INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Efficient schemes for total variation minimization under constraints in image processing

Pierre Weiss — Gilles Aubert — Laure Blanc-Féraud

N° 6260

July 2007

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6260--FR+ENG

Efficient schemes for total variation minimization under constraints in image processing

Pierre Weiss , Gilles Aubert , Laure Blanc-Féraud

Thème COG — Systèmes cognitifs
Projets Ariana

Rapport de recherche n° 6260 — July 2007 — 36 pages

Abstract: This paper presents new algorithms to minimize total variation and more generally l^1-norms under a general convex constraint. The algorithms are based on a recent advance in convex optimization proposed by Yurii Nesterov [34]. Depending on the regularity of the data fidelity term, we solve either a primal problem, either a dual problem. First we show that standard first order schemes allow to get solutions of precision ϵ in $O(\frac{1}{\epsilon^2})$ iterations at worst. For a general convex constraint, we propose a scheme that allows to obtain a solution of precision ϵ in $O(\frac{1}{\epsilon})$ iterations. For a strongly convex constraint, we solve a dual problem with a scheme that requires $O(\frac{1}{\sqrt{\epsilon}})$ iterations to get a solution of precision ϵ. Thus, depending on the regularity of the data term, we gain from one to two orders of magnitude in the convergence rates with respect to standard schemes. Finally we perform some numerical experiments which confirm the theoretical results on various problems.

Key-words: l^1-norm minimization , total variation minimization, l^p-norms, duality, gradient and subgradient descents, Nesterov scheme, bounded and non-bounded noises, texture+geometry decomposition, complexity.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : +33 4 92 38 77 77 — Télécopie : +33 4 92 38 77 65

Algorithmes efficaces pour la minimisation de la variation totale sous contraintes

Résumé : Ce papier présente de nouveaux algorithmes pour minimiser la variation totale, et plus généralement des normes l^1, sous des contraintes convexes. Ces algorithmes proviennent d'une avancée récente en optimisation convexe proposée par Yurii Nesterov. Suivant la régularité de l'attache aux données, nous résolvons soit un problème primal, soit un problème dual. Premièrement, nous montrons que les schémas standard de premier ordre permettent d'obtenir des solutions de précision ϵ en $O(\frac{1}{\epsilon^2})$ itérations au pire des cas. Pour une contrainte convexe quelconque, nous proposons un schéma qui permet d'obtenir une solution de précision ϵ en $O(\frac{1}{\epsilon})$ itérations. Pour une contrainte fortement convexe, nous résolvons un problème dual avec un schéma qui demande $O(\frac{1}{\sqrt{\epsilon}})$ itérations pour obtenir une solution de précision ϵ. Suivant la contrainte, nous gagnons donc un à deux ordres dans la rapidité de convergence par rapport à des approches standard. Finalement, nous faisons quelques expériences numériques qui confirment les résultats théoriques sur de nombreux problèmes.

Mots-clés : norme l^1, minimisation variation totale, bruits bornés, bruits de compression, décomposition texture + géométrie, dualité, sous-gradient projeté, complexité, contraintes convexes

1 Introduction

In this paper we are interested in the fast resolution of a class of image restoration and decomposition problems that can be written under the general constrained form :

$$\inf_{u \in \mathbb{R}^n, F(u) \leq \alpha} ||Bu||_1 \tag{1.1}$$

or the "equivalent" Lagrangian form :

$$\inf_{u \in \mathbb{R}^n} ||Bu||_1 + \gamma F(u) \tag{1.2}$$

$\mathbb{R}^n$ is the discrete space of $2D$ images (n is the number of pixels). The results we present apply to any linear operator B. In this paper we concentrate on the use of $B = \nabla$, $||Bu||_1$ corresponding to the total variation (see the appendix for the discretization of differential operators). F is a convex proper function. We will pay a particular attention to functions that write $F(u) = |\lambda(Au - f)|_p$ where A is a linear invertible transform (identity, wavelet transform, convolution,...), $\lambda \in [0, \infty]^n$ is a possibly varying parameter, $p \in \{1, 2, \infty\}$ and f is an observed datum.

This formalism covers a wide range of useful applications. Some of them are listed in the following table :

	$p = 1$	$p = 2$	p=∞
$A = Identity$	Impulse noise denoising, Image decomposition [2, 35, 11, 15]	Gaussian noise denoising [41]	Bounded noise denoising [45]
$A = Convolution$	Robust deconvolution [19]	Image deconvolution [40, 12, 6]	No known reference
A = *Wavelet, local cosine transform*	Image decomposition or denoising [16]	Image denoising (No known reference)	Compression noise denoising [3, 43, 13]

This formalism also allows to do image zooming [29] using the Fourier transform or image cartoon + texture decompositions [30].

Considering pseudo-invertible transforms the preceding formalism would include other interesting applications like denoising using redundant transforms (dictionnaries, curvelets, ...) [26, 8] or image decompositions [42].

As we see the total variation is widely used in many image processing models. This is certainly due to its good theoretical properties and practical results. The difficulty in minimizing it lies in the non differentiability of the l^1-norm. It makes it a challenging task to design *efficient* numerical methods. This is very important for image processing applications which involve huge dimension problems. A lot of different techniques have been proposed. Some are PDE based with explicit [40, 41], semi-implicit [24] or fixed point [44] schemes. Other are based on the minimization of a discretized energy. Those include subgradient descents [28], Newton-like methods [25], second order cone programming [21], interior point methods [19], or graph based approaches [15, 10]. Recently, some authors tried to use primal-dual or dual-only approaches [12, 22, 9].

In this work, we propose new convergent schemes to solve (1.1) and (1.2). They are all based on first order explicit schemes proposed recently by Y. Nesterov [32, 34]. These schemes are given with explicit convergence rates (which is seldom seen in the literature), are optimal with respect to a certain class of convex problems, require little memory and are easy to parallelize and implement [1]. We compare their efficiency with some other classical first order schemes. We show their theoretical and practical superiority.

Depending on the regularity of F, we propose two different approaches motivated by the maximization of the theoretical rates of convergence. For general convex F, we follow the approach of Y. Nesterov in [34] and use a smooth approximation of the total variation. Doing so, getting a solution of precision ϵ requires $O(\frac{1}{\epsilon})$ iterations while most first order schemes require $O(\frac{1}{\epsilon^2})$ iterations. For strongly convex F (typically l^2-norms), we show that the resolution of a dual problem with a Nesterov's scheme leads to algorithms demanding $O(\frac{1}{\sqrt{\epsilon}})$ iterations to get an ϵ-solution.

The outline of the paper is as follows :

- In section 2, we settle the main notations and definitions. Then we show that many image processing problems such as restoration or decomposition can be expressed as (1.1) or (1.2).
- In section 3, we first analyse the convergence rates of some classical first order algorithms. Then we detail the proposed algorithms. Their convergence rates outperform the other classical schemes by one or two orders of magnitude depending on the regularity of F.
- In section 4, we compare our approach with some other existing methods.

2 Notations and application examples

2.1 Notations and definitions

Let us describe the notations we use throughout this paper.

To simplify the notations, we use $X = \mathbb{R}^n$, $Y = X \times X$, and $J(u) = ||\nabla u||_1$.

All the theory developed in this paper can be applied to color images using for instance color total variation [7]. Here we focus on gray-scale images.

$\bar{u}$ denotes a solution of (1.1) or (1.2). $f \in X$ will denote the given observed datum.

[1]this is an important feature for Graphic Processing Unit or Programmable Logic Device programming

For $u \in X$, $u_i \in \mathbb{R}$ denotes the i-th component of u.
For $g \in Y$, $g_i \in \mathbb{R}^2$ denotes the i-th component of g, and $g_i = (g_{i,1}, g_{i,2})$.
$< \cdot, \cdot >_X$ denotes the usual scalar product on X. For $u, v \in X$ we have :

$$< u, v >_X := \sum_{i=1}^{n} u_i v_i \tag{2.3}$$

$< \cdot, \cdot >_Y$ denotes the usual scalar product on Y. For $g, h \in Y$:

$$< g, h >_Y := \sum_{i=1}^{n} \sum_{j=1}^{2} g_{i,j} h_{i,j} \tag{2.4}$$

$| \cdot |_p$, $p \in [1, \infty[$ is the l^p norm on X :

$$|u|_p := (\sum_{i=1}^{n} |u_i|^p)^{1/p} \tag{2.5}$$

$| \cdot |_\infty$ is the l^∞-norm on X :

$$|u|_\infty = \max_{i \in \{1,2,\ldots,n\}} |u_i| \tag{2.6}$$

$|| \cdot ||_p$, for $p \in [1, \infty[$ is a norm on Y defined by :

$$||g||_p := (\sum_{i=1}^{n} |g_i|_2^p)^{1/p} \tag{2.7}$$

And :

$$||g||_\infty := \max_{i \in \{1,2,\ldots,n\}} |g_i|_2 \tag{2.8}$$

Finally $\lfloor a \rfloor$ is the integer part of $a \in \mathbb{R}$.

Definition 2.1. [Euclidean projector] Let $K \subset X$ be a convex set. The Euclidean projector on K is defined by :

$$\Pi_K(x) = arg \min_{u \in K} |u - x|_2$$

Definition 2.2. [Euclidean norm of an operator] Let B be a linear operator from X to Y. The Euclidean norm of B is defined by :

$$||B||_2 = \max_{x \in X, |x|_2 \leq 1} (||Bx||_2)$$

Definition 2.3. [Proper function] A convex function F on X is proper if and only if F is not identically equal to $+\infty$ and that it does not take the value $-\infty$ on X.

Definition 2.4. [Indicator function] Let $K \in X$ be a non empty closed convex subset of X. The indicator function of K, denoted χ_K, is defined by:

$$\chi_K(x) = \begin{cases} 0 & \text{if } x \in K \\ \infty & \text{otherwise} \end{cases} \tag{2.9}$$

Definition 2.5. [Subdifferential and subgradient] Let $J : X \to \mathbb{R}$ be a convex function. The subdifferential of J at point $u \in X$, is defined by :

$$\partial J(u) = \{\eta \in X, J(u) + < \eta, (x-u) >_X \leq J(x), \forall x \in X\} \tag{2.10}$$

$\eta \in \partial J(u)$ is called a subgradient.

The subdifferential can be thought of as the set of hyperplanes passing through point u which are less than the function J. On points where J is differentiable, the subdifferential reduces to a singleton : the classical gradient.

Definition 2.6. [L-Lipschitz differentiable function]
A function F defined on K is said to be L-Lipschitz differentiable if it is differentiable on K and that $|\nabla F(u_1) - \nabla F(u_2)|_2 \leq L|u_1 - u_2|_2$ for any $(u_1, u_2) \in K^2$.

Definition 2.7. [Strongly convex differentiable function]
A differentiable function F defined on a convex set $K \in X$ is said to be strongly convex if there exists $\sigma > 0$ such that :

$$< \nabla F(u) - \nabla F(v), u - v >_X \geq \frac{\sigma}{2}|u - v|_2^2 \tag{2.11}$$

for any $(u, v) \in K^2$. σ is called the convexity parameter of F. Note that property (2.11) implies that $|\nabla F(u) - \nabla F(v)|_2 \geq \frac{\sigma}{2}|u - v|_2$.

Definition 2.8. [Legendre-Fenchel Conjugate]
Let G be a convex proper application from X to $\mathbb{R} \cup \{\infty\}$. The conjugate function of G is defined by :

$$G^*(y) = \sup_{x \in X}(< x, y >_X - G(x)) \tag{2.12}$$

G^* is a convex proper function. Moreover, we have : $G^{**} = G$.

Definition 2.9. [ϵ-solutions]

An ϵ-solution of problem (1.1), is an element u_ϵ of $\{u, F(u) \leq \alpha\}$ satisfying $||Bu_\epsilon||_1 - ||B\bar{u}||_1 \leq \epsilon$, where $\bar{u}$ is a solution of (1.1).

An ϵ-solution of problem (1.2), is an element u_ϵ of X satisfying :

$$||Bu_\epsilon||_1 + F(u_\epsilon) - (||B\bar{u}||_1 + F(\bar{u})) \leq \epsilon$$

where $\bar{u}$ is a solution of (1.2).

2.2 Presentation of some applications

Many image processing models use the total variation $J(u) = ||\nabla u||_1$ as a prior on the images. This quantity somehow measures the oscillations of an image. It was introduced by Rudin, Osher and Fatemi (ROF) in [41] as a regularizing criterion for image denoising. Its main interest lies in the fact that it regularizes the images without blurring the edges. Nowadays it is appreciated for its ability to model the piecewise smooth or constant parts of an image. Let us show that the following formalism :

$$\inf_{u \in X, |\lambda(Au-f)|_p \leq \alpha} (J(u)) \tag{2.13}$$

describes lots of image processing problems.

2.2.1 $A = Id$, $p \in \{1, 2, \infty\}$ - Denoising or decomposition

Many image degradation models write : $f = u + b$. u is the original image, b is a white additive noise and f is the degraded observation. Suppose that we have a probability $P(u)$ over the space of images that is proportional to $exp(-J(u))$ [2]. Then it can be shown using the Bayes rule that the "best" way to retrieve u from f is to solve the following problem :

$$\inf_{u \in X, |u-f|_p \leq \alpha} (J(u)) \tag{2.14}$$

with $p = 1$ for impulse noise [2, 35, 11, 15], $p = 2$ for Gaussian noise [41], $p = \infty$ for uniform noise [45], and α a parameter depending on the variance of the noise. The noise might have a different variance on different parts of the image. In this case, we can solve the problem :

$$\inf_{u \in X, |\lambda(u-f)|_p \leq \alpha} (J(u)) \tag{2.15}$$

where $\lambda \in [0, \infty]^n$ is a parameter that will allow to treat differently the different regions of the image. On pixels where $\lambda_i = \infty$ the model will impose $\bar{u}_i = f_i$. On pixels where $\lambda_i = 0$, the value of $\bar{u}_i$ will only depend on the prior J. This idea was proposed in [40, 6].

Let us finally note that recently, the $BV - l^1$ model was shown to be an efficient model for the decomposition of an image into a cartoon and a texture [46].

[2] This is possible if we suppose that images have a bounded amplitude.

2.2.2 A =local cosine transform, wavelet transform, $p \in \{1, \infty\}$ Quantization, thresholding, and white noises denoising

Below, we list three different applications of this formalism:

- A classical way to compress a signal is to :

  1. Transform it with some linear, bijective application.
  2. Quantize the obtained coefficients to reduce the entropy.
  3. Use a lossless compression algorithm on the quantized coefficients.

  In image compression the first used transform was the local cosine transform in *jpeg*. The new standard is *jpeg*2000 which uses a wavelet transform. This kind of compression introduces artefacts like oscillations near the edges. Let u be an original image, and f a compressed image. The degradation operator Ψ can be written :

$$\Psi(u) = A^{-1}(Q(Au)) = f \tag{2.16}$$

  where Q is a uniform or non uniform quantizer and A is a linear transform. A natural way to recover u, is to look for the image of minimal total variation in the convex set $\Psi^{-1}(f)$ [3, 17]. This amounts to solving :

$$\inf_{u \in X, \forall i \in [1..n], \quad |(A(u-f))_i| \leq \frac{\alpha_i}{2}} (J(u))$$

  where α_i stands for the quantization steps. This problem can easily be redefined as :

$$\inf_{u \in X, |\lambda A(u-f)|_\infty \leq 1} (J(u)) \tag{2.17}$$

  with $\lambda \in [0, \infty]^n$.

- Wavelet thresholding is widely used to denoise signals. Such operations show good performances, but introduce oscillatory artefacts when using non redundant wavelet transforms. Solving a problem similar to (2.17) can be shown to reduce those artefacts.

- Recently a model similar to (2.17), with an l^1-norm instead of the l^∞-norm was proposed for image denoising [16].

2.2.3 $A =$ convolution, $p = 2$ - Image deconvolution

One of the fundamental problems of image processing is the deblurring. A common way to model image degradation is : $f = h \star u + b$. u is a given original image, b is a white Gaussian noise and h is a convolution kernel representing the degradation due to the optical system and sensors. To retrieve the original image, we can solve the following problem :

$$\inf_{u \in X, |h \star u - f|_2 \leq \alpha} (J(u)) \tag{2.18}$$

2.2.4 $A =$ Fourier transform - Image zooming

In view of Shannon's theorem, one could think that the best way to zoom an image is to use a zero-padding technique. Unluckily, this introduces oscillations near the edges. A simple way to avoid them is to solve the following problem :

$$\inf_{u \in X, |\lambda(Au - f)|_\infty \leq \alpha} (J(u)) \tag{2.19}$$

with f the zero-padded Fourier coefficients of the reduced image, $\lambda_i = \infty$ where f_i is known, and $\lambda_i = 0$ otherwise. This problem is a particular instance of a more general class of zooming techniques proposed in [29]. Let us now look at the numerical algorithms to solve (1.1) and (1.2).

3 The proposed algorithms

Problem (1.1) covers many useful applications but it is difficult to solve and many currently used algorithms are slow. This clearly limits the industrial interest for such models.

In this section, we first show that some commonly used approaches require $O(\frac{1}{\epsilon^2})$ iterations to provide an ϵ-solution. With such a rate getting a 10^{-3}-solution requires of order 10^6 iterations. Then we show how to apply the ideas of Y. Nesterov [34] to solve (1.1). This leads to an $O(\frac{1}{\epsilon})$ algorithm. We thus gain one order of convergence. With this new rate getting a 10^{-3}-solution requires of order 10^3 iterations. Finally, we show that when dealing with a strongly convex function F, the resolution of a dual problem can be done with an $O(\frac{1}{\sqrt{\epsilon}})$ algorithm. Thus, we again gain one order of convergence. With this rate getting a 10^{-3}-solution requires of order 30 iterations!

3.1 Some commonly used approaches

Maybe the most straightforward algorithm to solve (1.1) for general convex function F, is the projected subgradient descent algorithm. It writes :

$$\begin{cases} u^0 \in K \\ u^{k+1} = \Pi_K(u^k - t^k \frac{\eta^k}{|\eta^k|_2}) \end{cases} \tag{3.20}$$

Here, $t_k > 0$ for any k, η^k is any element of $\partial J(u^k)$ (see (2.10)) and Π_K is the Euclidean projector on $K = \{u, F(u) \leq \alpha\}$. It was proposed recently in some image processing papers [3, 28]. This kind of scheme has two severe drawbacks. First, it is difficult to design the sequence $\{t_k\}$. Secondly, even if the sequence $\{t_k\}$ is defined optimally, it might be very slow. It is shown in [31] that any algorithm only using the sequences $J(u^k)$ and $\partial J(u^k)$ has a worst case complexity of $O(\frac{1}{\epsilon^2})$.

Another widely spread technique consists in approximating the total variation [40, 44] by :

$$\tilde{J}_\mu(u) = \sum_{i=1}^{n} \sqrt{|(\nabla u)_i|^2 + \mu^2} \tag{3.21}$$

and use a projected gradient descent with constant step to minimize it. This idea was used from the first numerical attempts to solve total variation problems [41]. The evolution equation $\frac{\partial u}{\partial t} = div(\frac{\nabla u}{|\nabla u|})$ was replaced by $\frac{\partial u}{\partial t} = div(\frac{\nabla u}{\sqrt{|\nabla u|^2+\mu^2}})$. Again we can show that to get an ϵ-solution, one needs to choose μ of order ϵ and k of order $O(\frac{1}{\epsilon^2})$.

The two strategies presented are widely used but only lead to approximate solutions. In the following we introduce faster algorithms.

3.2 Nesterov's scheme

In [34], Y. Nesterov presents an $O(\frac{1}{\sqrt{\epsilon}})$ algorithm adapted to the problem:

$$\inf_{u \in Q} E(u) \tag{3.22}$$

where E is any convex, $L - Lipschitz - differentiable$ function, and Q is any convex, closed set. For this class of problems, it can be shown that no algorithm - only using the values and gradients of E - has a better rate of convergence than $O(\frac{1}{\sqrt{\epsilon}})$ uniformly on all problems of type (3.22). Y. Nesterov's algorithm is thus optimal for this class of problems.

His main result is as follows :

Theorem 3.1. *The following algorithm :*

$$\boxed{\begin{array}{ll} 1 & \text{Set } k = -1,\ G^{-1} = 0,\ x^k \in Q,\ L \text{ Lipschitz constant of } \nabla E \\ 2 & \text{Set } k = k+1, \text{ compute } \eta^k = \nabla E(x^k) \\ 3 & \text{Set } y^k = arg\min_{y \in Q} \left(< \eta^k, y - x^k >_X + \frac{1}{2} L||y - x^k||^2\right) \\ 4 & \text{Set } G^k = G^{k-1} + \frac{k+1}{2}\eta^k \\ 5 & \text{Set } z^k = \arg\min_{z \in Q} \left(\frac{L}{\sigma} d(z) + < G^k, z >_X\right) \\ 6 & \text{Set } x^{k+1} = \frac{2}{k+3} z^k + \frac{k+1}{k+2} y^k, \text{ go back to 2 until } k = N. \end{array}} \tag{3.23}$$

ensures that :

$$0 \leq F(y^k) - F(\bar{u}) \leq \frac{4Ld(\bar{u})}{\sigma(k+1)(k+2)} \tag{3.24}$$

At step 3, $||\cdot||$ denotes any norm, at step 5, d is any convex function satisfying:

$$d(x) \geq \frac{\sigma}{2}||x - x_0||^2 \text{ for some element } x_0 \in K and \sigma \text{ is the convexity parameter of } d. \quad (3.25)$$

Using inequality (3.24), it is easily seen that getting an ϵ-solution does not require more than $\sqrt{\frac{4Ld(\bar{u})}{\epsilon}}$ iterations. This shows that (3.23) is an $O(\frac{1}{\sqrt{\epsilon}})$ algorithm.

Supposing that steps 3 and 5 are achievable, this scheme has many qualities. It is simple to implement, does not require more than 5 times the size of the image and it is theoretically optimal (see [32] for a precise definition of its optimality). Note that the classical projected gradient descent is an $O(\frac{1}{\epsilon})$ algorithm.

3.3 Solving (1.1) for the total variation and some convex functions F

Unfortunately, algorithm (3.23) cannot be used directly to solve (1.1) due to the non differentiability of the total variation. To cope with this difficulty, we use a smooth approximation of J.

How to smooth the total variation?

Following the ideas in [34], we use a uniform smooth approximation of J. First note that :

$$J(u) = \sup_{q \in Y, ||q||_\infty \leq 1} (< \nabla u, q >_Y) \quad (3.26)$$

The approximation we propose writes :

$$J_\mu(u) = \sup_{q \in Y, ||q||_\infty \leq 1} \left(< \nabla u, q >_Y - \frac{\mu}{2}||q||_2^2\right) + \frac{\mu}{2} \quad (3.27)$$

It is easily shown that $J_\mu(u) = \sum_{i=1}^n \psi_\mu(|(\nabla u)_i|)$ with :

$$\psi_\mu(x) = \begin{cases} |x| & \text{if } |x| \geq \mu \\ \frac{x^2}{2\mu} + \frac{\mu}{2} & \text{otherwise} \end{cases} \quad (3.28)$$

ψ_μ is thus the so called Huber function. J_μ seems more appropriate than $\tilde{J}_\mu$ defined in (3.21), as both approximations are $\frac{||div||_2^2}{\mu}$-Lipschitz differentiable [3], but :

$$0 \leq \tilde{J}_\mu(u) - J(u) \leq n\mu \quad (3.29)$$

[3] the Lipschitz constant determines the convergence rate of most first order schemes

while :

$$0 \leq J_\mu(u) - J(u) \leq \frac{n\mu}{2} \tag{3.30}$$

The approximation J_μ thus seems "twice" better. Let us finally note that :

$$\nabla J_\mu(u) = -div(\Psi) \quad \text{with } \Psi_i = \begin{cases} \frac{(\nabla u)_i}{|(\nabla u)_i|} & \text{if } |(\nabla u)_i| \geq \mu \\ \frac{(\nabla u)_i}{\mu} & \text{otherwise} \end{cases} \tag{3.31}$$

From now on, we concentrate on the resolution of :

$$\inf_{u \in X, F(u) \leq \alpha} J_\mu(u) \tag{3.32}$$

How to achieve steps 3 and 5?

To apply algorithm (3.23), we first have to choose a norm $||\cdot||$ and a function d. For simplicity and because we found no numerical interest in other choices, we choose the Euclidean norm $|\cdot|_2$ and set $d(x) = \frac{1}{2}|x - x_0|_2^2$, where $x_0 \in K$ is for instance the center of K. With this choice $\sigma = 1$. We have to find - preferably in closed form - the expressions of the *argmin* at steps 3 and 5.

Proposition 3.1. *With the above choices, step* 3 *reduces to* $y^k = \Pi_K(x^k - \frac{\eta^k}{L})$ *and step* 5 *reduces to* $z^k = \Pi_K(x_0 - \frac{G^k}{L})$. Π_K *stands for the Euclidean projector on* K.

Proof: Let us solve the problem $argmin_{y \in K}(f(y))$ with $f(y) =< \eta, y >_X + \frac{L}{2}|y - x|_2^2$. From first order optimality conditions, we get that the solution $\bar{y}$ of this problem satisfies $< (-\nabla f(\bar{y})), w - \bar{y} > \leq 0$ for any $w \in K$. This is equivalent to $< (x - \frac{\eta}{L}) - \bar{y}, w - \bar{y} >_X \geq 0$ for any $w \in K$ and finally, thanks to projection theorem to $\bar{y} = \Pi_K(x - \frac{\eta}{L})$.

■

How to chose μ and the number of iterations?

Convergence rate (3.24) and bound (3.30) tell us that after k iterations of a Nesterov scheme, the worst case precision is $\frac{2||div||_2^2 d(\bar{u})}{\mu k^2} + \frac{n\mu}{2}$. Using this quantity, it is easily shown that the fastest way to get an ϵ-solution is to chose $\mu = \frac{\epsilon}{n}$ and the number of iterations :

$$N = \lfloor \frac{2\sqrt{2}||div||_2}{\epsilon}\sqrt{nD(K)} \rfloor + 1 \tag{3.33}$$

where $D(K) = \max_{u \in K}(d(u))$. This shows that algorithm (3.23) has a complexity in $O(\frac{1}{\epsilon})$.

Unfortunately, in most problems, knowing that $|J(y^k) - J(\bar{u})| \leq \epsilon$ does not bring any quantitative information on more important features like $|y^k - \bar{u}|_\infty$. Thus, we cannot use directly the bound (3.24) to define the number of iterations.

Experimentally, for images rescaled in $[0, 1]$, we can check that the solution of (3.32) obtained by choosing $\mu = 0.02$ is very close perceptually to the solution of (1.1). Choosing $\mu = 0.001$ leads to solutions that are perceptually identical to the solution of (1.1).

From this remark and (3.33) the approach we suggest is thus to choose $\mu \in [0.001, 0.01]$ and there is no reason to chose $N > \frac{2\sqrt{2}||div||_2}{n\mu}\sqrt{nD(K)}$. In all cases we studied (except deconvolution) $D(K) \sim \theta n$, with $\theta \in]0, 1[$. Thus, the maximum iterations needed to get a good approximate solution is :

$$N = \frac{2\sqrt{2}||div||_2\sqrt{\theta}}{\mu} \tag{3.34}$$

This quantity does not exceed 8000 iterations for the worst case problem, and lies in $[30, 400]$ for most practical applications.

The theoretical rate of convergence leads to low iterations number and computing times.

Let us show how to apply the ideas presented for some applications.

Restoration involving linear invertible transforms : $F(u) = |\lambda(Au - f)|_p$ **with** $p \in \{1, 2, \infty\}$

One of the most interesting data term is $F(u) = |\lambda(Au - f)|_p$ where $p \in \{1, 2, \infty\}$, A is a linear *invertible* transform, $\lambda \in [0, \infty]^n$ are some parameters, and f are the some given data. To apply algorithm (3.23) to problem (3.32), we need to be able to compute projections on $\{u, |\lambda(Au - f)|_p \leq \alpha\}$. As this might be cumbersome, we use a change of variable and solve the equivalent problem :

$$\inf_{z, |\lambda z|_p \leq \alpha} (E_\mu(z)) \tag{3.35}$$

with $E_\mu(z) = J_\mu(A^{-1}(z+f))$. The solution $\bar{u}$ of (3.32) can be retrieved from the solution $\bar{z}$ of (3.35) by :

$$\bar{u} = A^{-1}(\bar{z} + f) \tag{3.36}$$

It is easy to show that E_μ is L-Lipschitz differentiable with $L = \frac{||div||_2^2||A^{-1}||_2^2}{\mu}$.

For all invertible transforms the final algorithm to solve (3.35) writes :

$$\boxed{\begin{array}{ll} 1 & \text{Chose } N \text{ and } \mu \text{ depending on the precision required} \\ & \text{Set } k=0,\ G^{-1}=0,\ L=\frac{||div||_2^2||A^{-1}||_2^2}{\mu},\ x^k=0 \\ 2 & \text{Compute } \eta^k = A^{-*}\nabla J_\mu(A^{-1}(x^k+f)) \\ 3 & \text{Set } y^k = \Pi_K(x^k - \frac{\eta_k}{L}) \\ 4 & \text{Set } G^k = G^{k-1} + \frac{k+1}{2}\eta^k \\ 5 & \text{Set } z^k = \Pi_K(-\frac{G^k}{L}) \\ 6 & \text{Set } x^{k+1} = \frac{2}{k+3}z^k + \frac{k+1}{k+2}y^k \\ 7 & \text{Set } k=k+1, \text{ go back to 2 until } k=N. \\ 8 & \text{Set } \bar{u} = A^{-1}(y^{k-1}+f) \end{array}} \tag{3.37}$$

At steps 3 and 5, $K = \{u \in X, |\lambda u|_p \leq \alpha\}$. We refer the reader to the appendix for the expressions of the projections on weighted l^p-balls.

Figure 1 shows the result of a compression noise restoration using this technique for $\mu = 0.01$ (60 iterations until convergence) and $\mu = 0.001$ (120 iterations until convergence). The price per iteration is about 2 wavelet transforms. We do not give our computational times as we used a slow Matlab implementation of Daubechies 9-7 wavelet transform. We refer the reader to section (4) for comparisons with other algorithms. Let us note that to our knowledge, few or no precise schemes exist in the literature to solve this problem. We note that large oscillations are removed, while thin details are preserved. The main default of this model is that the contrast of the details decrease.

Deconvolution : $F(u) = |h \star u - f|_2$

In this paragraph we present a way to do deconvolution using a Nesterov scheme. This problem is particularly difficult and cannot be solved by the previous algorithm, because the convolution matrices are generally non-invertible or ill-conditioned. In the following, we show a way to handle this problem.

We wish to solve the following problem :

$$\inf_{u \in X, |h \star u - f|_2 \leq \alpha} (J_\mu(u)) \tag{3.38}$$

Let A denote the Fourier Transform. As it is an isometry from X to X, this problem is equivalent to :

$$\begin{aligned} & \inf_{u \in X, |A(h \star u - f)|_2 \leq \alpha} (J_\mu(u)) & (3.39) \\ \Leftrightarrow\ & \inf_{u \in X, |AhAu - Af|_2 \leq \alpha} (J_\mu(u)) & (3.40) \\ \Leftrightarrow\ & \inf_{z \in \mathbb{C}^n, |Ahz - Af|_2 \leq \alpha} \left(J_\mu(A^{-1}z)\right) & (3.41) \end{aligned}$$

In this formulation is $z \to J_\mu(A^{-1}z)$ is L-Lipschitz-differentiable with $L = \frac{||div||_2^2}{\mu}$. The interests of this formulation are that we have a very fast algorithm to do projections on

Figure 1: Example of image decompression - TL : Original image - TR : Compressed image using Daubechies 9-7 wavelet transform (the quantization coefficients are chosen by hand to outline the model features) - BL : Solution of (2.17) using $\mu = 0.01$ - BR : Solution of (2.17) using $\mu = 0.001$

$K = \{u \in X, |Ahz - Af|_2 \leq \alpha\}$ (see the appendix section (6.2)), and that the Lipschitz constant of the gradient of $z \to J_\mu(A^{-1}z)$ does not blow up. Note that the set $K = \{u \in X, |Ahz - Af|_2 \leq \alpha\}$ might be unbounded if the Fourier transform of h contains zeros. Thus we lose the convergence rate unless we estimate an upper bound on $d(\bar{u})$. Practically, the Nesterov scheme remains efficient.

The cost per iteration is around 2 fast Fourier transforms and 2 projections on ellipsoids. For a 256×256 image, the cost per iteration is 0.2 seconds (we used the $fft2$ function of Matlab and implemented a C code with Matlab mex compiler for the projections).

Image texture + cartoon decomposition : $F(u) = \lambda|u - f|_G$

The first application of total variation in image processing was proposed by Rudin-Osher-Fatemi in [41]. It consisted in choosing $F(u) = |u - f|_2$. In [30], Y. Meyer studied theoretically this model, and figured out its limitation to discriminate well a cartoon in a noise or a texture. He observed that this limitation could be overpassed using a different data term than the rather uninformative L^2-distance to the data. To simplify the presentation, we present the model in the discrete setting and refer the interested reader to [30, 5] for more details.

Y. Meyer defined a norm :

$$||v||_G = \inf_{g \in Y}\{||g||_\infty, div(g) = v\} \tag{3.42}$$

and proposed to decompose an image f into a cartoon u and a texture v using the following model :

$$\inf_{(u,v) \in X^2, f = u+v}\{J(u) + \lambda||v||_G\} \tag{3.43}$$

The G-norm of an oscillating function remains small and it blows up for characteristic function. That is why this model should permit to better extract oscillating patterns of the images.

Y. Meyer did not propose any numerical method to solve his model. The first authors who tried to compute a solution were L. Vese and S. Osher in [38]. Later, other authors tackled this problem. Let us cite the works of J.F. Aujol et. al. in [4] and of D. Goldfarb et. al. in [21]. The former is based on a first order scheme which solves a differentiable approximation of Meyer's model, while the latter solves it exactly with second order cone programming methods. In the following, we propose a new efficient scheme.

Y. Meyer's discretized problem writes :

$$\inf_{u \in X}\{J(u) + \lambda \inf_{g \in Y, div(g) = f-u}\{||g||_\infty)\}\} \tag{3.44}$$

We have the following result :

Proposition 3.2.
Problem (3.44) can be reformulated as follows :

$$\inf_{g \in Y, ||g||_\infty \leq \alpha} \{J(f - div(g))\} \tag{3.45}$$

Proof: The main idea is that we can use the change of variable :

$$u = f - div(g) \tag{3.46}$$

to get an optimization problem that depends only of one variable g. The operator div is surjective from Y to $\tilde{X} = X - \{(\gamma, \gamma, ..., \gamma), \gamma \in \mathbb{R}\}$, so that :

$$\inf_{u \in X} \{J(u) + \lambda \inf_{g \in Y, div(g) = f - u} \{||g||_\infty\}\} = \inf_{g \in Y} \{J(f - div(g)) + \lambda ||g||_\infty\} \tag{3.47}$$

Turning the Lagrange multiplier λ into a constraint, we get the following minimization problem :

$$\inf_{g \in Y, ||g||_\infty \leq \alpha} \{J(f - div(g))\} \tag{3.48}$$

■

Instead of solving problem (3.45), we solve :

$$\inf_{g \in Y, ||g||_\infty \leq \alpha} \{J_\mu(f - div(g))\} \tag{3.49}$$

and get an $O(\frac{1}{\epsilon})$ algorithm. Note that the solution of (3.49) is unique while that of Meyer's model is not.

Also note that if we replace the l^∞-norm by an l^2-norm in (3.45), we get the model of Osher-Solé-Vese [37]. In Figure (4), we also show the result of (3.45) with an l^1-norm instead of the l^∞-norm. We do not provide any theoretical justification to this model, we present it to alleviate the curiosity of the reader. Formula (3.45) allows to easily constrain the properties of the g field. This might also be interesting for spatially varying processing.

In all experiments we took $\mu = 0.001$ to smooth the total variation. After 200 iterations, very little perceptual modifications are observed in all experiments, while a projected gradient descent requires around 1000 iterations to get the same result. Let us finally precise that all the texture components have the same l^2-norm.

In Figure (3), we observe that Meyer's model does not allow to retrieve correctly the oscillating patterns of the clothes of Barbara. It can be shown that the amplitude of the

Figure 2: Image to be decomposed

texture (v component) is bounded by a parameter depending linearly on α in (3.45). That might explain the deceiving result. On the given example, Osher-Solé-Vese's model gives more satisfying results. This was already observed in [46].

The $BV - l^1$ model correctly separates the oscillating patterns and the geometry. The same observation holds when minimizing the l^1-norm of the g field in (3.45). We remark that both decompositions are very similar, except that the cartoon component of the $BV - l^1$ model is slightly less blurred than that of the new model. We think that one part of the explanation is that the numerical scheme for the new model is slightly more diffusive as it is based on second order finite differences.

3.4 Solving (1.2) for l^2 fidelity term

Having the previous section in mind, a straightforward approach to solve (1.2) is to smooth the total variation, if F is non-smooth, it can be smoothed too, and then one just needs to use a fast scheme like (3.23) adapted to the unconstrained minimization of Lipschitz differentiable functions [33]. This method should be functional, but in the case of strongly convex F - which notably corresponds to l^2 data fidelity term - one can do much better. We present an $O(\frac{1}{\sqrt{\epsilon}})$ algorithm rather than the previous $O(\frac{1}{\epsilon})$ algorithm. Thus we gain two orders of convergence compared to classical approaches!

Instead of directly attacking (1.2) we can solve its *dual problem* for which no smoothing is needed. The key idea is that for strongly convex F, F^* is Lipschitz differentiable.

We first recall some facts of convex analysis (see [18], for a complete reference).

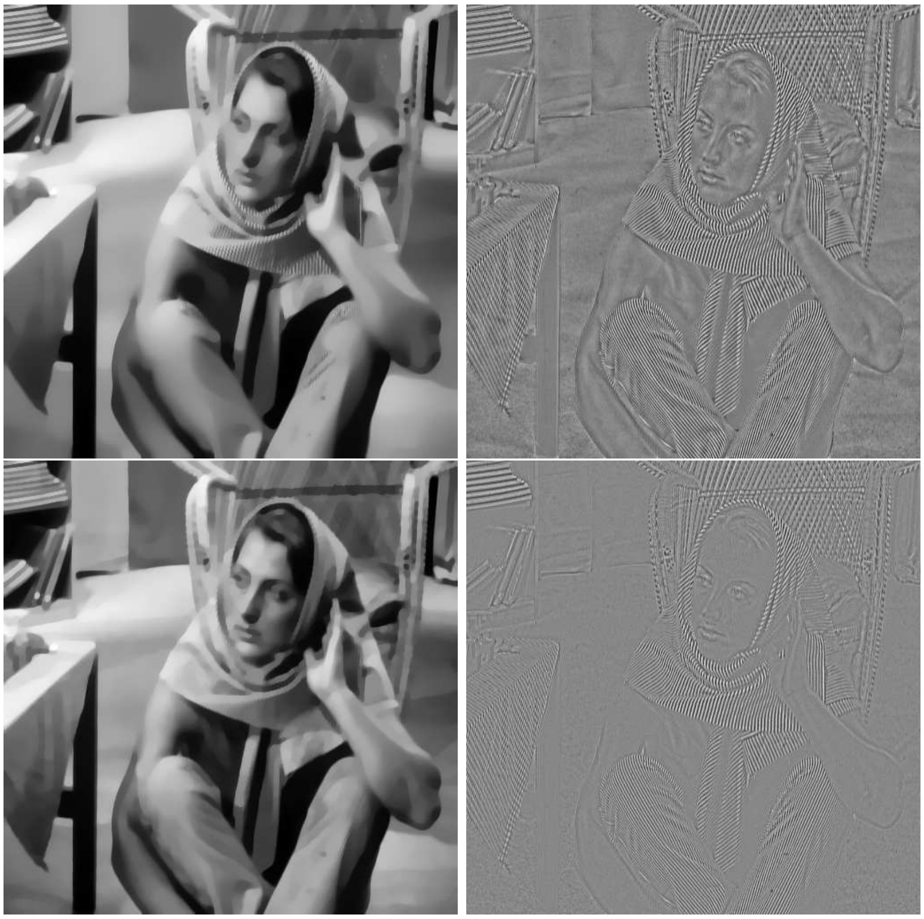

Figure 3: Cartoon + texture decompositions. Top : Meyer's model. Bottom : Osher-Solé-Vese's model

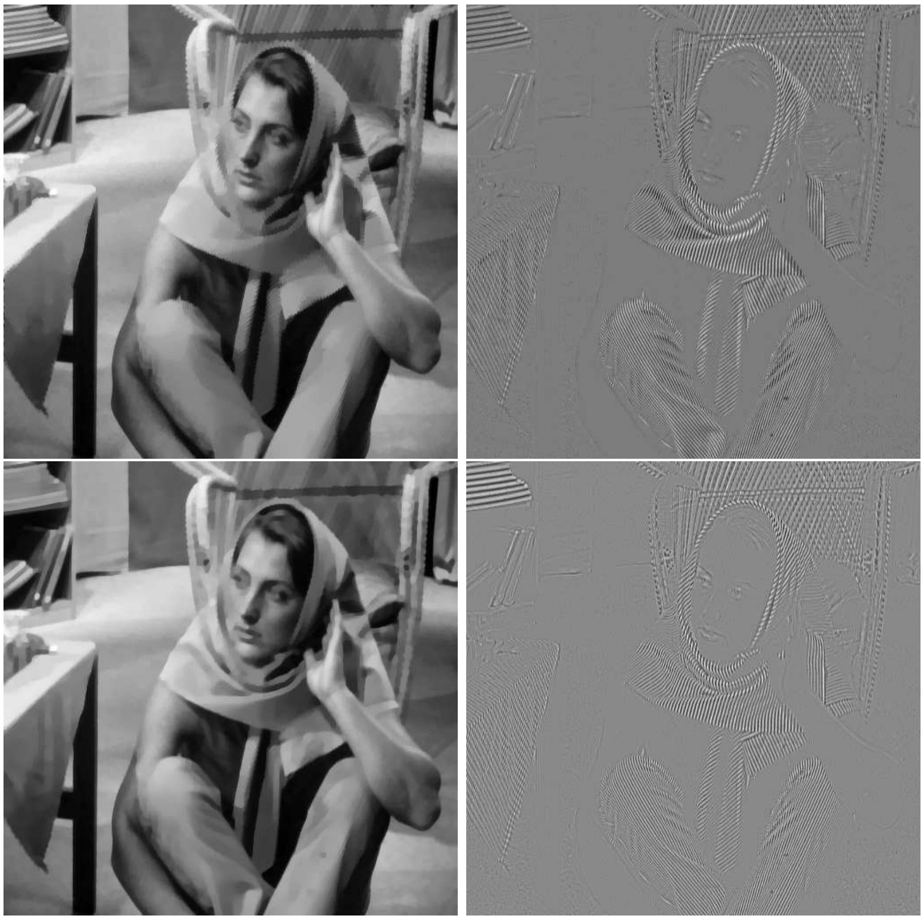

Figure 4: Cartoon + texture decompositions. Top : $BV - l^1$ model. Bottom : result of minimizing the l^1-norm of the g field in (3.45)

Let $F : X \to \mathbb{R}$ and $G : Y \to \mathbb{R}$ be two convex proper functions. Let $\mathcal{P}$, be the primal problem :

$$\inf_{u \in X} G(\nabla u) + F(u) \tag{3.50}$$

The dual problem $\mathcal{P}^*$ is then defined by :

$$\inf_{q \in Y} G^*(-q) + F^*(-div(q)) \tag{3.51}$$

Let $\bar{u}$ and $\bar{q}$ be the solutions of $\mathcal{P}$ and $\mathcal{P}^*$ respectively. Those solutions are related through the extremality relations :

$$F(\bar{u}) + F^*(-div(\bar{q})) = < -div(\bar{q}), \bar{u} >_X \tag{3.52}$$

$$G(\nabla \bar{u}) + G^*(-\bar{q}) = < -\bar{q}, \nabla \bar{u} >_Y \tag{3.53}$$

3.4.1 Application to our problem

To apply the previous theory to our problem, we take :

$$G(q) = ||q||_1 \tag{3.54}$$

We wish to solve :

$$\inf_{u \in X} (G(\nabla u) + F(u)) \tag{3.55}$$

and suppose that F is differentiable, strongly convex, with convexity parameter σ.

Theorem 3.2. *The dual problem of (3.55) is defined by :*

$$\inf_{q \in K} F^*(-div(q)) \tag{3.56}$$

with $K = \{q \in Y, ||q||_\infty \leq 1\}$.

The application $H : q \to F^*(-div(q))$, *is L-Lipschitz differentiable, with* $L = \frac{2||div||_2^2}{\sigma}$. *Problem (3.56) can thus be solved with a Nesterov scheme (no smoothing is needed!).*

$\bar{u}$ *can be retrieved from the solution* $\bar{q}$ *of (3.56) using :*

$$\bar{u} = \nabla F^*(-div(\bar{q})) \tag{3.57}$$

moreover :

$$\nabla \bar{u} = \bar{q} |\nabla \bar{u}| \tag{3.58}$$

This methods thus amounts to evolving the orientation of the level lines of u instead of u itself.

Proof :

1. Let us compute G^* :

$$
\begin{aligned}
G^*(q) &:= \sup_{r \in Y} < q, r >_Y - ||r||_1 && (3.59)\\
&= \sup_{t>0} \sup_{||r||_1 = t} < q, r >_Y - t && (3.60)\\
&= \sup_{t>0} t||q||_\infty - t && (3.61)\\
&= \chi_K(q) && (3.62)
\end{aligned}
$$

with $K = \{q \in Y, ||q||_\infty \leq 1\}$.

2. Let us show F^* is $\frac{2}{\sigma}$-Lipschitz differentiable.

$$F^*(u) = \sup_{v \in X} < u, v >_X - F(v) \quad (3.63)$$

First, note that F^* is convex (see section 2). As F is strictly convex, the solution of problem (3.63) exists and is unique. Let $v(u)$ denote the *argmax* in (3.63). From uniqueness of the solution of (3.63), we get that F^* is differentiable and its derivative is $v(u)$. From the optimality conditions we get that $u - \nabla F(v(u)) = 0$. Thus for any $(u_1, u_2) \in X^2$:

$$\nabla F(v(u_1)) - \nabla F(v(u_2)) = u_1 - u_2 \quad (3.64)$$

and :

$$
\begin{aligned}
|u_1 - u_2|_2 &= |\nabla F(v(u_1)) - \nabla F(v(u_2))|_2 && (3.65)\\
&\geq \frac{\sigma}{2}|v(u_1) - v(u_2)|_2 && (3.66)\\
&\geq \frac{\sigma}{2}|\nabla F^*(u_1) - \nabla F^*(u_2)|_2 && (3.67)
\end{aligned}
$$

This shows that F^* is $\frac{2}{\sigma}$-Lipschitz differentiable.

3. Let us show (3.57). The first extremality relation gives $F(\bar{u}) =< -div(\bar{q}), \bar{u} >_X - F^*(-div(\bar{q}))$. We also recall that $F^{**}(u) = F(u)$. So that $F(\bar{u}) = \sup_{v \in X} < \bar{u}, v >_X - F^*(v)$.

Those two equations imply that $-div(\bar{q})$ cancels the derivative of $v \rightarrow < \bar{u}, v >_X - F^*(v)$. This ends the proof.

4. Finally let us show equation (3.58). It is done using the second extremality relation :

$$\begin{aligned} G(\nabla \bar{u}) &= G^{**}(\nabla \bar{u}) && (3.68)\\ &= \sup_{q\in Y} < \nabla \bar{u}, q >_Y - G^*(q) && (3.69)\\ &= < -\bar{q}, \nabla \bar{u} >_Y - G^*(-\bar{q}) && (3.70)\end{aligned}$$

Thus $-\bar{q}$ solves problem (3.69). This yields the existence of multipliers μ_i such that :

$$(\nabla \bar{u})_i - \mu_i \bar{q}_i = 0 \qquad (3.71)$$

with $\mu_i = 0$ if $|\bar{q}_i|_2 < 1$, or $\mu_i > 0$ if $|\bar{q}_i|_2 = 1$. In both cases we get $\mu_i = |(\nabla \bar{u})_i|_2$.

■

Application example : $F(u) = |\lambda(Au - f)|_2^2$

An important class of strongly convex functions writes : $F(u) = |\lambda(Au - f)|_2^2$ with A a bijective linear application, and $\lambda \in]0, \infty]^n$. Let $\lambda_- = \min_i \lambda_i$ and $\lambda_-(A)$ denote the smallest eigenvalue of A.

Proposition 3.3. *If A is an invertible transform, and λ belongs to $]0, \infty]^n$, then F^* is L-Lipschitz differentiable, with $L \leq \frac{1}{2\lambda_-^2 \lambda_-(A)^2}$. Moreover :*

$$F^*(v) =< A^{-1} f, v > + \frac{1}{4} |\frac{A^{-*} v}{\lambda}|_2^2 \qquad (3.72)$$

Proof : F is obviously differentiable, with derivative $\nabla F(u) = 2A^* \lambda^2 (Au - f)$. Thus :

$$|\nabla F(u) - \nabla F(v)|_2 = 2|A^* \lambda^2 A(u - v)|_2 \geq 2\lambda_-^2 \lambda_-^2(A)|u - v|_2$$

F is thus strongly convex with convexity parameter $4\lambda_-^2 \lambda_-(A)^2$. Then :

$$\begin{aligned} F^*(v) &= \sup_{u\in X} \left(< u, v >_X - |\lambda(Au - f)|_2^2\right) && (3.73)\\ &= \sup_{w\in X} \left(< \frac{A^{-1}(w + \lambda f)}{\lambda}, v > - |w|_2^2\right) && (3.74)\\ &= < A^{-1} f, v > + \sup_{r\geq 0} \left(r|\frac{A^{-*} v}{\lambda}|_2 - r^2\right) && (3.75)\\ & && (3.76)\end{aligned}$$

And we get the result by canceling the derivative of $r \to r|\frac{A^{-*}v}{\lambda}|_2 - r^2$.

■

To solve problem :

$$\inf_{u\in X} ||\nabla u||_1 + |\lambda(Au - f)|_2^2 \tag{3.77}$$

We set $K = \{q \in Y, ||q||_\infty \leq 1\}$ and the algorithm we propose writes :

$$\boxed{\begin{array}{ll} 0 & \text{Choose } N. \\ 1 & \text{Set } k=0,\ G^{-1}=0,\ L=\frac{||div||_2^2}{2\lambda_-^2\lambda_-(A)^2},\ x^k=0 \\ 2 & \text{Compute } \eta^k = \nabla(A^{-1}f) - \frac{1}{2}\nabla(A^{-1}(\frac{A^{-*}(div(x^k))}{\lambda^2})) \\ 3 & \text{Set } y^k = \Pi_K(x^k - \frac{\eta_k}{L}) \\ 4 & \text{Set } G^k = G^{k-1} + \frac{k+1}{2}\eta^k \\ 5 & \text{Set } z^k = \Pi_K(-\frac{G^k}{L}) \\ 6 & \text{Set } x^{k+1} = \frac{2}{k+3}z^k + \frac{k+1}{k+2}y^k \\ 7 & \text{Set } k=k+1, \text{ go back to 2 until } k=N. \\ 8 & \text{Set } u^N = A^{-1}f - \frac{1}{2}A^{-1}(\frac{A^{-*}div(y^k)}{\lambda^2}) \end{array}} \tag{3.78}$$

Using (3.24), we easily get that at iteration N we have :

$$0 \leq F^*(-div(y^n)) - F^*(-div(\bar{q})) \leq \frac{n||div||_2^2}{\lambda_-^2\lambda_-(A)^2}\frac{1}{N^2} \tag{3.79}$$

so that to get an ϵ-solution on the dual problem, we do not need to choose N greater than $\frac{||div||_2\sqrt{n}}{\lambda_-\lambda_-(A)}\frac{1}{\sqrt{\epsilon}}$.

4 Numerical results and discussion

We cannot do an exhaustive comparison of all numerical methods that solve total variation problems. The bibliography about this problem contains more than 50 items. Most are time consuming to implement and their efficiency heavily depends on some choices like preconditionners. We thus restrict our experimental numerical comparisons to two easy, widely used first order methods. Namely :

- the [projected]-subgradient descent with optimal step.
- the [projected]-gradient descent after smoothing of the total variation.

4.1 Experimental results

Is there a numerical difference between the smooth approximations of total variation?

In equation (3.28), we presented a way to approximate the total variation, and claimed that it should be better than the classical approximation (3.21). In all numerical experiments we perform, the minimization of (3.28) leads to solutions that have a lower total variation than (3.21), but the visual aspect of the solutions are the same. As the complexity of computing ∇J_μ or $\nabla \tilde{J}_\mu$ is the same, we think that using J_μ definitely is a better choice if one aims at approximating the total variation using first order methods.

Some comparisons for the constrained Rudin-Osher-Fatemi problem.

The problem we choose for numerical comparisons is the Rudin-Osher-Fatemi model. It consists in solving :

$$\inf_{u \in X, |u-f|_2 \leq \alpha} (J(u)) \tag{4.80}$$

We note $\bar{J}$ the minimum in (4.80). Let us described the methods we implement for comparisons.

- The first is a *projected gradient descent with optimal constant step* :

$$\left\{ \begin{array}{l} u^0 = f \\ u^{k+1} = \Pi_K(u^k - t\nabla J_\mu(u^k)) \end{array} \right. \tag{4.81}$$

  The optimal step can be shown to be $t = \frac{2\mu}{||div||_2^2}$ [39]. We set $K = \{u \in X, |u-f|_2^2 \leq \alpha^2\}$.

- The second is a *projected subgradient descent with optimal step* :

$$\left\{ \begin{array}{l} u^0 = f \\ u^{k+1} = \Pi_K(u^k - t_k \frac{\eta^k}{||\eta^k||_2}) \end{array} \right. \tag{4.82}$$

  η^k must belong to $\partial J(u^k)$ for convergence. We choose :

$$\eta^k = -div(\Psi) \text{ with } \Psi_i = \left\{ \begin{array}{cc} \frac{(\nabla u^k)_i}{|(\nabla u^k)_i|_2} & \text{if } |(\nabla u^k)_i|_2 > 0 \\ 0 & \text{otherwise} \end{array} \right. \tag{4.83}$$

  and $t_k = \frac{J(u^k) - \bar{J}}{|\eta^k|_2}$. This step is optimal in the sense that it leads to an $O(\frac{1}{\epsilon^2})$ algorithm. As $\bar{J}$ is unknown, some authors try to evaluate it iteratively [23, 28]. To find it, we just let a program (Nesterov) run until convergence, and get the optimal value $\bar{J}$. Clearly this method is not usable in practice but serves as a reference.

- The third is the presented *Nesterov approach* (3.37).

We used the 256×256 Lena image rescaled in $[0, 1]$. We set $\alpha = 20$ in (4.80). This corresponds to the images in figure (5). The curves in figure (6) compare the decrease of the total variation w.r.t the number of iterations for the different methods.

Figure 5: Left : original image - Right : solution of (4.80) choosing $\alpha = 20$

We notice that in all cases, the Nesterov's scheme achieves much better than the projected gradient descent. The difference gets larger as the Lipschitz constant of the cost function increases. For any value of μ the Nesterov's approach competes with the projected subgradient descent.

Some comparisons for Lagrangian Rudin-Osher-Fatemi problem.

In this part, we solve the following problem :

$$\inf_{u \in X} \left(J(u) + \lambda^2 |u - f|_2^2 \right) \tag{4.84}$$

Our goal is to evaluate the efficiency of the dual approach presented in (3.78). We compare it with :

- the subgradient descent with optimal step.
- the Nesterov approach evaluated previously with $\mu = 0.005$. This approach solved the primal constrained problem. We thus have to find parameters α and λ such that problems (4.80) and (4.84) are equivalent. In order to do so, we let the Nesterov method applied to the dual problem of (4.84) run until convergence. This gives us a solution $\bar{u}_\lambda$. Then we compute $\alpha = |\bar{u}_\lambda - f|_2$.

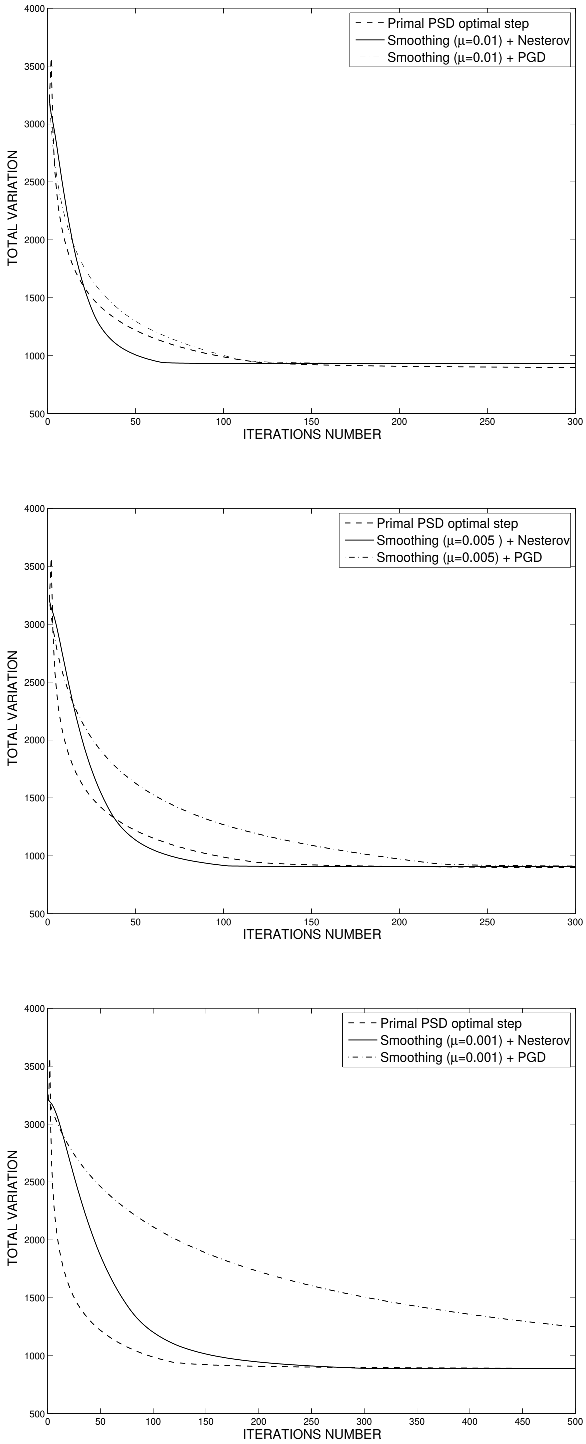

Figure 6: Total variation VS number of iterations for $\mu \in \{1e-2, 5e-3, 1e-3\}$. PSD stands for Projected subgradient descent. PGD stands for projected gradient descent.

- the projected gradient descent applied to the dual problem (3.56). This approach is slightly faster than A. Chambolle's algorithm [9] (see for instance [10]).

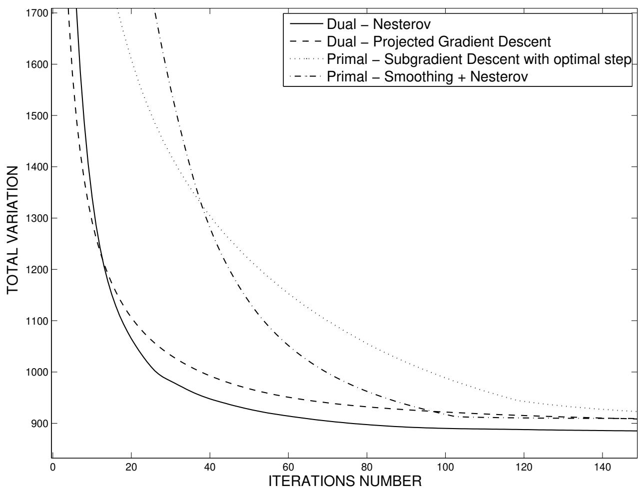

Figure 7: Total variation w.r.t to the iterations number.

The dual problem solved with a Nesterov scheme clearly outperforms all tested approaches. After 500 iterations (6 seconds) it gives a precision that can be attained by no other scheme in less than 2 minutes.

Another interesting remark is that the smoothing technique competes with the projected gradient descent applied to the dual problem and that they have the same theoretical $O(\frac{1}{\epsilon})$ rate of convergence. Let us precise that the computational effort per iteration of the Nesterov technique is between one and twice that of the projected gradient descent.

4.2 Discussion

Stability of the Nesterov scheme

People who have tried the Nesterov technique, report experimental failure (see for instance [47]). Our conclusion is opposite. In all tested cases, the Nesterov algorithm is faster than the projected gradient descent. We hope this will incite people to use it.

Comparisons with other methods

Second order methods are commonly used to solve problem (1.2) and it seems they represent the closest rivals of our approach. Many papers suggest the use of 'half-quadratic minimization' [20] which was shown recently to be equivalent to quasi-Newton techniques [36]. Those methods are proved to converge linearly [1]. Such a rate is better asymptotically than our polynomial energy decays. This results in convergence after fewer iterations. The counterpart clearly is the need to solve a huge linear system at each iteration. The efficiency of this method strongly depends on the conditioning number of the system and the choice of preconditioners. It is thus difficult to compare both approaches. We think that our method should be advantageous in the case of low resources programming environments. More computational comparisons should be led to give a definitive answer in the case of PC programming.

Second order cone programming was proposed recently [46] and leads to very precise solutions, but the computing times seem to be higher than those of the simple gradient descents for moderate precision.

A promising approach based on graph-cuts was proposed recently [15, 10]. The authors solve (1.2) for $A = Id$ and $p \in \{1, 2, \infty\}$. They show that they get exact solutions (up to a quantization parameter) in a finite number of iterations. The computing times obtained seem to be slightly higher than those of the dual algorithm [10] for Rudin-Osher-Fatemi problem. They should be better than our approach for the $BV - L^1$ problem. We think that the presented approach allows to solve a larger class of problems and is simpler to implement.

5 Conclusion

We presented efficient first order algorithms to minimize the total variation under many smooth or non-smooth convex sets. Those schemes are simple to implement and present low computing times. They are based on a recent advance [34] in convex optimization. Their efficiency is comparable or better than state of the art methods.

In this paper we focused on total variation problems. More numerical experiments should be led for the minimization of l^1-norm of linear transforms.

Acknowledgement: The first author would like to thank Alexis Baudour for useful mathematical discussions.

A Appendix

A.1 Discretization of differential operators

In this section, to simplify the notations, we denote $u(i, j)$ the value of u on pixel (i, j). n_x and n_y will represent the number of pixels in the horizontal and vertical directions respectively.

To discretize the gradient we used in all experiments the following classical first order scheme borrowed from [9]. For $u \in X$:

$$(\nabla u)(i,j) = ((\partial_1 u)(i,j), (\partial_2 u)(i,j)) \tag{1.85}$$

∇u is an element of Y.

$$(\partial_1 u)(i,j) = \begin{cases} u(i+1,j) - u(i,j) & \text{if } i < n_x \\ 0 & \text{if } i = n_x \end{cases} \tag{1.86}$$

$$(\partial_2 u)(i,j) = \begin{cases} u(i,j+1) - u(i,j) & \text{if } j < n_y \\ 0 & \text{if } j = n_y \end{cases} \tag{1.87}$$

This definition allows to define the divergence properly by duality, imposing :

$$< \nabla u, p >_Y = - < u, div(p) >_X \tag{1.88}$$

Simple computation gives :

$$\begin{aligned}(div(p))(i,j) \quad &= \quad \begin{cases} p^1(i,j) - p^1(i-1,j) & \text{if } 1 < i < n_x \\ p^1(i,j) & \text{if } i = 1 \\ -p^1(i-1,j) & \text{if } i = n_x \end{cases} \\ &+ \quad \begin{cases} p^2(i,j) - p^2(i,j-1) & \text{if } 1 < j < n_y \\ p^2(i,j) & \text{if } j = 1 \\ -p^2(i,j-1) & \text{if } j = n_y \end{cases}\end{aligned} \tag{1.89}$$

Note that the operator div is surjective from Y to $X - \{(\gamma, \gamma, ..., \gamma), \gamma \in \mathbb{R}\}$.
Moreover it can be shown [9] that $||div||_2 \leq 2\sqrt{2}$.

A.2 Projections on weighted l^p-balls ($p \in \{1, 2, \infty\}$)

Until now, we supposed that we could do Euclidean projections on weigthed l^p balls. Some projection operators are not straightforward to implement and we propose solutions to that problem. Let $K = \{y \in X, |\lambda(y-f)|_p \leq \alpha\}$, with $\lambda \in [0, \infty]^n$. The problem of projection on K can be written analytically :

$$\Pi_K(x) = \arg\min_{\{y \in K\}} \left(|y - x|_2^2\right) \tag{1.90}$$

Let $\bar{y}$ denote the solution of (1.90). A first important remark that holds for any p is that if $\lambda_i = 0$, then $\bar{y}_i = x_i$. If $\lambda_i = \infty$ then $\bar{y}_i = f_i$. Thus in all projection algorithms the first step is to set all those known values. This allows to restrict our attention to the case $\lambda \in]0, \infty[^n$.

Projections on weighted l^∞-balls

The simplest projector is the one on weighted l^∞-balls. It writes in closed form :

$$\bar{y}_i = \begin{cases} x_i & \text{if } |\lambda_i(f_i - x_i)| \le \alpha \\ f_i + \frac{x_i - f_i}{|x_i - f_i|}\frac{\alpha}{\lambda_i} & \text{otherwise} \end{cases} \tag{1.91}$$

Projections on weighted l^1-balls

Up to a change of variable, the projection on a weighted l^1-ball writes :

$$\Pi_K(x) = \underset{\{u, |\lambda u|_1 \le \alpha\}}{\arg\min} |u - x|_2^2 \tag{1.92}$$

with $\lambda \in]0, \infty[^n$ and $\alpha > 0$.

- First notice that if $|\lambda x|_1 \le \alpha$, then $\bar{u} = x$.

- In the other cases, existence and uniqueness of a minimizer results from strict convexity of $|u - x|_2^2$ and convexity of K. There exists $\sigma \in [0, \infty[$ s.t. the solution of (1.92) is given by the solution of the Lagrangian problem :

$$\Pi_K(x) = \underset{u \in \mathbb{R}^n}{\arg\min} |u - x|_2^2 + \sigma|\lambda u|_1 \tag{1.93}$$

The solution of this problem is in closed form :

$$u(\sigma)_i = \begin{cases} x_i - sgn(x_i)\frac{\sigma\lambda_i}{2} & \text{if } |x_i| \ge \frac{\sigma\lambda_i}{2} \\ 0 & \text{otherwise} \end{cases} \tag{1.94}$$

Let $\Psi(\sigma) = |\lambda u(\sigma)|_1$. Our problem is to find $\bar{\sigma}$ such that $\Psi(\bar{\sigma}) = \alpha$. Ψ is a convex function (thus continuous) and decreasing. Moreover $\Psi(0) = |\lambda x|_1$, and $\lim_{\sigma \to \infty} \Psi(\sigma) = 0$. From intermediary values theorem, for any $\alpha \in [0, |\lambda x|_1]$, there exists $\bar{\sigma}$ s.t. $\Psi(\bar{\sigma}) = \alpha$.

$$\begin{aligned} \Psi(\sigma) &= \sum_{i=1}^{n} |\lambda_i \bar{u}_i| && (1.95) \\ &= \sum_{i, |x_i| \ge \sigma\lambda_i/2} \lambda_i(|x_i| - \sigma\lambda_i/2) && (1.96) \\ &= \sum_{i, y_i \ge \sigma} \lambda_i |x_i| - \sigma\lambda_i^2/2 && (1.97) \end{aligned}$$

with $y_i = \frac{2|x_i|}{\lambda_i}$. Now, it is important to remark that Ψ is a piecewise linear decreasing function. The changes of slopes might only occur at values $\sigma = y_j$. Thus, an algorithm to find $\bar{\sigma}$ is the following :

1. For $i \in [1..n]$, compute $y_i = \frac{2|x_i|}{\lambda_i}$. [O(n) operations]
2. Using a *sort* function, store the permutation j s.t. $k \to y_{j(k)}$ is increasing. [O(n)log(n) operations]
3. Compute the partial sums : $\Psi(y_{j(k)}) = E(k) = \sum_{i=k}^{n} \lambda_{j(i)}|x_{j(i)}| - \frac{y_{j(k)}\lambda_{j(i)}^2}{2}$. E is decreasing. [O(n) operations]
4. – If $E(1) < \alpha$, set $a1 = 0$, $b1 = |\lambda x|_1$, $a_2 = y_{j(1)}$, $b2 = E(1)$. [O(1) operations]
   – Otherwise, find $\bar{k}$ s.t. $E(\bar{k}) \geq \alpha$ and $E(\bar{k}+1) < \alpha$. Set $a1 = y_{j(\bar{k})}$, $b1 = |E(\bar{k})|_1$, $a_2 = y_{j(\bar{k}+1)}$, $b2 = E(\bar{k}+1)$. [O(n) operations]
5. Set $\bar{\sigma} = \frac{(a_2-a_1)\alpha + b_2 a_1 - b_1 a_2}{b_2 - b_1}$. [O(1) operations]
6. Set $\bar{u} = u(\bar{\sigma})$ using (1.94). [O(n) operations]

Projections on weighted l^2-balls

The projection on a weighted l^2-ball (an ellipsoid) writes :

$$\Pi_K(x) = \underset{\{y, |\lambda y|_2^2 \leq \alpha^2\}}{\arg\min} |y - x|_2^2 \tag{1.98}$$

Contrarily to the l^∞ and l^1 cases, we do not propose an exact solution to this problem. We give an algorithm that leads to solutions that have 'the level of precision of the machine'.

- First notice that $\bar{y} = x$ if $|\lambda x|_2 \leq \alpha$.
- Otherwise it can be shown using Lagrange multipliers that the solutions of (1.98) writes :

$$\bar{y} = \frac{x}{\bar{\sigma}|\lambda|^2 + 1} \tag{1.99}$$

  for some parameter $\bar{\sigma} > 0$. Moreover, we know that $|\lambda \bar{y}|_2^2 = \alpha^2$. Let $\Psi(\sigma) = |\frac{\lambda x}{\sigma|\lambda|^2+1}|_2^2$. We are looking for a parameter $\bar{\sigma}$ s.t. $\Psi(\bar{\sigma}) = \alpha^2$. It can be shown that Ψ is convex decreasing. To find $\bar{\sigma}$ we used a Newton method. It writes :

  1. Set $k = 0$, $\sigma^k = 0$.
  2. Compute $\alpha^k = \Psi(\sigma^k)$.
  3. Compute $\beta^k = \Psi'(\sigma^k) = -2\sum_{i=1}^{n} \frac{\lambda_i^4 x_i^2}{(\sigma^k \lambda_i^2 + 1)^3}$.
  4. Set $\sigma^{k+1} = \sigma^k + \frac{(\alpha^2 - \alpha^k)}{\beta^k}$.
  5. Set $k = k + 1$, go back to 2 until $\alpha^k - \alpha^2 \leq \epsilon$.
  6. Set $\bar{y} = \frac{x}{\sigma^k|\lambda|^2+1}$.

Theoretically, this scheme converges superlinearly. In all our numerical experiments on deconvolution, we never needed more than 15 iterations to get a 10^{-15} precision. The average number of iterations is 6. Thus, we think that the projection on a weighted l^2 ball can be thought of as an $O(n)$ algorithm.

References

[1] M. Allain, J. Idier, and Y. Goussard. On global and local convergence of half-quadratic algorithms. *IEEE Trans. on Image Processing*, 15(5):1130–1142, 2006.

[2] S. Alliney. A Property of the Minimum Vectors of a Regularizing Functional Defined by Means of the Absolute Norm. *IEEE T. Signal Proces.*, 45:913–917, 1997.

[3] F. Alter, S. Durand, and J. Froment. Adapted Total Variation for Artifact Free Decompression of Jpeg Images. *JMIV*, 23:199–211, 2005.

[4] JF. Aujol. Contribution à l'analyse de textures en traitement d'images par méthodes variationnelles et équations aux dérivées partielles. *Thèse de l'université de Nice-Sophia-Antipolis*, 2004.

[5] J.F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Image Decomposition into a Bounded Variation Component and an Oscillating Component. *Journal of Mathematical Imaging and Vision*, 22:71–88, 2005.

[6] M. Bertalmio, V. Caselles, B. Rougé, and A. Solé. Total variation image restoration with local constraints. *Journal of scientific computing*, 19, 2003.

[7] Peter Blomgren and Tony F. Chan. Color TV: Total Variation Methods for Restoration of Vector Valued Images. *IEEE Transactions on Image Processing*, 7(3), 1998.

[8] E. Candes and F. Guo. New multiscale transforms, minimum total variation synthesis:application to edge-preserving image reconstruction. *Signal Processing*, 82(11):1519–1543, 2002.

[9] A. Chambolle. An Algorithm for Total Variation Minimization and Applications. *JMIV*, 20:89–97, 2004.

[10] A. Chambolle. Total variation minimization and a class of binary MRF models. *Preprint CMAP*, 578, 2005.

[11] T. Chan and S. Esedoglu. Aspects of total variation regularized L^1 function approximation. *SIAM J. Appl. Math.*, pages 1817–1837, 2005.

[12] Tony F. Chan, Gene H. Golub, and Pep Mulet. A nonlinear primal-dual method for total variation-based image restoration. *SIAM Journal on Scientific Computing*, 20(6), 1999.

[13] R. R. Coifman and A. Sowa. Combining the calculus of variations and wavelets for image enhancement. *Applied and computational harmonic analysis*, 9(1):1–18, 2000.

[14] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Journal on Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

[15] J. Darbon and M. Sigelle. Image Restoration with Discrete Constrained Total Variation Part I: Fast and Exact Optimization . *Journal of Mathematical Imaging and Vision*, 26(3), 2006.

[16] S. Durand and M. Nikolova. Denoising of frame coefficients using L1 data-fidelity term and edge-preserving regularization. *SIAM Journal MMS*, To appear.

[17] Sylvain Durand and Jacques Froment. Reconstruction of wavelet coefficients using total variation minimization. *SIAM Journal of Scientific Computing*, 24(5):1754–1767, 2003.

[18] I. Ekeland and R. Temam. Convex analysis and variational problems. *Studies in Mathematics and its Applications, American Elsevier Publishing Company*, 1976.

[19] Hayoing Fu, Michael K. NG, M. Nikolova, and J.L. Barlow. Efficient Minimization of Mixed $L^2 - L^1$ and $L^1 - L^1$ Norms for Image Restoration. *SIAM J. of Scientific Computing*, 3, 2005.

[20] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transaction on Image Processing*, 7:932–946, 1995.

[21] Donald Goldfarb and Wotao Yin. Second-Order Cone Programming Methods for Total Variation Based Image Restoration. *SIAM J. Scientific Computing*, 27:622–645, 2005.

[22] M. Hintermüller and G. Stadler. An infeasible primal-dual algorithm for TV-based inf-convolution-type image restoration. *SIAM Journal of Scientific Computing*, 28, 2006.

[23] K.C. Kiwiel. The efficiency of subgradient projection methods for convex optimization. Part I: General level methods and Part II: Implementations and extensions. *SIAM J. Control Optim.*, 34:660–697, 1996.

[24] D. Krishnan, P. Lin, and X.C. Tai. An efficient operator splitting method for noise removal in images. *Commun. Comp. Phys.*, 1(5):847–858, 2006.

[25] Y. Li and F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Transactions on Image Processing*, 5, 1996.

[26] S. Lintner and F. Malgouyres. Solving a variational image restoration model which involves L^∞ constraints. *Inverse Problems*, 20:815–831, 2004.

[27] Jian Luo and Patrick Combettes. A Subgradient Projection Algorithm For Non-Differentiable Signal Recovery. *NSIP*, pages 452–456, 1999.

[28] Jian Luo and Patrick Combettes. An adaptive level set method for nondifferentiable constrained image recovery. *IEEE Transactions on Image Processing*, 11:1295–1304, 2002.

[29] F. Malgouyres and F. Guichard. Edge direction preserving image zooming: A mathematical and numerical analysis. *SIAM Journal on Numerical Analysis*, 39(1):1–37, 2002.

[30] Yves Meyer. Oscillating patterns in image processing and in some nonlinear evolution equations. *The Fifteenth Dean Jaqueline B. Lewis Memorial Lectures*, 2001.

[31] A.S. Nemirovskii and D.B. Yudin. Problem Complexity and Method Efficiency in Optimization. *Wiley, New-York*, 1983.

[32] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[33] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(\epsilon^{-2})$. *Doklady AN SSSR*, 269(3):543–547, 1983.

[34] Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematic Programming, Ser. A*, 103:127–152, 2005.

[35] M. Nikolova. A variational approach to remove outliers and impulse noise. *Math. Imag. Vis.*, 20:99–120, 2004.

[36] M. Nikolova and R. Chan. The equivalence of half-quadratic minimization and the gradient linearization iteration. *IEEE Trans. on Image Processing*, To appear.

[37] SJ. Osher, A. Sole, and LA. Vese. Image Decomposition and Restoration using Total Variation Minimization and the H^{-1} Norm. *Multiscale Modeling and Simulation : SIAM*, pages 349–370, 2003.

[38] S.J. Osher and LA. Vese. Modeling Textures with Total Variation Minimization and Oscillating Patterns. *J. Sci. Comput.*, pages 553–572, 2003.

[39] Boris T. Polyak. Introduction to Optimization. *Translation Series in Mathematics and Engineering, Optimization Software*, 1987.

[40] L. Rudin and S. Osher. Total variation based image restoration with free local constraints. *Proc. IEEE ICIP*, 1:31–35, 1994.

[41] L. Rudin, S. Osher, and E. Fatemi. Nonlinear Total Variation Based Noise Removal. *Physica D*, 60:259–268, 1992.

[42] J.-L. Starck, M. Elad, and D.L. Donoho. Image Decomposition Via the Combination of Sparse Representation and a Variational Approach. *IEEE Transaction on Image Processing*, 14:1570–1582, 2005.

[43] S. Tramini, M. Antonini, M. Barlaud, and G. Aubert. Quantization Noise Removal for Optimal Transform Decoding. *International Conference on Image Processing*, pages 381–385, 1998.

[44] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.

[45] P. Weiss, G. Aubert, and L. Blanc-Féraud. Some Applications of l^∞- Constraints in Image Processing. *INRIA Research Report*, (6115), 2006.

[46] Wotao Yin, Donald Goldfarb, and Stanley Osher. A Comparison of Total Variation Based Texture Extraction Models. *to appear in Journal of Visual Communication and Image Representation*, 2006.

[47] T. Zeng and F. Malgouyres. Primal/dual implementation of a Basis Pursuit model. *CCSD Preprint*, 2007.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399