type quadrature rules. In *Asilomar Conf. Signals Systems Comput.*, pages 1257–1261, Pacific Grove, CA, USA, 24-27 Oct. 1999. 23

M. Nikolova. Weakly constrained minimization: Application to the estimation of images and signals involving constant regions. *J. Math. Imaging Vision*, 21(2): 155–175, Sep. 2009. 14, 116

P. Ochs, A. Dosovitskiy, T. Brox, and T. Pock. On iteratively reweighted algorithms for non-smooth non-convex optimization in computer vision. *SIAM J. Imaging Sci.*, 8(1):331–372, Feb. 2015. 41

J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Adaptive total variation image deblurring: A majorization-minimization approach. *Signal Process.*, 89(9): 1683–1693, Sept. 2009. 17

A. Onose, R. E. Carrillo, J.D. McEwen, and Y. Wiaux. A randomised primal-dual algorithm for distributed radio-interferometric imaging. In *IEEE Eur. Signal Process. Conf. (EUSIPCO 2016)*, pages 1448–1452, Budapest Hungary, 29 1ug.-2 Sep. 2016. 55

J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, New York, USA, 1970. 40

N. Paragios, Y. Chen, and O. Faugeras. *Handbook of Mathematical Models in Computer Vision*. Springer Science & Business Media, 2006. 28

S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE Signal Process. Mag.*, 20(3):21–36, May 2003. 21

S. Peleg and Y. Yeshurun. Superresolution in MRI: application to human white matter fiber tract visualization by diffusion tensor imaging. *Magn. Reson. Med.*, 45(1):29–35, Jan. 2001. 22

D. Perrone and P. Favaro. A logarithmic image prior for blind deconvolution. *Int. J. Comput. Vis.*, 17(2):159–172, Apr. 2016. 15, 114, 118

J.-C. Pesquet and A. Repetti. A class of randomized primal-dual algorithms for distributed optimization. *J. Nonlinear Convex Anal.*, 16(12):2453–2490, Dec. 2015. 55, 86, 87, 93

L. Pickup, S. Roberts, and A. Zisserman. A sampled texture prior for image super-resolution. In *Adv. Neural Inf. Process. Syst. (NIPS 2003)*, pages 1587–1594, 2003. 24

T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *IEEE Conf. Comput. Vis. Pattern Recogn. (CVPR 2009)*, pages 810–817, Miami, Florida, USA, 20-26 Jun. 2009. 49

T. Pock, D. Cremers, H. Bischof, and Chambolle A. Global solutions of variational models with convex regularization. *SIAM J. Imaging Sci.*, 3(4):1122–1145, Dec. 2010. 8

M. Protter and M. Elad. Super-resolution with probabilistic motion estimation. *IEEE Trans. Image Process.*, 18(8):1899–1904, Aug. 2009. 24

P. Purkait and B. Chanda. Super resolution image reconstruction through bregman iteration using morphologic regularization. *IEEE Trans. Image Process.*, 21(9): 4029–4039, Sept. 2012. 24

N. Pustelnik, C. Chaux, J.-C. Pesquet, and C. Comtat. Parallel algorithm and hybrid regularization for dynamic PET reconstruction. In *IEEE Med. Imaging Conf.*, pages 2423–2427, Knoxville, Tennessee, USA, 30 Oct. - 6 Nov. 2010. 14

N. Pustelnik, C. Chaux, and J.-C. Pesquet. Parallel proximal algorithm for image restoration using hybrid regularization. *IEEE Trans. Image Process.*, 20(9):2450–2462, Sep. 2011. 87

N. Pustelnik, A. Benazza-Benhayia, Y. Zheng, and J.-C. Pesquet. Wavelet-based image deconvolution and reconstruction. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 2016. 14

Z. Qu, P. Richtárik, and T. Zhang. Randomized dual coordinate ascent with arbitrary sampling. In *Adv. Neural Inf. Process. Syst. (NIPS 2015)*, pages 865–873. Montréal, Canada, 7-12 Dec. 2015. 55

A. Repetti, M. Q. Pham, L. Duval, E. Chouzenoux, and J.-C. Pesquet. Euclid in a taxicab: Sparse blind deconvolution with smoothed l1/l2 regularization. *Signal Process. Lett.*, 22(5):539–543, May 2015. 15, 114, 118, 123

W. H. Richardson. Bayesian-based iterative method of image restoration. *J. Optical Soc. Amer.*, 62(1):55–59, Jan. 1972. 17

P. Richtárik and M. Takác. Efficient serial and parallel coordinate descent methods for huge-scale truss topology design. In *Int. Conf. Oper. Res. (OR 2011)*, pages 27–32, Zurich, Switzerland, 30 Aug.- 2 Sept. 2011. 55

P. Richtárik and M. Takác. Distributed coordinate descent method for learning with big data. *J. Mach. Learn. Res.*, 17(75):1–25, Feb. 2016. 86, 93

R. T. Rockafellar. *Conjugate Duality and optimization*. SIAM, 1974. 35, 36

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.*, 14(5):877–898, Aug. 1976. 39

Y. Romano, J. Isidoro, and P. Milanfar. Raisr: Rapid and accurate image super resolution. *IEEE Trans. Comput. Imaging*, pages 1–17, 2016. DOI: 10.1109/TCI.2016.2629284. 26

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1):259–268, Nov. 1992. 13, 17, 114, 116

A. Saha and A. Tewari. On the finite time convergence of cyclic coordinate descent methods. *SIAM J. Optim.*, 23(1):576–601, 2013. 55

S. Salzo. The variable metric forward-backward splitting algorithm under mild differentiability assumptions. Technical report, 2016. 43

C. A. Segall, A. K. Katsaggelos, R. Molina, and J. Mateos. Bayesian resolution enhancement of compressed video. *IEEE Trans. Image Process.*, 13(7):898–910, Jul. 2004. 25

I. Selesnick and M. A. T. Figueiredo. Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors. In *SPIE Wavelets XIII*, volume 7446, page 15, Aug. 2009. 14, 18

P. Sen and S. Darabi. Compressive image super-resolution. In *Asilomar Conf. Signals Systems Comput.*, pages 1235–1242, Pacific Grove, CA, USA, 1-4 Nov. 2009. 25

K. Seshadrinathan and A. C. Bovik. Motion tuned spatio-temporal quality assessment of natural videos. *IEEE Trans. Image Process.*, 19(2):335–350, Feb. 2010. 74, 134

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *J. Mach. Learn. Res.*, 14(1):567–599, Feb. 2013. 55

Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph.*, 27(3):1–10, Aug. 2008. 19, 20

H. Shekarforoush, M. Berthod, J. Zerubia, and M. Werman. Subpixel Bayesian estimation of albedo and height. *Int. J. Comput. Vis.*, 19(3):289–300, Aug. 1996. 24

H. F. Shen, L. P. Zhang, B. Huang, and P. X. Li. A MAP approach for joint motion estimation, segmentation, and super-resolution. *IEEE Trans. Image Process.*, 16 (2):479490, Feb. 2007. 24

W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2016)*, pages 1874–1883, Las Vegas, USA, 26 Jun.- 1 Jul. 2016. 27

M. Shimizu, S. Yoshimura, M. Tanaka, and M. Okutomi. Super-resolution from image sequence under influence of hot-air optical turbulence. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2008)*, pages 1–8, Anchorage, Alaska, USA, 23-28 Jun. 2008. 24

Y. Song, E.-H. Djermoune, J. Chen, C. Richard, and D. Brie. Online deconvolution for pushbroom hyperspectral imaging systems. Technical report, 2017. 20

F. Sroubek and P. Milanfar. Robust multichannel blind deconvolution via fast alternating minimization. *IEEE Trans. Image Process.*, 21(4):1687–1700, Apr. 2012. 20, 21, 28, 148

T. A. Stephenson and T. Chen. Adaptive markov random fields for example-based super-resolution of faces. *EURASIP J. Adv. Signal Process.*, 2006:1–11, Jan. 2006. 25

A. J. Tatem, H. G. Lewis, P. M. Atkinson, and M. S. Nixon. Super-resolution target identification from remotely sensed images using a Hopfield neural network. *IEEE Trans. Geosci. Remote Sens.*, 39(4):781–796, Apr. 2001. 22

A. M. Tekalp, M. Ozkan, and M. Sezan. High-resolution image reconstruction from lower-resolution image sequences and spacevarying image restoration. In *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP 1992)*, pages 169–172, San Francisco, CA, USA, 23-26 Mar. 1992. 23

C. M. Thillou and M. Mirmehdi. An introduction to super-resolution text. *Adv. Pattern Recognit.*, pages 305–327, Mar. 2007. 24

A. N. Tikhonov. On the stability of inverse problems. *Dokl. Akad. Nauk SSSR*, 39 (5):195–198, Jan. 1943. 12

A. N. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. Winston, Washington, DC, USA, 1977. 12

B. C. Tom and A. K. Katsaggelos. Reconstuction of a high-resolution image by simultaneous registration, restoration and interpolation of low-resolution images. In *IEEE Int. Conf. Image Process. (ICIP 1995)*, pages 539–542, Washington D.C. USA, 23-26 Oct. 1995. 25

R. Tsai and T. Huang. Multiple frame image restoration and registration. In *Adv. Comput. Vis. Image Process.*, pages 317–339, Greenwich, England, 1984. 22, 23

B. C. Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.*, 38(3):667–681, Apr. 2013. 48

C. Vogel and M. Oman. Fast, robust total variation-based reconstructionof noisy, blurred image. *IEEE Trans. Image Process.*, 7(6):813–824, Jun. 1999. 17

C. G. R. Wallis, Y. Wiaux, and J. D. McEwen. Sparse image reconstruction on the sphere: Analysis and synthesis. *IEEE Trans. Image Process.*, 26(11):5176–5187, Nov. 2017. 14

Y. Wang, L. Wang, H. Wang, and W. Li. End-to-end image super-resolution via deep and shallow convolutional networks. Technical report, 2016. https://arxiv.org/pdf/1607.07680v1.pdf. 27

Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4): 600–612, Apr. 2004. 74

Y.-L. You and M. Kaveh. Fourth-order partial differential equations for noise removal. *IEEE Trans. Image Process.*, 9(10):1723–1730, Oct. 2000. 14

Q. Yuan, L. Zhang, and H. Shen. Multiframe super-resolution employing a spatially weighted total variationmodel. *IEEE Trans. Circuits Syst. Video Technol.*, 22(3): 379–392, Mar. 2012. 24

H. Zhang, L. Zhang, and H. Shen. A super-resolution reconstruction algorithm for hyper spectral images. *Signal Process.*, 92(9):2082–2096, Sept. 2012. 22

L. Zhang, H. Zhang, H. Shen, and P. Li. A super-resolution reconstruction algorithm for surveillance images. *Signal Process.*, 90(3):848–859, Mar. 2010. 22

W. Zhao and H. S. Sawhney. Is super-resolution with optical flow feasible? In *Eur. Conf. Comput. Vis.*, pages 599–613, Copenhagen, Denmark, 2831 May 2002. 24

F. Zhou, W. Yang, and Q. Liao. Interpolation-based image superresolution using multisurface fitting. *IEEE Trans. Image Process.*, 21(7):3312–3318, Jul. 2012. 29

F. Zhou, T.-S. Xia, and Q. Liao. Nonlocal pixel selection for multisurface fitting-based super-resolution. *IEEE Trans. Circuits Syst. Video Technol.*, 24(12):2013–2017, Dec. 2014. 29

S. Zhuo, D. Guo, and T. Sim. Robust flash deblurring. In *IEEE Conf. Comput. Vis. Pattern Recognt. (CVPR 2010)*, pages 2440–2447, San Francisco, CA, USA, 13-18 Jun. 2010. 20

W. W. W. Zou and P. C. Yuen. Very low resolution face recognition problem. *IEEE Trans. Image Process.*, 21(1):327–340, Jan. 2012. 22, 25