EFFICIENT SCHEMES FOR TOTAL VARIATION MINIMIZATION UNDER CONSTRAINTS IN IMAGE PROCESSING

PIERRE WEISS , LAURE BLANC-FÉRAUD*, AND GILLES AUBERT†

Abstract. This paper presents new fast algorithms to minimize total variation and more generally l^1-norms under a general convex constraint. Such problems are standards of image processing. The algorithms are based on a recent advance in convex optimization proposed by Yurii Nesterov. 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 $O\left(\frac{1}{\epsilon^2}\right)$ iterations at worst. We propose a scheme that allows to obtain a solution of precision ϵ in $O\left(\frac{1}{\epsilon}\right)$ iterations for a general convex constraint. For a strongly convex constraint, we solve a dual problem with a scheme that requires $O\left(\frac{1}{\sqrt{\epsilon}}\right)$ iterations to get a solution of precision ϵ. Finally we perform some numerical experiments which confirm the theoretical results on various problems of image processing.

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.

AMS subject classifications. 65K05, 65K10, 68U10, 94A08

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

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

or the "equivalent" Lagrangian form

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

$\mathbb{R}^n$ is the discrete space of $2D$ images (n is the number of pixels). $||\nabla u||_1$ corresponds to the discrete total variation (see the appendix for the discretization of differential operators). F is a convex proper function. We will give a particular attention to functions F that write as l^p-norms of affine transforms of the images.

In section (3) we review the applications of such a formalism and show that it is widely used. This is certainly due to the fact that total variation has interesting theoretical properties and leads to good 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 [46, 47], semi-implicit [29] or fixed point [50] schemes. Others are based on the minimization of a discretized energy. Those include subgradient descents [17] and subgradient projections [18], Newton-like methods [30], second order cone programming [26], interior point methods [24], or

*ARIANA, projet commun CNRS/INRIA/UNSA, INRIA Sophia Antipolis, 2004, route des Lucioles, BP93, 06902, Sophia Antipolis Cedex, France (pierre.armand.weiss@gmail.com, laure.blanc_feraud@sophia.inria.fr)

†Laboratoire J.A.Dieudonné , UMR CNRS 6621, Université de Nice Sophia-Antipolis, Parc Valrose 06108 Nice Cedex 2, France (gaubert@math.unice.fr)

graph based approaches [20, 12]. Recently, some authors tried to use primal-dual or dual-only approaches [14, 27, 11].

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 [36, 37]. 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 [37] and use a smooth approximation of the total variation. Doing so, getting a solution of precision ϵ requires $O\left(\frac{1}{\epsilon}\right)$ iterations while most first order schemes require $O\left(\frac{1}{\epsilon^2}\right)$ 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 $O\left(\frac{1}{\sqrt{\epsilon}}\right)$ iterations to get an ϵ-solution.

The outline of the paper is as follows :

- In section (2) we settle the main notations and definitions.
- In section (3) we show that many image processing problems such as restoration or decomposition can be expressed as (1.1) or (1.2).
- In section (4) we analyse two commonly used first order approaches to solve problem (1.1).
- In section (5) we detail the proposed algorithm to solve the constrained problem (1.1). It is based on a regularization of the total variation followed by a fast Nesterov's algorithm. Its convergence rate outperforms the other classical schemes by one order of magnitude.
- In section (6) we give an algorithm that solves the lagrangian problem (1.2) for strongly convex function F. It is based on the resolution of a dual problem with a Nesterov's scheme.
- In section (7), we finally compare our approach with some other existing first order methods.

2. Notations and definitions.

2.1. Notations. 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 [8]. To simplify the notations, 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.
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 $g_i = (g_{i,1}, g_{i,2})$.
$\langle \cdot, \cdot \rangle_X$ denotes the usual scalar product on X. For $u, v \in X$ we have

$$\langle u, v \rangle_X := \sum_{i=1}^{n} u_i v_i. \tag{2.1}$$

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

$\langle \cdot, \cdot \rangle_Y$ denotes the usual scalar product on Y. For $g, h \in Y$

$$\langle g, h \rangle_Y := \sum_{i=1}^{n} \sum_{j=1}^{2} g_{i,j} h_{i,j}. \tag{2.2}$$

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

$$|u|_p := \left(\sum_{i=1}^{n} |u_i|^p \right)^{1/p}. \tag{2.3}$$

$|\cdot|_\infty$ is the l^∞-norm on X

$$|u|_\infty = \max_{i \in \{1,2,\ldots,n\}} (|u_i|). \tag{2.4}$$

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

$$||g||_p := \left(\sum_{i=1}^{n} |g_i|_2^p \right)^{1/p} \tag{2.5}$$

and

$$||g||_\infty := \max_{i \in \{1,2,\ldots,n\}} (|g_i|_2). \tag{2.6}$$

Let A be a linear invertible transform. A^* denotes its complex conjugate. A^{-*} denotes the complex conjugate of A^{-1}.
Finally $\lfloor a \rfloor$ is the integer part of $a \in \mathbb{R}$.

2.2. Definitions and some recalls of convex optimization [23, 36, 45].

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*

$$\chi_K(x) = \begin{cases} 0 & \text{if } x \in K \\ \infty & \text{otherwise} \end{cases} \tag{2.7}$$

.

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*

$$\partial J(u) = \{\eta \in X, J(u) + \langle \eta, (x-u)\rangle_X \leq J(x), \forall x \in X\}. \tag{2.8}$$

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

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 $\sigma \rangle 0$ such that*

$$\langle \nabla F(u) - \nabla F(v), u - v\rangle_X \geq \frac{\sigma}{2}|u - v|_2^2 \tag{2.9}$$

for any $(u, v) \in K^2$. σ is called the convexity parameter of F. Note that property (2.9) 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*

$$G^*(y) = \sup_{x \in X}(\langle x, y\rangle_X - G(x)). \tag{2.10}$$

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

DEFINITION 2.9 (ϵ-solutions). *Let $\bar{u}$ be a solution of (1.1). An ϵ-solution of problem (1.1), is an element u_ϵ of $\{u, F(u) \leq \alpha\}$ satisfying*

$$||\nabla u_\epsilon||_1 - ||\nabla \bar{u}||_1 \leq \epsilon$$

Let $\bar{u}$ be a solution of (1.2). An ϵ-solution of problem (1.2), is an element u_ϵ of X satisfying

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

3. 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 [47] 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. In this section, we give a non exhaustive review of the different applications in which it is involved. We give a particular attention to functions F that write

$$F(u) = |\lambda(Au - f)|_p \tag{3.1}$$

where A is a linear invertible transform (identity, wavelet transform, Fourier transform,...), λ is a *diagonal* matrix whose elements belong to $[0, \infty]$, p belongs to $\{1, 2, \infty\}$, and f is a given datum. Let us show that this formalism covers a wide range of applications.

3.1. $A = Id$**,** $p \in \{1, 2, \infty\}$ **- 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 using the Maximum A Posteriori estimator is to solve the following problem

$$\inf_{u \in X, |u-f|_p \leq \alpha} (J(u)) \tag{3.2}$$

with $p = 1$ for impulse noise [2, 41, 13, 20], $p = 2$ for Gaussian noise [47], $p = \infty$ for uniform noise [51], 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

$$\inf_{u \in X, |\lambda(u-f)|_p \leq \alpha} (J(u)) \tag{3.3}$$

where $\lambda = diag(\lambda_i)$ with $\lambda_i \in [0, \infty]$ is a diagonal matrix 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 [46, 7]. This also allows to do tasks like inpainting [15]. Recently, the $BV - l^1$ model was also shown to be an efficient model for the decomposition of an image into a cartoon and a texture [52].

3.2. $A =$ **wavelet transform,** $p \in \{1, \infty\}$**.** In this part, we describe three applications. Namely, the restoration of compressed images, the restoration of images that have been thresholded in the wavelet domain and the denoising of white noises.

- 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

  $$\Psi(u) = A^{-1}(Q(Au)) \tag{3.4}$$

  where Q is a uniform or non uniform quantizer and A is a linear transform (local cosine transform, wavelet transform,...). A natural way to recover u, is to look for the image of minimal total variation in the convex set $\Psi^{-1}(f)$ [4, 21]. This amounts to solving

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

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

  $$\inf_{u \in X, |\lambda A(u-f)|_\infty \leq 1} (J(u)) \tag{3.5}$$

  with the diagonal coefficients of λ belonging to $[0, \infty]$.

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

- 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 (3.5), (A being a wavelet transform) can be shown to reduce those artefacts.
- Recently a model similar to (3.5), with an l^1-norm instead of the l^∞-norm was proposed for image denoising [22]. We refer the reader to [22] for further details.

3.3. A = Fourier transform, $p = 2$ - Image deconvolution, image zooming.

- 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

$$\inf_{u\in X, |h\star u-f|_2\leq\alpha} (J(u))\,. \tag{3.6}$$

  The operator $h\star$ is linear, it can thus be represented by a $n\times n$ matrix H. It is shown in [39] that the FFT diagonalizes H if $\star$ denotes the convolution operation with periodic boundary conditions and the DCT diagonalizes H if h is symmetric and $\star$ denotes the convolution with Neumann boundary conditions. In any case, we see that $H = A^{-1}\lambda A$, λ being a diagonal matrix and A denoting either the FFT, either the DCT. As both transforms are isometries from X to X, we have

$$|h\star u - f|_2 = |Hu - f|_2 = |\lambda Au - Af|_2. \tag{3.7}$$

  Finally (3.6) is equivalent to

$$\inf_{u\in X, |\lambda Au-Af|_2\leq\alpha} (J(u))\,. \tag{3.8}$$

- 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

$$\inf_{u\in X, |\lambda(Au-f)|_2\leq\alpha} (J(u)) \tag{3.9}$$

  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 [32].

3.4. Summary. We summarize the applications detailled previously in table 3.1.

This formalism also allows to do image cartoon + texture decompositions [33], inpainting [15] and restoration with perturbed sampling [3, 10]. Considering pseudo-invertible transforms it would include other interesting applications like denoising using redundant transforms (dictionnaries, curvelets, ...) [31, 9] or image decompositions [48]. Let us now look at the numerical algorithms to solve (1.1) and (1.2).

TABLE 3.1
Summary of the problems covered by our formalism.

	$p=1$	$p=2$	p=∞
A =Identity	Impulse noise denoising, Image decomposition [2, 41, 13, 20]	Gaussian noise denoising [47]	Bounded noise denoising [51]
A =Fourier transform	Robust deconvolution [24]	Image deconvolution [46, 14, 7], Image zooming [32]	No known reference
A = Wavelet, local cosine transform	Image decomposition or denoising [22]	Image denoising (No known reference)	Compression noise denoising [4, 49, 16]

4. Classical first order approaches. 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 show that two commonly used approaches require $O\left(\frac{1}{\epsilon^2}\right)$ iterations to provide an ϵ-solution. With such a rate getting a 10^{-3}-solution requires (on the worst case) of order 10^6 iterations.

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

$$\left\{ \begin{array}{l} u^0 \in K \\ u^{k+1} = \Pi_K \left(u^k - t^k \frac{\eta^k}{|\eta^k|_2} \right) \end{array} \right. . \tag{4.1}$$

Here, $t^k > 0$ for any k, η^k is any element of $\partial J(u^k)$ (see (2.8)) and Π_K is the Euclidean projector on $K = \{u, F(u) \leq \alpha\}$. It was proposed recently in some image processing papers [4, 17]. 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 [34] that any algorithm only using the sequences $J(u^k)$ and $\partial J(u^k)$ has a worst case complexity of $O\left(\frac{1}{\epsilon^2}\right)$. We refer the reader to [28] and [17] for optimal choices of the sequence $\{t_k\}$ in algorithm (4.1). Let us finally precise that scheme (4.1) might converge much faster if J belongs to certain function classes (see for instance [45]), but total variation does not possess the required properties.

4.2. Smoothing and projected gradient descent. Another widely spread technique consists in smoothing the total variation [46, 50] by

$$\tilde{J}_\mu(u) = \sum_{i=1}^{n} \sqrt{|(\nabla u)_i|^2 + \mu^2} \tag{4.2}$$

and use a projected gradient descent with constant step to minimize it. Let us analyse its rate of convergence.

PROPOSITION 4.1. *The following algorithm:*

$$\left\{ \begin{array}{l} u^0 \in K \\ u^{k+1} = \Pi_K \left(u^k - \tau \mathrm{div} \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \mu^2}} \right) \right) \end{array} \right. \tag{4.3}$$

where $\tau = \frac{2\mu}{||\text{div}||_2^2}$ *and* $\mu = \frac{\epsilon}{n}$ *ensures that after* N *iterations* $|J(u^N) - J(\bar{u})| \leq \epsilon$ *with* $N \leq O\left(\frac{1}{\epsilon^2}\right)$.

The proof is given in the appendix. To get an ϵ-solution, we thus need to choose μ of order $\frac{\epsilon}{n}$ and N of order $O\left(\frac{1}{\epsilon^2}\right)$. The two strategies presented are widely used but require large computing times to get acceptable estimates of the solutions. In the following sections we introduce much faster algorithms.

5. A new algorithm to minimize the total variation under simple constraints. In [37], Y. Nesterov presents an efficient scheme to minimize non-differentiable convex functions on convex sets. His idea is as follows:

- Approximate the non-differentiable function by a differentiable one.
- Compensate the approximation error using a fast scheme adapted to differentiable functions.

In this section, we show how to apply his ideas to total variation problems.

5.1. How to smooth the total variation?. Following the ideas in [37], we use a smooth approximation of J. First note that

$$J(u) = \sup_{q \in Y, ||q||_\infty \leq 1} \left(\langle \nabla u, q \rangle_Y \right). \tag{5.1}$$

The approximation we propose writes

$$J_\mu(u) = \sup_{q \in Y, ||q||_\infty \leq 1} \left(\langle \nabla u, q \rangle_Y - \frac{\mu}{2} ||q||_2^2 \right) + \frac{n\mu}{2}. \tag{5.2}$$

This corresponds to the Moreau-Yosida regularization. It is easily shown that $J_\mu(u) = \sum_{i=1}^n \psi_\mu(|(\nabla u)_i|)$ with

$$\psi_\mu(x) = \begin{cases} |x| & \text{if } |x| \geq \mu \\ \frac{x^2}{2\mu} + \frac{\mu}{2} & \text{otherwise} \end{cases}. \tag{5.3}$$

ψ_μ is called Huber function. J_μ seems more appropriate than $\tilde{J}_\mu$ defined in (4.2), as both approximations are $\frac{||\text{div}||_2^2}{\mu}$-Lipschitz differentiable [3], but

$$0 \leq \tilde{J}_\mu(u) - J(u) \leq n\mu \tag{5.4}$$

while

$$0 \leq J_\mu(u) - J(u) \leq \frac{n\mu}{2}. \tag{5.5}$$

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

$$\nabla J_\mu(u) = -\text{div}\left(\Psi\right) \quad \text{with } \Psi_i = \begin{cases} \frac{(\nabla u)_i}{|(\nabla u)_i|} & \text{if } |(\nabla u)_i| \geq \mu. \\ \frac{(\nabla u)_i}{\mu} & \text{otherwise.} \end{cases} \tag{5.6}$$

In all numerical experiments we perform, the minimization of (5.3) leads to solutions that have a lower total variation than (4.2), 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.

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

5.2. Nesterov's scheme for differentiable function. In [37], Y. Nesterov presents an $O\left(\frac{1}{\sqrt{\epsilon}}\right)$ algorithm adapted to the problem

$$\inf_{u\in K}\left(E(u)\right) \tag{5.7}$$

where E is any convex, L-Lipschitz differentiable function, and K 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 $O\left(\frac{1}{\sqrt{\epsilon}}\right)$ uniformly on all problems of type (5.7). Y. Nesterov's algorithm is thus optimal for this class of problems. In this algorithm, two sequences $\{x^k\}, \{y^k\} \in K$ are updated recursively in order to satisfy $\forall x \in K$

$$A^k E(y^k) \leq \frac{L}{2}||x - x^0||^2 + \sum_{i=0}^{k} \alpha^i (E(x^i) + \langle \nabla E(x^i), x - x^i\rangle_X). \tag{5.8}$$

In this equation α^i is a sequence of increasing coefficients and $A^k = \sum_{i=0}^{k} \alpha^i$ will define the rate of convergence. The right-hand side of (5.8) is an approximation of $A^k E(x)$. The linear part is a lower approximation of $A^k E(x)$ and a fortiori of $A^k E(\bar{x})$. Condition (5.8) thus ensures that $E(y^k) - E(\bar{x}) \leq \frac{L}{2A^k}||\bar{x} - x^0||^2$.

The idea underlying this condition is to exploit the fact that the gradient of a convex function not only gives its local ascent direction but also indicates facts about its global topological properties. It is thus possible - as in the conjugate gradient algorithm - to accelerate the convergence rate of the first order schemes by using the information brought by the gradients at all iterations. That is why the right-hand side of (5.8) is a linear combination of the gradients. Based on these ideas, Y. Nesterov shows the following result in his paper [37].

THEOREM 5.1. *Let $\bar{u}$ be a solution of (5.7). The following algorithm*

***Algorithm 1**: Accelerated gradient descent*

Input: *Number of iterations N, a starting point $x^0 \in K$.*
Output: *y^N an estimate of $\bar{u}$.*

begin
  Set $G^{-1} = 0$
  Set L =Lipschitz constant of ∇E
  for *k going from 0 to N* **do**
    Set $\eta^k = \nabla E(x^k)$.
    Set $y^k = \arg\min_{y\in K}\left(\langle \eta^k, y - x^k\rangle_X + \frac{1}{2}L||y - x^k||^2\right)$.
    Set $G^k = G^{k-1} + \frac{k+1}{2}\eta^k$.
    Set $z^k = \arg\min_{z\in K}\left(\frac{L}{\sigma}d(z) + \langle G^k, z\rangle_X\right)$.
    Set $x^{k+1} = \frac{2}{k+3}z^k + \frac{k+1}{k+2}y^k$.
  end
end

ensures that

$$0 \leq E(y^k) - E(\bar{u}) \leq \frac{4Ld(\bar{u})}{\sigma(k+1)(k+2)}. \tag{5.9}$$

At step 6, $||\cdot||$ *denotes any norm, at step* 8, d *is any convex function satisfying* $d(x) \geq \frac{\sigma}{2}||x - x_0||^2$ *for some element* $x_0 \in K$. σ *is the convexity parameter of* d.

Using inequality (5.9), it is easily seen that getting an ϵ-solution does not require more than $\sqrt{\frac{4Ld(\bar{u})}{\epsilon}}$ iterations. This shows that (1) is an $O\left(\frac{1}{\sqrt{\epsilon}}\right)$ 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 [36] for a precise definition of its optimality). Let us remind that the classical projected gradient descent is an $O\left(\frac{1}{\epsilon}\right)$ algorithm.

5.3. Solving the constrained problem for some convex functions F. The scheme we propose consists in solving

$$\inf_{u \in \mathbb{R}^n, F(u) \leq \alpha} (J_\mu(u)) \tag{5.10}$$

with algorithm (1). Let us precise how to achieve steps 3 and 5 and how to choose the regularization parameter μ.

5.3.1. How to achieve steps 6 and 8?. To apply algorithm (1), 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 the center of K or an estimate of the solution. With this choice, the convexity parameter of d is $\sigma = 1$. We have to find - preferably in closed form - the expressions of the arg min at steps 3 and 5.

PROPOSITION 5.2. *With the above choices, step* 6 *reduces to* $y^k = \Pi_K\left(x^k - \frac{\eta^k}{L}\right)$ *and step* 8 *reduces to* $z^k = \Pi_K\left(x_0 - \frac{G^k}{L}\right)$. Π_K *stands for the Euclidean projector on* K.

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

5.3.2. How to choose μ and the number of iterations?. In [40], the author shows that the singularity at 0 of the l^1-norm is responsible for the so-called staircase effect: the solutions of total variation problems are - roughly speaking - piecewise constant. A way to avoid that is to use a regularized operator such as (5.3). In practice, in the case of denoising, this leads to better SNR and more satisfying visual results. Figures (5.1) illustrate this fact. Lena image is degraded adding Gaussian noise with different variances. Then it is denoised using various μ parameters. It can be seen that the optimal μ value is around 0.002 independently of the initial SNR. In restoration applications, for natural images of amplitude 1, our experiments led us to the conclusion that the optimal μ should be taken in the range $[0.001, 0.005]$ and that few visual differences are observed in that range.

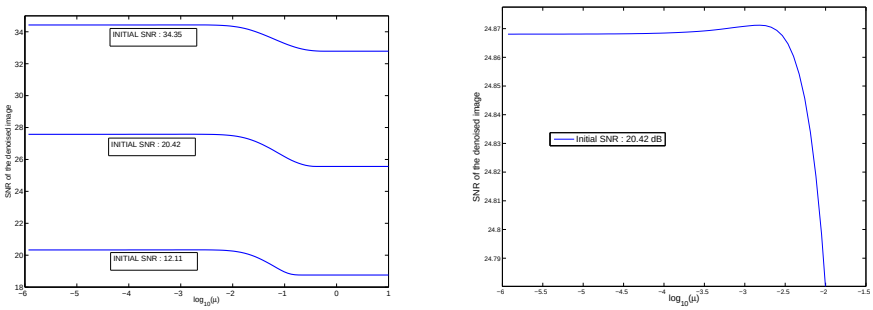

FIGURE 5.1. *SNR of the denoised image w.r.t. the μ parameter.*

In some situations we would like to exactly minimize the total variation ($\mu = 0$). Making a regularization might thus seem unappropriate. Actually the following proposition shows that smoothing the total variation is still a good solution.

PROPOSITION 5.3. *Let $K = \{u \in X, F(u) \leq \alpha\}$ and $D = \max_{u \in K}(d(u))$. Algorithm (1) applied to problem (5.10) with $\mu = \frac{\epsilon}{n}$ and $N = \lfloor \frac{2\sqrt{2}||\mathrm{div}\,||_2\sqrt{Dn}}{\epsilon} \rfloor + 1$ ensures that $|J(y^N) - \bar{J}| \leq \epsilon$ where $\bar{J}$ denotes the infimum of (1.1).*

Proof. Let $\bar{J}_\mu$ denote the solution of (5.10), let $\bar{u}$ be the solution of problem (1.1) and $L = \frac{||\mathrm{div}\,||_2^2}{\mu}$. We have

$$J(y^k) \overset{(5.5)}{\leq} J_\mu(y^k) \tag{5.11}$$

$$\overset{(5.9)}{\leq} \bar{J}_\mu + \frac{4LD}{k^2} \tag{5.12}$$

$$\leq J_\mu(\bar{u}) + \frac{4LD}{k^2} \tag{5.13}$$

$$\overset{(5.5)}{\leq} \bar{J} + \frac{n\mu}{2} + \frac{4LD}{k^2}. \tag{5.14}$$

We thus obtain $0 \leq J(y^k) - \bar{J} \leq \frac{4LD}{k^2} + \frac{n\mu}{2}$. To obtain an ϵ-solution, it is thus sufficient to have $\frac{4LD}{k^2} + \frac{n\mu}{2} \leq \epsilon$. Setting $\mu \leq \frac{2\epsilon}{n}$ and $k = \lfloor \sqrt{\frac{4||\mathrm{div}\,||_2^2 D}{\mu(\epsilon - \frac{n\mu}{2})}} \rfloor + 1$ thus gives an ϵ-solution. To get the result, it suffices to maximize the denominator in the previous equation. □

We thus gain one order in the convergence rate compared to classical algorithms. It shows that smoothing the total variation is a good way to exploit its structure. 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 the bound (5.9) to define the number of iterations. Experimentally, for images rescaled in $[0, 1]$, we can check that the solution of (5.10) obtained by choosing $\mu = 0.01$ is very close perceptually to the solution of (1.1). Choosing $\mu = 0.002$ leads to solutions that are perceptually identical to the solution of (1.1), independently of the problem dimension. From this remark, we infere that a sufficient precision ϵ lies in $[0.002\frac{n}{2}, 0.01\frac{n}{2}]$. Thus, using proposition (5.3) the approach we suggest consists in choosing $\mu \in [0.001, 0.005]$ and there is no reason to choose $N > \frac{2\sqrt{2}||\mathrm{div}\,||_2\sqrt{Dn}}{n\mu}$. In all

cases we studied (except deconvolution) $D \sim \theta n$, with $\theta \in]0,1[$. Thus, the maximum iterations needed to get a good approximate solution is

$$N = \frac{2\sqrt{2}||\text{div}\,||_2\sqrt{\theta}}{\mu}. \tag{5.15}$$

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.

5.4. Some application examples.

5.4.1. Restoration involving linear invertible transforms : $F(u) = |\lambda(Au-f)|_p$ with $p \in \{1,2,\infty\}$. We showed in section (3) that 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, λ is a diagonal matrix with elements in $[0,\infty]^n$, and f are the given data. To apply algorithm (1) to problem (5.10), we need to be able to compute projections on $\{u, |\lambda(Au-f)|_p \leq \alpha\}$. As this might be cumbersome, we use the change of variable

$$z = Au - f \tag{5.16}$$

and solve the equivalent problem

$$\inf_{z, |\lambda z|_p \leq \alpha} (E_\mu(z)) \tag{5.17}$$

with $E_\mu(z) = J_\mu(A^{-1}(z+f))$. The solution $\bar{u}$ of (5.10) can be retrieved from the solution $\bar{z}$ using formula $\bar{u} = A^{-1}(\bar{z}+f)$. It is easy to show that E_μ is L-Lipschitz differentiable with $L = \frac{||\text{div}\,||_2^2||A^{-1}||_2^2}{\mu}$. For all invertible transforms the final algorithm to solve (5.17) writes:

Algorithm 2: Y. Nesterov's scheme for problem (5.10)

Input: Number of iterations N and regularization parameter μ.
(depending on the precision required).
Output: u^N an estimate of $\bar{u}$.

begin
  Set $G^{-1} = 0$.
  Set $L = \frac{||\text{div}\,||_2^2||A^{-1}||_2^2}{\mu}$.
  Set $x^k = 0$.
  for k *going from* 0 *to* N **do**
    Set $\eta^k = A^{-*}\nabla J_\mu(A^{-1}(x^k+f))$.
    Set $y^k = \Pi_K\left(x^k - \frac{\eta^k}{L}\right)$.
    Set $G^k = G^{k-1} + \frac{k+1}{2}\eta^k$.
    Set $z^k = \Pi_K\left(-\frac{G^k}{L}\right)$.
    Set $x^{k+1} = \frac{2}{k+3}z^k + \frac{k+1}{k+2}y^k$.
  end
  Set $u^N = A^{-1}(y^{k-1}+f)$.
end

At steps 7 and 9, $K = \{u \in X, |\lambda u|_p \leq \alpha\}$. We refer the reader to the appendix for the expressions of the projections on weighted l^p-balls.

Figure 5.2 shows the result of a compression noise restoration using this technique for $\mu = 0.06$ (60 iterations until visual stability) and $\mu = 0.0004$ (210 iterations until visual stability). Clearly, for such bounded noises it is preferable to use the regularized total variation in order to avoid the staircase effect. This was already remarked in [51]. 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 (7) for comparisons with other algorithms. Let us note that to our knowledge, no precise schemes exist in the literature to solve this problem. Large oscillations are removed, while thin details are preserved. The main drawback of this model is that the contrast of the details decreases.

FIGURE 5.2. *Example of image decompression - TL: Original image (scaled im* $[0, 1]$*) - TR: Compressed image using Daubechies 9-7 wavelet transform (the implementation is similar to Jpeg2000) - BL: Solution of (3.5) using* $\mu = 0.006$ *- BR : Solution of (3.5) using* $\mu = 0.0004$

5.4.2. Deconvolution : $F(u) = |h \star u - f|_2$**.** In this paragraph we present a new way to do deconvolution using a Nesterov's scheme. This problem is particularly difficult and cannot be solved by the previous algorithm as the convolution matrices

are generally non-invertible or ill-conditioned. In (3.3), we showed that problem

$$\inf_{u\in X, |h\star u-f|_2\leq\alpha} (J_\mu(u)) \tag{5.18}$$

is equivalent to

$$\inf_{u\in X, |\lambda z-Af|_2\leq\alpha} \left(J_\mu(A^{-1}z)\right) \tag{5.19}$$

where A is the discrete cosine transform and λ is a diagonal matrix. In this formulation $z \to J_\mu(A^{-1}z)$ is L-Lipschitz-differentiable with $L = \frac{||\text{div}\,||_2^2}{\mu}$. The interests of this formulation are that we have a very fast Newton algorithm to do projections on $K = \{u \in X, |\lambda z - Af|_2 \leq \alpha\}$ (see the appendix paragraph (9.2.3)), and that the Lipschitz constant of the gradient of $z \to J_\mu(A^{-1}z)$ does not blow up. Note that the set $K = \{u \in X, |\lambda z - Af|_2 \leq \alpha\}$ might be unbounded if λ contains zeros on its diagonal. Thus we lose the convergence rate unless we estimate an upper bound on $d(\bar{u})$. Practically, the Nesterov scheme remains very efficient (see Figure (7.3)). The cost per iteration is around 2 DCTs and 2 projections on ellipsoids. For a 256×256 image, the cost per iteration is 0.2 seconds (we used the *dct*2 function of Matlab and implemented a C code with Matlab mex compiler for the projections). Figure (5.3) shows an experimental result. We display the bottom right result to show that it is useless (for visual purposes) to choose very small μ parameters. Perceptually, the bottom left picture is the same while the computing times needed to obtain it are much lower.

5.4.3. 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 [47]. It consisted in choosing $F(u) = |u-f|_2$. In [33], Y. Meyer studied this model theoretically, and figured out its limitation to discriminate 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 [33, 6] for more details. Y. Meyer defined a norm

$$||v||_G = \inf_{g\in Y} (||g||_\infty, \text{div}\,(g) = v) \tag{5.20}$$

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

$$\inf_{(u,v)\in X^2, f=u+v} (J(u) + \lambda||v||_G)\,. \tag{5.21}$$

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 problem. The first authors who tried to compute a solution were L. Vese and S. Osher in [44]. Later, other authors tackled this problem. Let us cite the works of J.F. Aujol et. al. in [5] and of D. Goldfarb et. al. in [26]. 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

$$\inf_{u\in X} \left(J(u) + \lambda \inf_{g\in Y, \text{div}\,(g)=f-u} (||g||_\infty)\right). \tag{5.22}$$

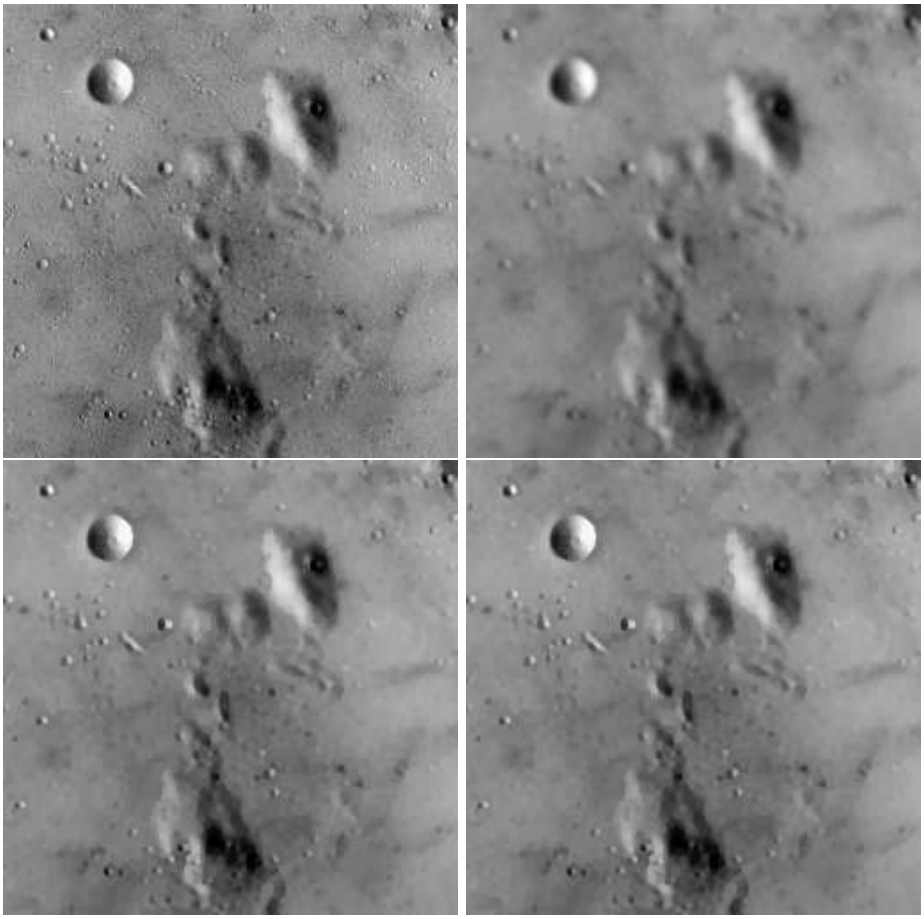

FIGURE 5.3. *Image deconvolution - TL : Original Image - TR : Convolved noisy image - BL : Solution of (5.19)* $\mu = 0.001$, $N = 150$ *- BR : Solution of (5.19)* $\mu = 10^{-7}$, $N = 10^5$

PROPOSITION 5.4. *Problem (5.22) can be reformulated as follows:*

$$\inf_{g \in Y, ||g||_\infty \leq \alpha} (J(f - \operatorname{div}(g))) \tag{5.23}$$

Proof. The idea simply is to use the change of variable $u = f - \operatorname{div}(g)$ in order 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

$$\inf_{u \in X} \left(J(u) + \lambda \inf_{g \in Y, \operatorname{div}(g) = f - u} (||g||_\infty) \right) = \inf_{g \in Y} (J(f - \operatorname{div}(g)) + \lambda ||g||_\infty). \tag{5.24}$$

Turning the Lagrange multiplier λ into a constraint, we get the result. □

Instead of solving problem (5.23), we solve

$$\inf_{g \in Y, ||g||_\infty \leq \alpha} (J_\mu(f - \operatorname{div}(g))) \tag{5.25}$$

and get an $O\left(\frac{1}{\epsilon}\right)$ algorithm. Note that the solution of (5.25) 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 (5.23), we get the model of Osher-Solé-Vese [43]. In Figure (5.6), we also show the result of (5.23) 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 and show that it competes with the $BV - l^1$ model. Formula (5.23) allows to easily constrain the properties of the g field. This might also be interesting for spatially varying processing.

FIGURE 5.4. *Image to be decomposed*

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 (5.5), 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 texture (v component) is bounded by a parameter depending linearly on α in (5.23). That might explain the deceiving result. On the given example, Osher-Solé-Vese's model gives more satisfying results. This was already observed in [52].

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 (5.23). 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 and that the new model better extracts the oscillating patterns (chair and clothes for instance). We think that the blurring effect is due to the numerical scheme which is slightly more diffusive for the new model as it is based on fourth order finite differences.

6. A new algorithm to solve the lagrangian problem for strongly convex data 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 (1) adapted to the unconstrained minimization of Lipschitz differentiable functions [35]. This method should be efficient, 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 $O\left(\frac{1}{\sqrt{\epsilon}}\right)$ algorithm rather

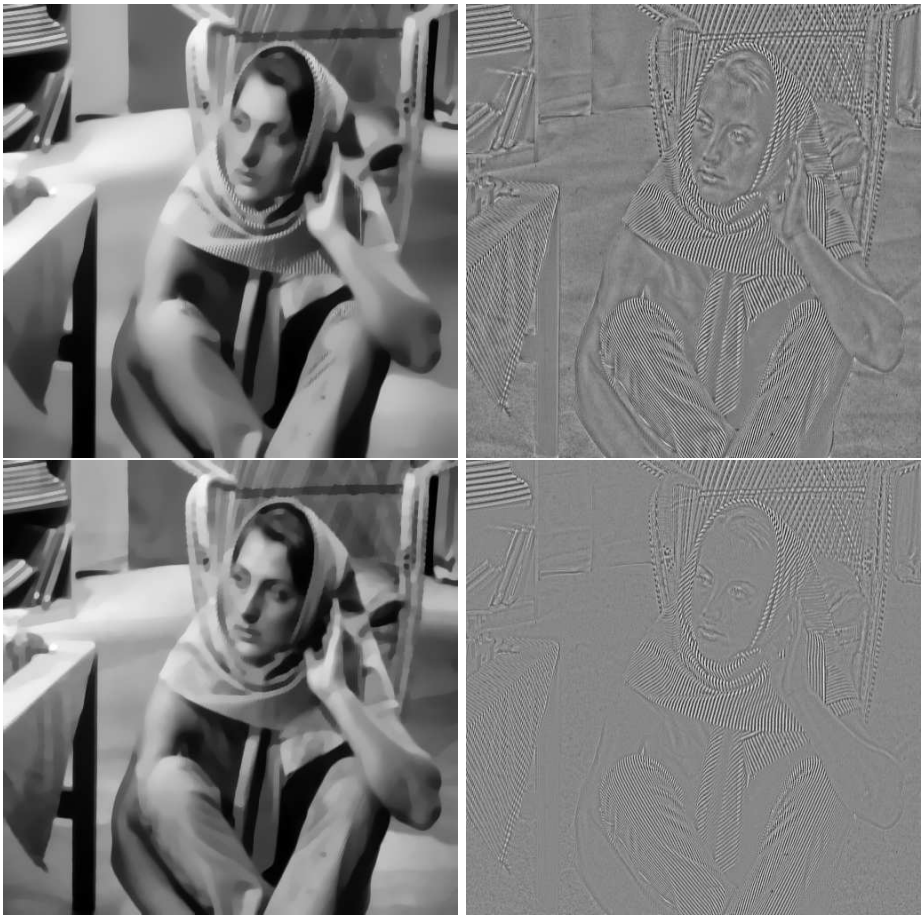

FIGURE 5.5. *Cartoon + texture decompositions. Top: Meyer's model. Bottom: Osher-Solé-Vese's model*

than the previous $O\left(\frac{1}{\epsilon}\right)$ algorithm. The proposed algorithm can notably solve the problem of Rudin-Osher-Fatemi with local constraints, the problem of deconvolution in the case of an invertible transform and the cartoon+texture decomposition model of Vese-Osher.

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 [23], for a complete reference).

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

$$\inf_{u \in X} (G(\nabla u) + F(u)). \tag{6.1}$$

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

$$\inf_{q \in Y} \left(G^*(-q) + F^*(-div(q))\right). \tag{6.2}$$

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

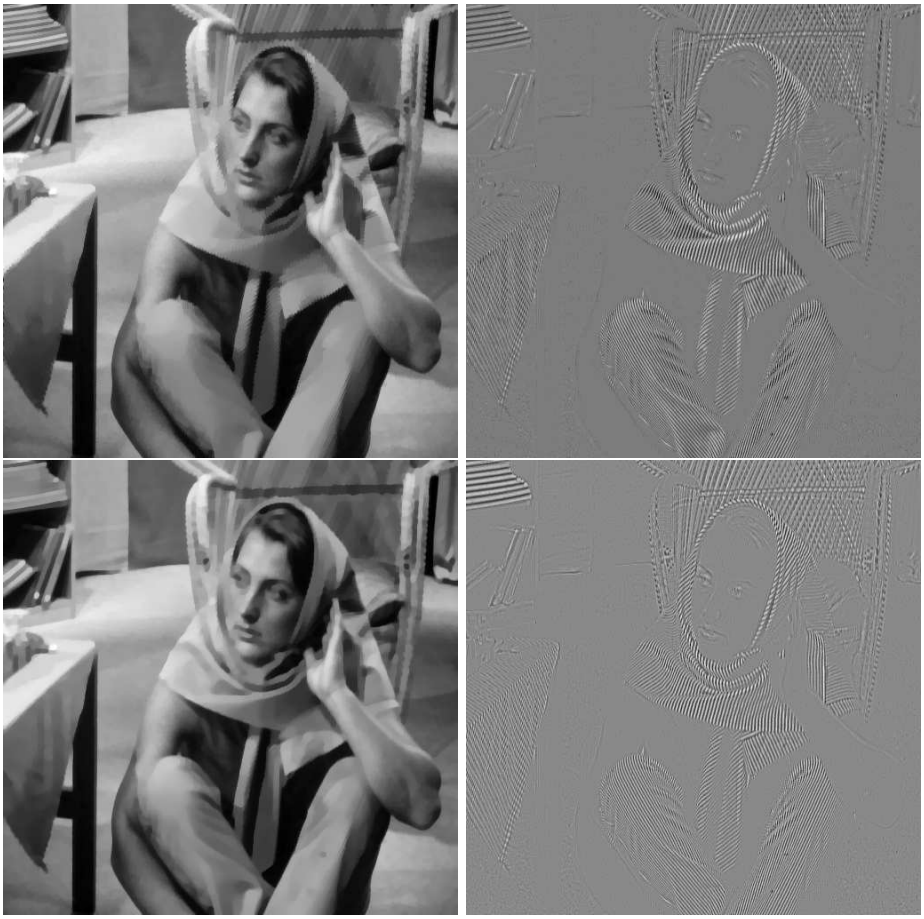

FIGURE 5.6. *Cartoon + texture decompositions. Top: $BV - l^1$ model. Bottom: result of minimizing the l^1-norm of the g field in (5.23)*

through the extremality relations

$$F(\bar{u}) + F^*(-div(\bar{q})) = \langle -div(\bar{q}), \bar{u} \rangle_X \tag{6.3}$$

and

$$G(\nabla \bar{u}) + G^*(-\bar{q}) = \langle -\bar{q}, \nabla \bar{u} \rangle_Y . \tag{6.4}$$

Furthermore we have

$$G(\nabla \bar{u}) + F(\bar{u}) = -(G^*(-\bar{q}) + F^*(-div(\bar{q}))). \tag{6.5}$$

6.0.4. Application to our problem. To apply the previous theory to our problem, we take

$$G(q) = ||q||_1 \tag{6.6}$$

We wish to solve

$$\inf_{u \in X} (G(\nabla u) + F(u)) \tag{6.7}$$

where F is differentiable, strongly convex, with convexity parameter σ.

THEOREM 6.1. *The dual problem of (6.7) is defined by*

$$\inf_{q\in K}\left(F^*(-div(q))\right) \tag{6.8}$$

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 (6.8) can thus be solved with a Nesterov scheme (no smoothing is needed!).* $\bar{u}$ *can be retrieved from the solution* $\bar{q}$ *of (6.8) using*

$$\bar{u} = \nabla F^*(-div(\bar{q})), \tag{6.9}$$

moreover

$$\nabla\bar{u} = \bar{q}|\nabla\bar{u}|. \tag{6.10}$$

This methods thus amounts to evolving the orientation of the level lines of u *instead of its intensity.*

Proof.

1. Let us compute G^*:

$$\begin{aligned} G^*(q) &= \sup_{r\in Y}\left(\langle q, r\rangle_Y - ||r||_1\right) && (6.11)\\ &= \sup_{t>0}\left(\sup_{||r||_1=t}\left(\langle q, r\rangle_Y - t\right)\right) && (6.12)\\ &= \sup_{t>0}\left(t||q||_\infty - t\right) && (6.13)\\ &= \chi_K(q) && (6.14) \end{aligned}$$

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

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

$$F^*(u) = \sup_{v\in X}\left(\langle u, v\rangle_X - F(v)\right) \tag{6.15}$$

   First, note that F^* is convex (see section 2). As F is strictly convex, the solution of problem (6.15) exists and is unique. Let $v(u)$ denote the arg max in (6.15). From uniqueness of the solution of (6.15), 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 $(u_1, u_2) \in X^2$

$$\nabla F(v(u_1)) - \nabla F(v(u_2)) = u_1 - u_2 \tag{6.16}$$

   and

$$\begin{aligned} |u_1 - u_2|_2 &= |\nabla F(v(u_1)) - \nabla F(v(u_2))|_2 && (6.17)\\ &\geq \frac{\sigma}{2}|v(u_1) - v(u_2)|_2 && (6.18)\\ &\geq \frac{\sigma}{2}|\nabla F^*(u_1) - \nabla F^*(u_2)|_2. && (6.19) \end{aligned}$$

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

3. Let us show (6.9). The first extremality relation gives

$$F(\bar{u}) = \langle -div(\bar{q}), \bar{u} \rangle_X - F^*(-div(\bar{q})).$$

We also recall that $F^{**}(u) = F(u)$. So that $F(\bar{u}) = \sup_{v \in X} (\langle \bar{u}, v \rangle_X - F^*(v))$. Those two equations imply that $-div(\bar{q})$ cancels the derivative of $v \rightarrow \langle \bar{u}, v \rangle_X - F^*(v)$. This ends the proof.

4. Finally let us show equation (6.10). It is done using the second extremality relation

$$\begin{aligned} G(\nabla \bar{u}) &= G^{**}(\nabla \bar{u}) && (6.20)\\ &= \sup_{q \in Y} (\langle \nabla \bar{u}, q \rangle_Y - G^*(q)) && (6.21)\\ &= \langle -\bar{q}, \nabla \bar{u} \rangle_Y - G^*(-\bar{q}). && (6.22) \end{aligned}$$

Thus $-\bar{q}$ solves problem (6.21). This yields the existence of multipliers μ_i such that

$$(\nabla \bar{u})_i - \mu_i \bar{q}_i = 0 \tag{6.23}$$

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 $\mu_i = |(\nabla \bar{u})_i|_2$.

□

6.0.5. Nesterov's algorithm in the general strongly convex case. Nesterov's algorithm applied to problem (6.8) writes:

Algorithm 3: Y. Nesterov's scheme for problem (6.8)

Input: Number of iterations N.
Output: u^N an estimate of $\bar{u}$.

begin
  Set $G^{-1} = 0$.
  Set $L = \frac{2||div||_2^2}{\sigma}$.
  Set $x^k = 0$.
  for *k going from* 0 *to* N **do**
    Set $\eta^k = \nabla(\nabla F^*(-div(x^k)))$.
    Set $y^k = \Pi_K \left(x^k - \frac{\eta^k}{L} \right)$.
    Set $G^k = G^{k-1} + \frac{k+1}{2} \eta^k$.
    Set $z^k = \Pi_K \left(-\frac{G^k}{L} \right)$.
    Set $x^{k+1} = \frac{2}{k+3} z^k + \frac{k+1}{k+2} y^k$.
  end
end

This algorithm returns a variable y^N that should be close to the solution of the dual problem. To retrieve the solution in the primal problem, we could thus set

$$u^N = \nabla F^*(-div(y^N)). \tag{6.24}$$

Furthermore, using (5.9), we get that

$$0 \leq F^*(-div(y^N)) - F^*(-div(\bar{q})) \leq \frac{8||div||_2^2 n}{\sigma(k+1)(k+2)} \tag{6.25}$$

which is a convergence rate on the dual variable. Another choice consists in using a linear combination of the solutions at all iterations. It leads to better practical efficiency and allows to prove convergence rates for the primal variables. Let us set $\bar{u}^N = \frac{2}{(N+1)(N+2)} \sum_{i=1}^N (i+1)u^i$ with $u^i = \nabla F^*(-div(x^i))$ and $\phi_*(u) = ||\nabla u||_1 + F(u)$. The following proposition summarizes the rates of convergence we obtain on the variable $\bar{u}^N$:

PROPOSITION 6.2. *Algorithm (3) ensures that*

$$0 \leq \phi_*(\bar{u}^N) - \phi_*(\bar{u}) \leq \frac{4||div||_2^2 n}{\sigma(N+1)(N+2)} \tag{6.26}$$

and

$$|\bar{u}^N - \bar{u}|_2 \leq \frac{2\sqrt{2}||div||_2\sqrt{n}}{\sigma N}. \tag{6.27}$$

The proof is given in the appendix.

6.0.6. 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 $\lambda = diag(\lambda_i)$ a diagonal matrix with $\lambda_i \in]0, \infty]$. Let $\lambda_- = \min_i \lambda_i$ and $\lambda_-(A)$ denote the smallest eigenvalue of A.

PROPOSITION 6.3. *F^* is L-Lipschitz differentiable, with $L \leq \frac{1}{2\lambda_-^2 \lambda_-(A)^2}$. Moreover*

$$F^*(v) = \langle A^{-1}f, v\rangle + \frac{1}{4}|\lambda^{-1}A^{-*}v|_2^2. \tag{6.28}$$

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 $\sigma = 4\lambda_-^2\lambda_-(A)^2$. Then

$$\begin{aligned} F^*(v) &= \sup_{u\in X} \left(\langle u, v\rangle_X - |\lambda(Au-f)|_2^2\right) && (6.29)\\ &= \sup_{w\in X} \left(\langle A^{-1}(\lambda^{-1}w + f), v\rangle - |w|_2^2\right) && (6.30)\\ &= \langle A^{-1}f, v\rangle + \sup_{r\geq 0} \left(r|\lambda^{-1}A^{-*}v|_2^2 - r^2|\lambda^{-1}A^{-*}v|_2^2\right) && (6.31)\\ & && (6.32) \end{aligned}$$

and we get the result by canceling the derivative of $r \to r - r^2$. ∎

To solve problem

$$\inf_{u\in X} \left(J(u) + |\lambda(Au-f)|_2^2\right) \tag{6.33}$$

the approach we propose consists in solving its dual problem (6.8) using a Nesterov algorithm. Setting $K = \{q \in Y, ||q||_\infty \leq 1\}$, the algorithm is as follows:

Algorithm 4: Y. Nesterov's scheme for problem (6.33)

Input: Number of iterations N.
Output: u^N an estimate of $\bar{u}$.

begin
- Set $G^{-1} = 0$.
- Set $L = \frac{||div||_2^2}{2\lambda_-^2 \lambda_-(A)^2}$
- Set $\bar{u}^N = 0$
- Set $x^k = 0$.
- **for** k *going from* 0 *to* N **do**
  - Set $\eta^k = \nabla(A^{-1}f) - \frac{1}{2}\nabla A^{-1}\lambda^{-2}A^{-*}div(x^k)$.
  - Set $y^k = \Pi_K\left(x^k - \frac{\eta^k}{L}\right)$.
  - Set $G^k = G^{k-1} + \frac{k+1}{2}\eta^k$.
  - Set $z^k = \Pi_K\left(-\frac{G^k}{L}\right)$.
  - Set $x^{k+1} = \frac{2}{k+3}z^k + \frac{k+1}{k+2}y^k$.
  - Set $\bar{u} = \bar{u} + (k+1)(A^{-1}f - \frac{1}{2}A^{-1}\lambda^{-2}A^{-*}div(x^N))$
- **end**
- Set $\bar{u}^N = \frac{2}{(N+1)(N+2)}\bar{u}$.

end

Note that in this formulation, we optimize a variable in $Y = X \times X$ instead of X. Using (6.26), we get that

$$\phi_*(\bar{u}^N) - \phi_*(\bar{u}) \leq \frac{||div||_2^2 n}{\lambda_-^2 \lambda_-^2(A) N^2} \tag{6.34}$$

7. 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 widely used first order methods. Namely: the projected gradient descent and the projected subgradient descent.

7.1. Some comparisons for the Rudin-Osher-Fatemi problem.. The problem we choose for numerical comparisons is the Rudin-Osher-Fatemi model. The reasons for this choice are that many recent papers give convergence results on that problem and that it allows to compare the primal and dual approaches. It consists in solving

$$\inf_{u \in X} \left(J(u) + \lambda^2 |u - f|_2^2\right) \tag{7.1}$$

or equivalently

$$\inf_{u\in X, |u-f|_2\leq\alpha} (J(u))\,. \tag{7.2}$$

Finding λ and α such that the solution of (7.1) is the same as that of (7.2) is not straightforward. To find those parameters, we let the Nesterov method applied to the dual problem of (7.1) run until convergence. This provides a solution $\bar{u}_\lambda$. Then we set $\alpha = |\bar{u}_\lambda - f|_2$. Let us describe the methods we implement for comparisons:

*Algo*1 is the presented *Nesterov approach* applied to the dual problem (6.8). This corresponds to algorithm (4).

*Algo*2 is the projected gradient applied to the dual problem (6.8). This approach is slightly faster than A. Chambolle's initial algorithm [11] (see [12] for a comparison).

*Algo*3 is the presented *Nesterov + smoothing approach* (2).

*Algo*4 is a *projected gradient descent with optimal constant step*

$$\begin{cases} u^0 = f \\ u^{k+1} = \Pi_K(u^k - t\nabla J_\mu(u^k)) \end{cases} \tag{7.3}$$

We set $K = \{u \in X, |u - f|_2^2 \leq \alpha^2\}$. The optimal step can be shown to be $t = \frac{2\mu}{||div||_2^2}$ [45].

We use the 256×256 Lena image rescaled in $[0, 1]$. We add a white gaussian noise ($\sigma = .15$). This corresponds to the images in figure (7.1). The bottom-left figure (BL) is given in order to show that the smoothing technique gives satisfying results in a really small number of iterations. The curves in figure (7.2) compare the distance from the current estimate to the minimizer w.r.t. the number of iterations for the different methods.

First note that the precision of the smoothing approaches (*Algo*3 and *Algo*4) is bounded below by a positive constant due to the approximation error. Therefore, it is useless - for a fixed regularization parameter μ - to iterate too much.

The dual problem solved with a Nesterov scheme (*Algo*1) clearly outperforms all tested approaches. *Algo*2 seems to be the second most efficient algorithm. However, it leads to precise solutions much slowlier.

In the primal formulation, Nesterov's algorithm (*Algo*3) outperforms the projected gradient descent (*Algo*4). In any case we see that it is preferable to use Nesterov's scheme compared to simpler schemes as the projected gradient descent. When it is possible (strongly convex data term), it is very interesting to solve a dual problem.

Let us finally precise that depending on the applications, the computational effort per iteration of Nesterov's technique is between one and twice that of the projected gradient descent.

Now let us compare more precisely the efficiency of the projected gradient descent applied to the dual of (7.1) (*Algo*2), with the "smoothing + Nesterov" technique (*Algo*3). In experiment (7.3), we treat the original Lena image and set $\lambda = 1$. We can see that for any iterations number k, there exists a pair (k, μ) such that the precision obtained by *Algo*2 is the same as that obtained by *Algo*3. It indicates that the "smoothing + Nesterov" algorithm has roughly the same efficiency as A. Chambolle's scheme. Note that this algorithm can be used in a much wider class of constrained problems.

Another interesting remark is that the best precision that can be obtained with the smoothing technique depends linearly on μ.

FIGURE 7.1. *TL: Original image (scaled in* $[0,1]$*) - TR: Noisy Lena - BL: Solution of (7.2) obtained using Algo*3 *and setting* $\mu = 0.01$ *and* $N = 50$ *iterations - BR: Exact solution of (7.2) obtained using Algo*1 *until convergence*

7.2. Comparisons for other constrained problems. In this paragraph, we focus on the primal formulations. Our aim is to compare three different algorithms for solving the constrained total variation problem (1.1) with different functions F. The tested algorithms are *Algo*3 and *Algo*4 setting $\mu = 0.001$ [4] and *Algo*5 which is described below:

*Algo*5 is a projected subgradient descent with optimal step (4.1). η^k must belong to $\partial J(u^k)$ for convergence. We choose:

$$\eta^k = -div(\Psi) \text{ with } \Psi_i = \begin{cases} \frac{(\nabla u^k)_i}{|(\nabla u^k)_i|_2} & \text{if } |(\nabla u^k)_i|_2 > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

and $t_k = \frac{J(u^k)-\bar{J}}{|\eta^k|_2}$. This step is optimal in the sense that it leads to an $O\left(\frac{1}{\epsilon^2}\right)$ algorithm. As $\bar{J}$ is unknown, some authors try to evaluate it iteratively

[4] this leads to solutions that are perceptually identical to the solutions using $\mu = 0$

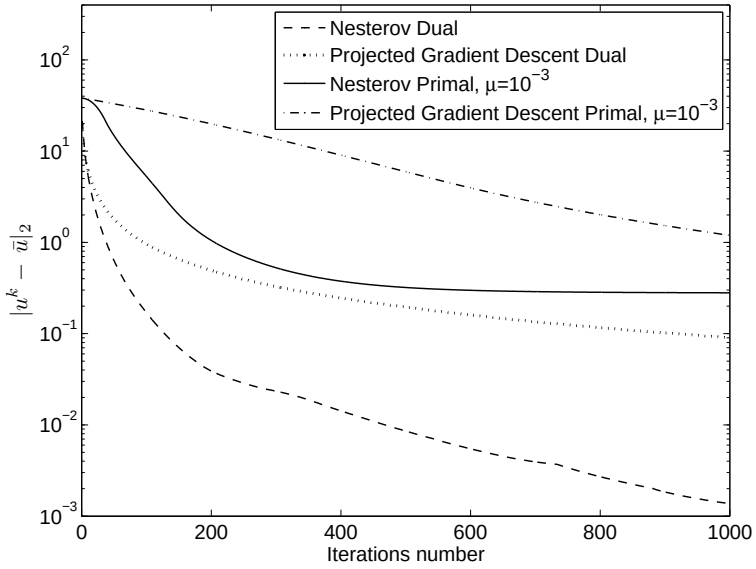

FIGURE 7.2. *Convergence rate comparison*

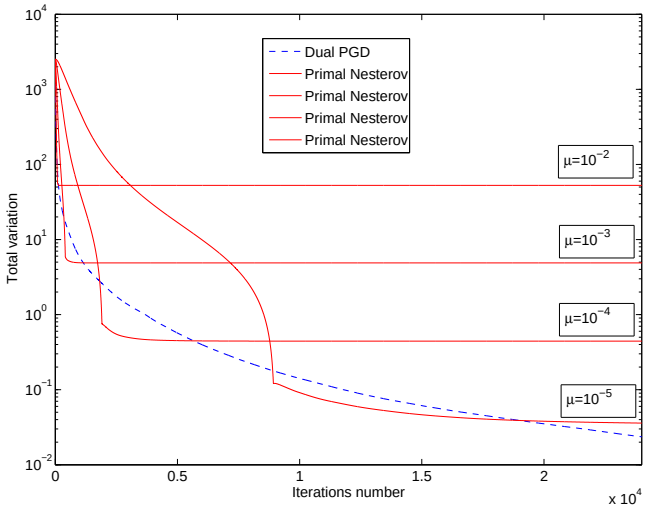

FIGURE 7.3. *Comparison of the projected gradient applied to the dual of (7.1) with the Nesterov's scheme applied to the smoothed version of (7.2)*

[28, 17]. 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.

We test the efficiency of the method under various constraints. The tested problems are:

- The Rudin-Osher-Fatemi problem (Fig. 7.4) which consists in choosing $F(u) = |u - f|_2$.

- The BV-L1 problem (Fig. 7.5). It consists in choosing $F(u) = |u - f|_1$.
- The deconvolution problem (Fig. 7.6), which consists in choosing $F(u) = |h \star u - f|_2$.

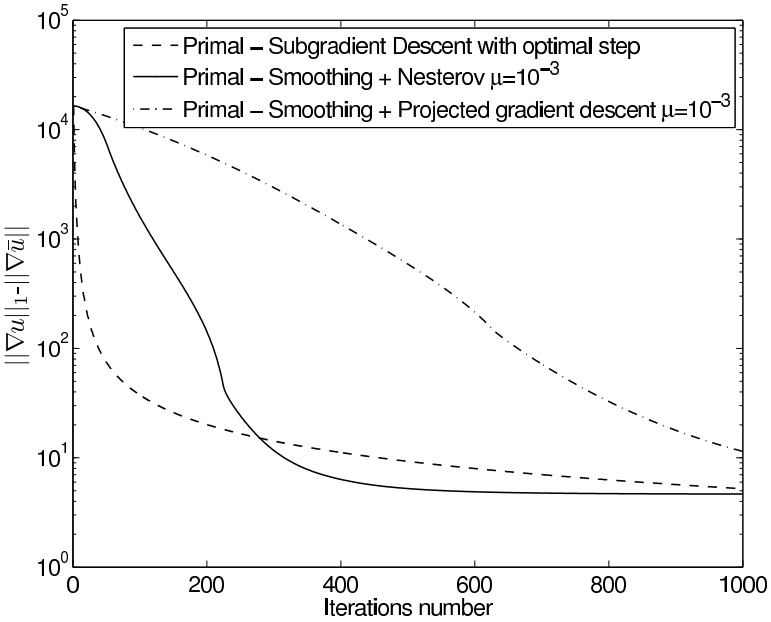

FIGURE 7.4. $||\nabla u^k||_1 - ||\nabla \bar{u}||_1$ *in log scale for ROF problem. Results on Figure (7.1).*

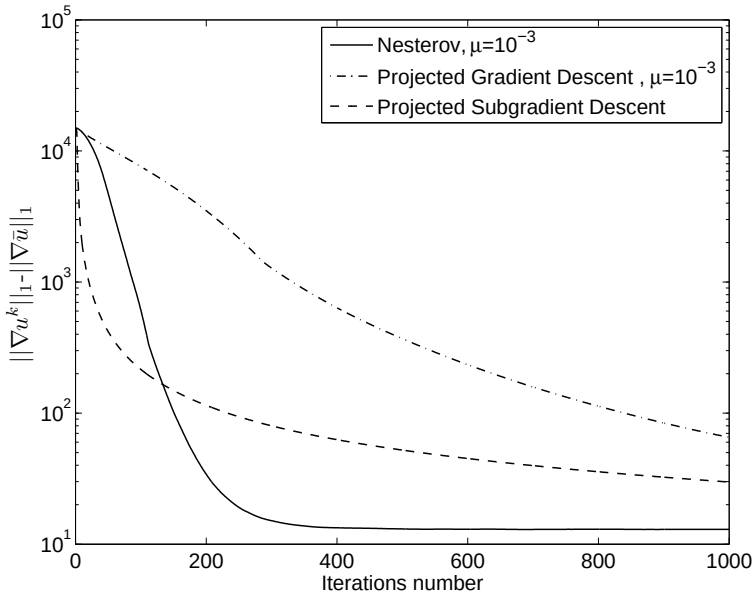

FIGURE 7.5. $||\nabla u^k||_1 - ||\nabla \bar{u}||_1$ *in log scale for BV-L1 problem. Results on Figure (5.6).*

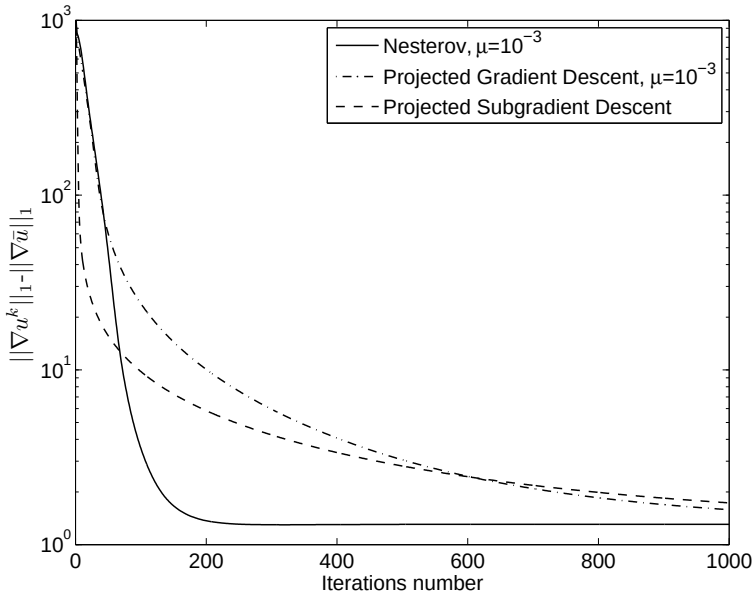

FIGURE 7.6. $||\nabla u^k||_1 - ||\nabla \bar{u}||_1$ *in log scale for deconvolution problem. Results on Figure (5.3).*

We notice that in all cases, Nesterov's scheme ($Algo3$) achieves much better than the projected gradient descent ($Algo4$). The projected subgradient descent with optimal step ($Algo5$) decreases the cost function very fast in the first iterations. Asymptotically, its rate of convergence seems to be lower than the proposed "Nesterov + smoothing" approach. Our conclusion is that the projected subgradient descent with precomputed sequences $\{t^k\}$ might be of interest to get approximate solutions in just a few iterations. To get accurate solutions in slightly higher computing times, the proposed approach "Nesterov + smoothing" approach is very appropriate.

7.3. Discussion.

7.3.1. Stability of the Nesterov scheme. In private correspondences, people having tested the Nesterov technique reported that it is unstable. We have found on the contrary, that in all tested cases, the Nesterov algorithm was stable and much faster than the projected gradient descent. The algorithm we apply [37] dates from 2005. The first optimal scheme was proposed in 1983 [35] and it seems that it is the one most people test. Our experiences confirm that the new proposed schemes are really efficient in practice as is shown by Y. Nesterov in his papers. Let us finally mention that the Nesterov's schemes were recently generalized and accelerated for a broader class of problems in [38]. Y. Nesterov gives further numerical results on l^1 type problems.

7.3.2. 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 [25] which was shown recently to be equivalent to quasi-Newton techniques [42]. 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.

Second order cone programming was proposed recently [52] and leads to very precise solutions, but the computing times seem to be very high. It is definitely a good choice to get 'exact' solutions.

A very promising approach based on graph-cuts was proposed recently [20, 12]. 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. That kind of approach continues being improved [19] and is clearly faster than our approach for the $BV - L^1$ problem. However, the approach presented in this paper allows to solve a much larger class of problems with a good efficiency and simpler implementations.

We think that very precise solutions are not needed in general in image processing. The visual system is unable to detect small perturbations [5]. Our method is thus very competitive with previously proposed schemes. It leads to precise enough solutions in short times and the algorithms are very easy to implement.

8. 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 [37] in convex optimization. Their efficiency is comparable or better than state of the art methods.

In this paper we focused on total variation problems. It is straightforward to replace the operators ∇ and $-div$ of the paper by other linear transforms B and B^*. This would allow to solve efficiently many other interesting problems like sparse reconstructions.

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

9. Appendix.

9.1. 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 [11]. For $u \in X$

$$(\nabla u)(i, j) = ((\partial_1 u)(i, j), (\partial_2 u)(i, j)). \tag{9.1}$$

∇u is an element of Y.

$$(\partial_1 u)(i, j) = \begin{cases} u(i+1, j) - u(i, j) & \text{if } i < n_x \\ 0 & \text{if } i = n_x \end{cases} \tag{9.2}$$

$$(\partial_2 u)(i, j) = \begin{cases} u(i, j+1) - u(i, j) & \text{if } j < n_y \\ 0 & \text{if } j = n_y \end{cases} \tag{9.3}$$

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

$$\langle \nabla u, p \rangle_Y = -\langle u, \operatorname{div}(p) \rangle_X. \tag{9.4}$$

[5] a uniform noise of amplitude 5 is almost invisible for images of amplitude 256

Simple computation gives

$$\begin{aligned}(\mathrm{div}\,(p))(i,j) &= \begin{cases} p^1(i,j)-p^1(i-1,j) & \text{if } 1<i<n_x \\ p^1(i,j) & \text{if } i=1 \\ -p^1(i-1,j) & \text{if } i=n_x \end{cases} \\ &+ \begin{cases} p^2(i,j)-p^2(i,j-1) & \text{if } 1<j<n_y \\ p^2(i,j) & \text{if } j=1 \\ -p^2(i,j-1) & \text{if } j=n_y \end{cases}\end{aligned} \tag{9.5}$$

Note that the operator div is surjective from Y to $X-\{(\gamma,\gamma,...,\gamma),\gamma\in\mathbb{R}\}$. Moreover it can be shown [11] that $||\mathrm{div}\,||_2\leq 2\sqrt{2}$.

9.2. 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 $K=\{y\in X,|\lambda(y-f)|_p\leq\alpha\}$, where λ is a diagonal matrix whose elements λ_i belong to $[0,\infty]$. The problem of projection on K can be written analytically

$$\Pi_K(x)=\underset{y\in K}{\arg\min}\left(|y-x|_2^2\right) \tag{9.6}$$

Let $\bar{y}$ denote the solution of (9.6). 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 elements $\lambda_i\in]0,\infty[$.

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

$$\bar{y}_i=\begin{cases} x_i & \text{if } |\lambda_i(f_i-x_i)|\leq\alpha \\ f_i+\frac{x_i-f_i}{|x_i-f_i|}\frac{\alpha}{\lambda_i} & \text{otherwise}\end{cases} \tag{9.7}$$

9.2.2. Projections on weighted l^1-balls. Up to a change of variable, the projection on a weighted l^1-ball writes

$$\Pi_K(x)=\underset{u,|\lambda u|_1\leq\alpha}{\arg\min}\left(|u-x|_2^2\right) \tag{9.8}$$

with $\lambda_i\in]0,\infty[$ and $\alpha>0$.

- First notice that if $|\lambda x|_1\leq\alpha$, 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 (9.8) is given by the solution of the Lagrangian problem

$$\Pi_K(x)=\underset{u\in\mathbb{R}^n}{\arg\min}\left(|u-x|_2^2+\sigma|\lambda u|_1\right) \tag{9.9}$$

The solution of this problem is in closed form

$$u(\sigma)_i=\begin{cases} x_i-sgn(x_i)\frac{\sigma\lambda_i}{2} & \text{if } |x_i|\geq\frac{\sigma\lambda_i}{2} \\ 0 & \text{otherwise}\end{cases} \tag{9.10}$$

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. $\Psi(\bar{\sigma})=\alpha$.

$$\Psi(\sigma)=\sum_{i=1}^{n}|\lambda_i\bar{u}_i| \tag{9.11}$$

$$=\sum_{i,|x_i|\geq\sigma\lambda_i/2}\lambda_i(|x_i|-\sigma\lambda_i/2) \tag{9.12}$$

$$=\sum_{i,y_i\geq\sigma}\lambda_i|x_i|-\sigma\lambda_i^2/2 \tag{9.13}$$

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 $\bar{\sigma}=\frac{(a_2-a_1)\alpha+b_2a_1-b_1a_2}{b_2-b_1}$. [O(1) operations]
6. Set $\bar{u}=u(\bar{\sigma})$ using (9.10). [O(n) operations]

9.2.3. Projections on weighted l^2-balls. The projection on a weighted l^2-ball (an ellipsoid) writes

$$\Pi_K(x)=\underset{\{y,|\lambda y|_2^2\leq\alpha\}}{\arg\min}|y-x|_2^2 \tag{9.14}$$

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 $|\lambda x|_2^2\leq\alpha$.
- Otherwise it can be shown using Lagrange multipliers that the solutions of (9.14) writes

$$\bar{y}_i=\frac{x}{\bar{\sigma}|\lambda_i|^2+1} \tag{9.15}$$

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

1. Set $k=0$, $\sigma^k=0$.
2. Compute $\alpha^k=\Psi(\sigma^k)$.
3. Compute $\beta^k=\Psi'(\sigma^k)=-2\sum_{i=1}^{n}\frac{\lambda_i^4x_i^2}{(\sigma^k\lambda_i^2+1)^3}$.

4. Set $\sigma^{k+1} = \sigma^k + \frac{(\alpha-\alpha^k)}{\beta^k}$.
5. Set $k = k+1$, go back to 2 until $|\alpha^k - \alpha| \leq \epsilon$.
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, and the ellipsoids are degenerate in that case. The average number of iterations is 6. The projection on a weighted l^2-balls is thus very fast.

9.3. Proof of proposition (4.1). *Proof.* Let $\bar{J}_\mu$ denote the solution of (5.10) and $\bar{J}$ denote the solution of (1.1). Using (5.5), we easily get that $|\bar{J} - \bar{J}_\mu| \leq \frac{n\mu}{2}$. The projected gradient is an $O(\frac{LD}{k})$ algorithm, where L is the Lipschitz constant of the gradient of the function to be minimized and D is the squared euclidean distance from the initial point to the solution (see for instance [45]). Thus algorithm (4.3) ensures that $|\bar{J}_\mu(u^k) - \bar{J}_\mu| \leq \frac{C}{k\mu}$. Where C is some constant independent of μ and k. Combining those two inequalities, we get that $|J(u^k) - \bar{J}| \leq \frac{C}{k\mu} + \frac{n\mu}{2}$. To get an ϵ-solution, it is thus sufficient to have $\frac{C}{k\mu} + \frac{n\mu}{2} < \epsilon$. It is the case if $k = \lfloor \frac{C}{\mu(\epsilon - \frac{n\mu}{2})} \rfloor + 1$. Maximizing the denominator in this expression, we get that the optimal pair (μ, k) is $\mu = \frac{\epsilon}{n}$ and $k = \lfloor \frac{Cn}{2\epsilon^2} + 1 \rfloor$. □

9.4. Proof of proposition (6.2). *Proof.* A direct consequence of equality (6.5) is

$$\inf_{q \in K} \left(F^*(-\operatorname{div}(q))\right) = -\inf_{u \in X} \left(||\nabla u||_1 + F(u)\right). \tag{9.16}$$

Let us introduce the notations. We set $\alpha^i = \frac{i+1}{2}$, $A^k = \sum_{i=0}^k \alpha^i = \frac{(k+1)(k+2)}{4}$. We denote $\phi(q) = F^*(-\operatorname{div}(q))$ and $\phi_*(u) = ||\nabla u||_1 + F(u)$. Equation (9.16) reduces to $\phi(\bar{q}) = -\phi_*(\bar{u})$.
Let us introduce the function

$$\Psi^k(x) = \frac{L}{2}||x - x^0||_2^2 + \sum_{i=0}^{k} \alpha^i(\phi(x^i) + \langle \nabla\phi(x^i), x - x^i\rangle_Y). \tag{9.17}$$

where L is the Lipschitz constant of ϕ. We proved in (6.1) that $L \leq \frac{||\operatorname{div}||_2^2}{\sigma}$. Using equation (5.8), we get that algorithm (3) ensures

$$A^k\phi(y^k) \leq \inf_{x \in K} \left(\Psi^k(x)\right) \tag{9.18}$$

We have $\phi(x^k) = F^*(-\operatorname{div}(x^k)) = \sup_{u \in X} \left(\langle u, -\operatorname{div}(x^k)\rangle_X - F(u)\right)$. Let u^k denote the solution of that problem. It is unique as F is strongly convex. As F is differentiable, u^k satisfies: $-\operatorname{div}(x^k) - \nabla F(u^k) = 0$. So that: $\phi(x^k) = \langle u^k, \nabla F(u^k) > -F(u^k)$. Moreover as F^* is defined as a supremum, its derivative is given by $\nabla F^*(-\operatorname{div}(x^k)) = u^k$. So that

$$\begin{aligned}
\phi(x^k + h) - \phi(x^k) &= \langle \nabla F^*(-\operatorname{div}(x^k)), -\operatorname{div}(h)\rangle_X + o(||h||_2) && (9.19)\\
&= \langle \nabla(\nabla F^*(-\operatorname{div}(x^k))), h\rangle_Y + o(||h||_2) && (9.20)\\
&= \langle \nabla u^k, h\rangle_Y + o(||h||_2) && (9.21)
\end{aligned}$$

Thus we get $\nabla\phi(x^k) = \nabla u^k$. Replacing $\phi(x^i)$ and $\nabla\phi(x^i)$ by their expressions in terms of u^i, we get

$$\begin{aligned}
&\phi(x^i) + \langle \nabla\phi(x^i), x - x^i\rangle_Y && (9.22)\\
&= \langle u^i, \nabla F(u^i)\rangle_X - F(u^i) + \langle u^i, \mathrm{div}\,(x^i)\rangle_X + \langle \nabla u^i, x\rangle_Y && (9.23)\\
&= -F(u^i) + \langle \nabla u^i, x\rangle_Y && (9.24)
\end{aligned}$$

Let us denote: $\bar{u}^k = \sum_{i=0}^k \frac{\alpha^i}{A^k} u^i$. As F is convex, $F(\bar{u}^k) \leq \sum_{i=1}^k \frac{\alpha^i}{A^k} F(u^i)$. So that

$$\begin{aligned}
&\inf_{x\in K}\left(\Psi^k(x)\right) && (9.25)\\
&= \inf_{x\in K}\left(\frac{L}{2}||x - x^0||_2^2 - A^k \sum_{i=1}^k \frac{\alpha^i}{A^k} F(u^i) + A^k\langle \nabla\bar{u}^k, x\rangle_Y\right) && (9.26)\\
&\leq -A^k F(\bar{u}^k) + \inf_{x\in K}\left(\frac{L}{2}||x - x^0||_2^2 + A^k\langle \nabla\bar{u}^k, x>\right) && (9.27)\\
&\leq -A^k(F(\bar{u}^k) + ||\nabla u^k||_1) + \frac{L}{2}||\frac{\nabla\bar{u}^k}{|\nabla\bar{u}^k|} - x^0||_2^2 && (9.28)\\
&\leq -A^k\phi_*(\bar{u}^k) + \frac{L}{2}||\frac{\nabla\bar{u}^k}{|\nabla\bar{u}^k|} - x^0||_2^2. && (9.29)
\end{aligned}$$

At line (9.28), $\bar{q}^k = \frac{\nabla\bar{u}^k}{|\nabla\bar{u}^k|}$ is defined only on the set $U = \{p, (|\nabla\bar{u}^k|)_p \neq 0\}$. On the complement of U a possible choice is to set $\bar{q}_p^k = x_p^0$. Now, taking $x^0 = 0$, we have $\sup_{x\in K}\left(||x^0 - x||_2^2\right) = n$.
So that finally $\inf_{x\in K}\left(\Psi^k(x)\right) \leq -A^k\phi_*(\bar{u}^k) + \frac{Ln}{2}$. Using (9.18), we get

$$\phi_*(\bar{u}^k) \leq -\phi(y^k) + \frac{Ln}{2A^k}. \tag{9.30}$$

As $\phi(\bar{q}) = -\phi_*(\bar{u})$, we have

$$\phi_*(\bar{u}^k) - \phi_*(\bar{u}) \leq -\phi(y^k) + \phi(\bar{q}) + \frac{Ln}{2A^k}. \tag{9.31}$$

Furthermore, as $y^k \in K$, $-\phi(y^k) + \phi(\bar{y}) \leq 0$. Finally, replacing every expression by their majoration, we get the first result

$$\phi_*(\bar{u}^k) - \phi_*(\bar{u}) \leq \frac{4||\mathrm{div}\,||_2^2 n}{\sigma(k+1)(k+2)}. \tag{9.32}$$

Then, we can remark that ϕ_* is strongly convex. Thus it satisfies (see [45]): $\phi_*(u+h) \geq \phi_*(u) + \langle\eta, h\rangle_X + \frac{\sigma}{2}|h|_2^2$ for any $\eta \in \partial\phi_*(x)$ and any $h \in X$. In particular $\phi_*(u) - \phi_*(\bar{u}) \geq \frac{\sigma}{2}|u - \bar{u}|_2^2$. Using (9.32) it is thus straightforward to get

$$|\bar{u}^k - \bar{u}|_2 \leq \frac{2\sqrt{2}||\mathrm{div}\,||_2\sqrt{n}}{\sigma k}. \tag{9.33}$$

□

[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] A. Almansa, V. Caselles, and G. Haro. Total variation regularized image restoration: the case of perturbed sampling. *SIAM Multiscale Modeling and Simulation*, 5, 2006.

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

[5] J.F. 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.

[6] 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.

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

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

[9] 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.

[10] M. Carlavan, P. Weiss, L. Blanc-Féraud, and J. Zerubia. Very efficient first order schemes for solving inverse problems in image restoration. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, submitted 2009.

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

[12] A. Chambolle. Total variation minimization and a class of binary MRF models. *EMMCVPR*, 578:136–152, 2005.

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

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

[15] T.F. Chan and J. Shen. Image processing and analysis - variational, pde, wavelet, and stochastic methods. *SIAM Publisher*, 2005.

[16] 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.

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

[18] P. L. Combettes and J. C. Pesquet. Image restoration subject to a total variation constraint. *IEEE Transactions on Image Processing*, 13(9):1295–1304, 2004.

[19] J. Darbon. Global Optimization for first order Markov random fields with submodular priors. *UCLA-CAM Report*, (08-61), September 2008.

[20] 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.

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

[22] S. Durand and M. Nikolova. Denoising of frame coefficients using L1 data-fidelity term and edge-preserving regularization. *SIAM Journal MMS*, 6:547–576, 2007.

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

[24] H. Fu, M. 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.

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

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

[27] 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.

[28] 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.

[29] 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.
[30] Y. Li and F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Transactions on Image Processing*, 5, 1996.
[31] S. Lintner and F. Malgouyres. Solving a variational image restoration model which involves L^∞ constraints. *Inverse Problems*, 20:815–831, 2004.
[32] 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.
[33] Y. Meyer. Oscillating patterns in image processing and in some nonlinear evolution equations. *The Fifteenth Dean Jaqueline B. Lewis Memorial Lectures*, 2001.
[34] A.S. Nemirovskii and D.B. Yudin. Problem Complexity and Method Efficiency in Optimization. *Wiley, New-York*, 1983.
[35] Y. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(\epsilon^{-2})$. *Doklady AN SSSR*, 269(3):543–547, 1983.
[36] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.
[37] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematic Programming, Ser. A*, 103:127–152, 2005.
[38] Y. Nesterov. Gradient methods for minimizing composite objective function. *CORE discussion paper*, 2007.
[39] Michael K. NG, Raymond H. Chan, and Wun-Cheung Tang. A Fast Algorithm for Deblurring Models With Neumann Boundary Conditions. *SIAM J. Sci. Comput.*, 21(3):851–866, 2000.
[40] M. Nikolova. Local strong homogeneity of a regularized estimator. *SIAM Journal on Applied Mathematics*, 61(2):633–658, 2001.
[41] M. Nikolova. A variational approach to remove outliers and impulse noise. *Math. Imag. Vis.*, 20:99–120, 2004.
[42] M. Nikolova and R. Chan. The equivalence of half-quadratic minimization and the gradient linearization iteration. *IEEE Trans. on Image Processing*, 16:1623–1627, June 2007.
[43] S.J. Osher, A. Sole, and L.A. Vese. Image Decomposition and Restoration using Total Variation Minimization and the H^{-1} Norm. *Multiscale Modeling and Simulation : SIAM*, pages 349–370, 2003.
[44] S.J. Osher and L.A. Vese. Modeling Textures with Total Variation Minimization and Oscillating Patterns. *J. Sci. Comput.*, pages 553–572, 2003.
[45] B.T. Polyak. Introduction to Optimization. *Translation Series in Mathematics and Engineering, Optimization Software*, 1987.
[46] L. Rudin and S. Osher. Total variation based image restoration with free local constraints. *Proc. IEEE ICIP*, 1:31–35, 1994.
[47] L. Rudin, S. Osher, and E. Fatemi. Nonlinear Total Variation Based Noise Removal. *Physica D*, 60:259–268, 1992.
[48] 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.
[49] 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.
[50] C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.
[51] P. Weiss, G. Aubert, and L. Blanc-Féraud. Some Applications of l^∞- Constraints in Image Processing. *INRIA Research Report*, (6115), 2006.
[52] W. Yin, D. Goldfarb, and S. Osher. A Comparison of Total Variation Based Texture Extraction Models. *ournal of Visual Communication and Image Representation*, 18:240–252, 2007.