INSTITUTE OF PHYSICS PUBLISHING INVERSE PROBLEMS

Inverse Problems **20** (2004) 815–831 PII: S0266-5611(04)66318-0

Solving a variational image restoration model which involves L^∞ constraints

Stéphane Lintner[1] **and François Malgouyres**[2]

[1] ACM, MC 217-50, Caltech, Pasadena, CA 91125, USA
[2] LAGA/L2TI, Université Paris 13, 99 avenue Jean-Batiste Clement, 93430 Villetaneuse, France

E-mail: lintner@acm.caltech.edu and malgouy@math.univ-paris13.fr

Received 16 July 2003, in final form 22 January 2004
Published 26 March 2004
Online at stacks.iop.org/IP/20/815 (DOI: 10.1088/0266-5611/20/3/010)

Abstract

In this paper, we seek a solution to linear inverse problems arising in image restoration in terms of a recently posed optimization problem which combines total variation minimization and wavelet-thresholding ideas. The resulting nonlinear programming task is solved via a dual Uzawa method in its general form, leading to an efficient and general algorithm which allows for very good structure-preserving reconstructions. Along with a theoretical study of the algorithm, the paper details some aspects of the implementation, discusses the numerical convergence and eventually displays a few images obtained for some difficult restoration tasks.

1. Introduction

A broad variety of image restoration tasks can be expressed as inverse problems of the following form:

$$v = H(u) + b, \tag{1}$$

where $u \in \mathbb{R}^n$ is the original image to be recovered from $v \in \mathbb{R}^n$, H is a known linear map from $\mathbb{R}^n$ into $\mathbb{R}^n$, and $b \in \mathbb{R}^n$ is Gaussian noise. Common examples are denoising, deblurring, linear local contrast changes, image zooming[3] and local inpainting. Combinations of these are also of interest in practice: for instance, one might want to deblur an image while filling in small missing parts.

1.1. Combining total variation methods and wavelet-like thresholding

In the past decade, many restoration methods have been developed to deal with such problems, and among them are the two well-known approaches: total variation minimization, which was initiated in [20], and wavelet-like thresholding, as proposed originally in [7].[4] These

[3] A study of variational zooming can be found in [18].
[4] Detailed references can be found in the other papers cited below.

approaches have been opposed for quite a while, but current research is increasingly focusing on combinations of both. In fact, the merging of variational ideas and thresholding techniques, pioneered by Coifman *et al* (see [6, 24]), has led recently to various new and promising algorithms. To our knowledge, it is currently being developed along five directions: in [6] and [8], a set of wavelet coefficients is interpolated according to a total variation criterion. In [3, 11] the authors propose to determine the most meaningful wavelet coefficients, also via a total variation criterion. Yet another algorithm is proposed in [14, 24] where information is removed from the residual image $H(w) - v$ by using a wavelet analysis, thus allowing for the composition of wavelet techniques with any other method and, in particular, a variational one. Further work is also to be found in [15], where it is shown that the conditioning of the inverse of H can be improved using wavelet packets in the data fidelity term of a variational approach.

Eventually, there is the combined approach with which we will be concerned in this paper, and which takes the form of the following optimization problem:

$$(P_*) : \begin{cases} \min TV(w), \\ \text{over } \|H(w) - v\|_{\mathcal{D},\infty} \leqslant \tau, \end{cases}$$

where the fidelity term is given by the l^∞ norm of the scalar products with the elements of a dictionary of features $\mathcal{D}$, which contains m elements in $\mathbb{R}^n$:

$$\|u\|_{\mathcal{D},\infty} = \sup_{\Psi \in \mathcal{D}} |\langle u, \Psi \rangle|.$$

1.2. About (P^*)

The closest model to (P^*) was first designed for image decompression and was studied in [6, 23], while the definition of the constraints (which included the use of a dictionary) appeared in [13] as an argument for the noise selection. (P_*) by itself, only evoked in [22], was studied in [17]: there, an existence result in the continuous framework of $BV \cap L^2$ was given, the link with soft wavelet thresholding and the Rudin–Osher–Fatemi algorithm discussed and the penalty method studied to find a solution to (P_*). (Detailed proofs appeared in [16]). Finally, an attempt to solve (P_*) via an Uzawa algorithm was made in [2], wherein two conjectures on the asymptotic optimality of its solution were stated for a dictionary made of curvelets.

As a reminder, here are a few important points on (P^*):

(i) From a variational point of view, the model is mostly a modification of the Rudin–Osher–Fatemi algorithm: the traditional, rather uninformative L^2-norm is replaced by an L^∞ norm which carries far more information and reduces the 'search space' for the minimum of the TV norm [17]. A good choice of the dictionary $\mathcal{D}$ preserves much more structure and textures in the restored images, thus reducing the typical 'washout' effect of the total variation minimization.

(ii) From a wavelet point of view, this model takes into account as much information as the classical thresholding techniques: the restored image belongs to

$$N_{H,\mathcal{D},\tau} = \{u, \|H(u) - v\|_{\mathcal{D},\infty} \leqslant \tau\}.$$

The difference with (for instance) soft thresholding is that all coefficients preserve a degree of freedom up to the threshold; it is the total variation minimization which will fix the value inside this interval of certainty. As a result, the usual Gibbs phenomenon due to brutal discarding of coefficients is avoided, and the model leads to artefact-free images. [2, 8, 17].

(iii) The *a priori* freedom in the choice of $\mathcal{D}$ has great potential interest, since it allows the definition of optimal features Ψ to be recovered, thus performing a redundant analysis. (Research involving redundancy is quite active, see for instance [4, 10, 14, 19, 24].)

(iv) (P_*) might find other applications in image processing such as segmentation or registration. In fact, it could even be proposed for inverse problems outside image restoration for which the noise is well suited for thresholding (radar speckle, for example, is not), and for which some features Ψ can be defined to allow a good modelling of the data to be recovered.

Unfortunately, available algorithms for (P^*) work for a very restrained class of applications only, such as denoising (see [2, 17]), image decompression (see [2, 6, 23]) and some simple deblurring tasks (see [17]). In addition, these early algorithms also suffer from numerical weaknesses, such as being either very unstable or poorly convergent. Nevertheless, the images already obtained are of very high quality, and the goal of this paper is to develop an efficient and robust method for solving (P_*) in its general form.

1.3. A difficult optimization task

(P_*) is clearly not an easy problem to solve: it is high dimensional, the geometry of the constraints is not trivial, and, as a result, most available optimization techniques are not adaptable from a practical point of view; projections, for instance, are simply not computable. In fact, only sequential unconstrained minimization methods seem to provide reasonable approaches, and among them three families can be distinguished: penalty methods, dual methods of the Uzawa type and augmented Lagrangian methods.

The penalty method (see [17] for a complete presentation) is in essence quite straightforward, but it is known for being ill-conditioned as the penalty parameter increases. As a result, constraints are very quickly enforced, thus allowing very good structure preservation, while the total variation is hard to minimize thoroughly and requires many iterations, which is why the images can sometimes have a noisy aspect. Let us add here that the algorithm proposed in [17] was unable to cope with operators H which were not quasi-diagonal in the dictionary, and therefore seemed quite limited. We will see later that this was an unnecessary assumption.

The Uzawa method, on the other hand, is quite infamous for its poor capability to enforce the constraints accurately, which in a certain way, makes it the opposite of the penalty method: applied to (P_*), it should remove noise very efficiently, while textures might be harder to retain. The results in [2] already show that this 'poor' convergence is not a problem in practice, and it is a well-known fact that the Uzawa method provides a neat tool for problems in which numerical accuracy in the constraints is not necessarily needed (see, for instance, [9, 12]). Of great interest for (P_*) is also the fact noted in [2] that the Uzawa method takes full advantage of the linear constraints, thus allowing fast computations. As yet though, the algorithm derived in [2] only deals with trivial operators H, and suffers from a fundamental instability that leads to severe blow-ups.

Augmented Lagrangian methods seem to provide a natural compromise: they are, in a certain way, a mixture of penalty and dual methods, and as a result, they compensate the defects of both, and are known for being more efficient (which is why we mention them here). In practice though, their application can be quite cumbersome: they involve more parameters that need to be tuned, and in addition, constraints typically appear in a quadratic form (as in the penalty method), which means that they need to be evaluated at each step. In our case, this corresponds to transforms in $\mathcal{D}$ which for large images and dictionaries become extremely costly in time: we will thus leave the augmented Lagrangian algorithms aside, and focus on the simpler methods only.

1.4. About this paper

In this paper, we discuss the Uzawa method, and show how a stable and efficient algorithm can be derived which solves (P_*) in its general form. Of special interest is the fact that no particular requirement is made on the degradation operator H, which makes our algorithm quite straightforward to use: once the dictionary $\mathcal{D}$ has been fixed, only the threshold τ needs to be chosen. (Theoretically, τ only depends on the noise variance and the dictionary $\mathcal{D}$.) In particular, high-quality results are obtained without ever inverting or approximating H, thus allowing quite difficult restoration tasks to be dealt with.

The paper is organized as follows. Section 2 briefly reviews duality for convex programming problems, and explains how to adapt the Uzawa method to (P_*) in a robust way. Section 3 details some useful numerical aspects. In particular, formulae for the gradients are given for an arbitrary choice of the operator H and the dictionary $\mathcal{D}$, thus allowing general and fast computations[5]. Section 4 discusses the numerical convergence of our algorithm, comparing in particular the penalty method to the Uzawa method. Section 5 eventually illustrates the full range of applications of our method, by comparing its results with the classical Rudin–Osher–Fatemi algorithm: in the simple denoising case, but also for a difficult deblurring task, and a 'mixed' restoration problem.

2. Uzawa method and total variation

Let us review briefly the points of importance in duality theory for minimization problems of the form

$$(P): \begin{cases} \min J(u), \\ \text{over } \varphi_i(u) \leqslant 0, \qquad i = 1, \ldots, m, \end{cases}$$

where J is a strictly convex, C^1 functional over $\mathbb{R}^n$ and the functions φ_i are linear[6].

2.1. Classical Lagrangian duality

The generalized Lagrangian for (P) is defined by

$$L(u, \lambda) = J(u) + \sum_{i=1}^{m} \lambda_i \varphi_i(u).$$

Since J is strictly convex, $L(u, \lambda)$ is strictly convex in u, for every $\lambda = (\lambda_i) \in \mathbb{R}^m_+$ ($\mathbb{R}^m_+$ denotes the set of all the elements of $\mathbb{R}^m$ with positive coordinates). If we further assume that $L(u, \lambda)$ is coercive in u, then, for any $\lambda \in \mathbb{R}^m_+$, the generalized Lagrangian has a unique minimum $u_\lambda = \min_u L(u, \lambda)$, which depends continuously on λ. Theory shows that solving (P) is equivalent to finding saddle points of $L(u, \lambda)$, which in turn, by defining

$$G(\lambda) = L(u_\lambda, \lambda),$$

is equivalent to solving the dual problem

$$(Q): \max_{\lambda \in \mathbb{R}^m_+} G(\lambda).$$

As is well known, (Q) is a simpler problem since it consists in maximizing a concave functional over the positive half-space $\mathbb{R}^m_+$. Recall also that, if λ^* is a maximizer of G, then the corresponding u_{λ^*} is a solution to (P). The latter can computed by an *unconstrained* minimization of $L(u, \lambda^*)$, over $u \in \mathbb{R}^n$.

[5] This section partly applies to the penalty method, and improves the corresponding section of [17].

[6] For a rigorous discussion on duality, see [5, 9] or [12]. Also, the linearity of the constraints is not a theoretical necessity, but it is the key for fast computations in practice.

2.2. Ascent method of the Uzawa type

Since $L(u, \lambda)$ is strictly convex and coercive in u for all $\lambda \in \mathbb{R}_+^m$, G can also be shown to be differentiable, and very appropriately, its gradient is given by

$$\frac{\partial G}{\partial \lambda_i}(\lambda) = \varphi_i(u_\lambda).$$

Note that calculation of this gradient requires one unconstrained minimization of $L(u, \lambda)$ in u and one evaluation of the constraints at u_λ. In principle, maximization of G can therefore be done by applying any projected gradient-ascent method, and this is precisely what the Uzawa method is about. The point of importance is of course the following: while the generated sequence of dual variables $(\lambda^p)_{p\in\mathbb{N}}$ converges to λ^*, the sequence of primal variables u_{λ^p}, because of the continuous dependence, will converge to u^*, the unique solution to (P). (A precise description of the algorithm is provided in section 2.4.)

2.3. Application to (P_*)

Recall that we are interested in solving

$$\min TV(u),$$
$$|\langle H(u) - v, \Psi_i\rangle| \leqslant \tau, \qquad \Psi_i \in D.$$

To fit the duality statements presented above, we need linear constraints, C^1 smoothness and strict convexity of J, but also coercivity of the Lagrangian:

- Linear constraints are easy to obtain, by splitting $|\langle H(u) - v, \Psi_i\rangle| \leqslant \tau$ into $\phi_i^{\pm}(u) \leqslant 0$, with
$$\phi_i^{\pm}(u) = \pm\langle H(u) - v, \Psi_i\rangle - \tau.$$
- C^1 smoothness and strict convexity of $J(u)$ are also not an issue since in practice, we use a common smooth, strictly convex and coercive approximate of the total variation (see [1]):
$$TV_\beta(u) = \sum \varphi_\beta(|\nabla u|) + \beta(\langle u, \mathbb{1}\rangle)^2, \tag{2}$$
where $\beta > 0$ and for any $t \in \mathbb{R}$, $\varphi_\beta(t) = \sqrt{t^2 + \beta^2}$.
- The main problem is that $L(u, \lambda)$ may not be coercive. Indeed, since $TV_\beta(u)$ has a linear growth at infinity, the set Γ defined by
$$\Gamma = \left\{\lambda \in \mathbb{R}_+^m, L(u, \lambda) \text{ is not coercive in } u\right\}$$
is not empty. Obviously, for any λ in the interior of Γ, u_λ is not even defined, since $L(u, \lambda)$ is not bounded from below.

A very simple way to avoid any problem is given by the following remark: let $J(u) \geqslant 0$ for any $u \in \mathbb{R}^n$. Then *minimizing $J(u)$ over Ω, is the same as minimizing $J(u)^2$ over this same Ω.*

Moreover, the following property holds.

Proposition 1. *Let $J : \mathbb{R}^n \to \mathbb{R}$ be convex and coercive over $\mathbb{R}^n$ and, for any $M > 0$ and $i \in \{1, \ldots, M\}$, let $\varphi_i : \mathbb{R}^n \to \mathbb{R}$ be linear. Then, for any $\lambda \in \mathbb{R}^M$,*

$$L(u, \lambda) = J(u)^2 + \sum_{i=1}^{M} \lambda_i \varphi_i(u)$$

is coercive in u.

The proof of the proposition is given in the appendix.

As a result, we obtain the following minimization problem:

$$(P_{**}) : \begin{cases} \min TV_\beta(u)^2, \\ \text{over } \phi_i^\epsilon(u) \leqslant 0, \qquad i = 1, \ldots, m, \qquad \epsilon = \pm. \end{cases}$$

Clearly, the squaring preserves the smoothness and convexity properties of the functional, but it allows robust minimizations in u, even for small values of β: the Uzawa method can thus be applied to (P_{**}) in a straightforward manner. And of course, arguments similar to those given in [1] show that solutions to (P_{**}) converge to solutions to (P_*) as β goes to 0.

Remark. In [2], a small quadratic term of the form $\varepsilon\|u\|^2$ was added to the functional, so as to ensure the coercivity of the Lagrangian. Even though theoretically valid, this proves quite useless in practice since for small values of ε, the quadratic term effectively starts to compensate the linear part only when the norm $\|u\|^2$ becomes very large. As a result numerical estimates of u_λ tend to blow up, and the algorithm becomes highly unstable. One way around is of course to choose non-negligible values of ε in which case the quadratic term does stabilize the algorithm. For this to happen though, it is necessary to have the L^2 term of the same magnitude as the TV norm, but then, the results are not solutions to (P^*) any more: in fact, they differ quite significantly from the correct minimum (see section 4).

2.4. Uzawa algorithm for (P_{**})

The Lagrangian is now given by

$$L(u, \lambda) = TV^2(u) + \sum_{i=1}^{m} (\lambda_{i,+} - \lambda_{i,-}) \langle H(u), \Psi_i \rangle + C_\lambda, \tag{3}$$

where C_λ is a constant and is thus irrelevant when minimizing L, and the classical Uzawa algorithm runs as follows:

(i) Make an initial choice for λ^0.
(ii) For λ^p, compute $u_p = \arg\min L(u, \lambda^p)$ by a steepest gradient descent.
(iii) Make a gradient ascent of the dual functions by the following update: $\lambda_{i,\pm}^{p+1} = \left(\lambda_{i,\pm}^p + \rho_p . \phi_i^\pm(u_p)\right)_+$, where $(t)_+ = \sup(t, 0)$ is the projection onto the positive half-space, and ρ_p is the step size.
(iv) Stop if the constraints are satisfied, or return to step 2.

Since the algorithm simply consists in a projected gradient ascent, convergence can be achieved for various choices of steps: constant steps, diverging-sum rule ($\rho_p = \rho_0 \frac{m}{m+p}$ for instance) and of course more complicated adaptive methods. Unfortunately, simple rigid schemes depend heavily on the choice of ρ_0, and we observed in practice that large values of ρ_0 could lead to serious blow-offs if not corrected after a few iterations. Adaptive steps on the other hand are extremely costly, and a steepest gradient ascent for instance is out of question here, which is why we chose the following compromise: to avoid the (only) instability observed with rigid schemes, we simply take a constant step ρ_0 and verify that the dual function $G(\lambda)$ doest not decrease significantly; if it does, the step is divided by two and the iteration repeated.

3. Numerical aspects

3.1. Fast unconstrained minimization of $L(u, \lambda)$

To minimize $L(u, \lambda)$ over $\mathbb{R}^n$, we use a steepest gradient descent, and decompose $L(u, \lambda)$ into two parts:

$$L(u, \lambda) = J^2(u) + F_\lambda(u).$$

Here, $J(u) = TV_\beta(u)$, and basic algebra gives the following expression:

$$F_\lambda(u) = \sum_{i=1}^{m} (\lambda_{i,+} - \lambda_{i,-}) \langle H(u), \Psi_i \rangle = \langle u, A_\lambda \rangle,$$

with

$$A_\lambda = H^* \left(\sum_i (\lambda_{i,+} - \lambda_{i,-}) \Psi_i \right),$$

H^* being the adjoint of H.

The gradient of $F_\lambda(u)$ in u is then simply given by

$$\nabla_u L(u, \lambda) = 2J(u) \nabla J(u) + A_\lambda.$$

Since A_λ can be precomputed, the gradient descent of $L(u, \lambda)$ becomes straightforward.

Important remarks.

(i) A_λ is easy to compute if $\mathcal{D}$ allows a fast evaluation of the following map:

$$T : (\alpha_i) \in \mathbb{R}^m \to \sum_i \alpha_i \Psi_i \in \mathbb{R}^n.$$

On the other hand, the update of λ (see (iii) in section 2.4), also requires the evaluation of all the coefficients $\langle H(u) - v, \Psi_i \rangle$. Hence, the only requirement on $\mathcal{D}$ for our algorithm to work efficiently is that both a decomposition into scalar products $\langle w, \Psi_i \rangle$, and a reconstruction mapping T be available. Note that T is just the inverse of $w \to (\langle w, \Psi_i \rangle)_i$ if $(\Psi_i)_i$ is an orthonormal basis. If $\mathcal{D}$ contains a collection of orthonormal bases, T is the sum of all the reconstruction operators.

(ii) The above derivation works for *any* linear operator which can be quickly evaluated along with its adjoint. This is a rather weak requirement: for a blurring kernel for instance, the adjoint is given by the symmetric kernel; an inpainting mask is self-adjoint, etc. Note also that no inversion is ever needed. We will see that having a computable pseudo-inverse makes the initialization of the Uzawa method more handy, but it is by no means a necessity.

(iii) Also, there is no *numerical* need to adapt $\mathcal{D}$ to H: the algorithm will converge and yield a result in any case, and this is a major improvement over existing wavelet-based deblurring algorithms which require quasi-diagonal approximations. Of course, end results will be better if the dictionary is well adapted: after all, the goal is to minimize $\mathrm{E}(\|b\|_{\mathcal{D},\infty})$! Note by the way that the constraints can be rewritten as $|\langle u, H^*(\Psi_i) \rangle - \langle v, \Psi_i \rangle| \leqslant \tau$, which shows that $\mathcal{D}$ should be designed so as to carry information along the thresholding features $H^*(\Psi)$.

Our examples will show that good results can be obtained in practice without any deep analysis: even though some constraints might be weak, the redundancy of $\mathcal{D}$ usually compensates for them by also including other, stronger constraints.

(iv) Eventually, let us add that the above derivation also holds for more general fidelity terms, such as the quadratic penalty term proposed in [17]. If φ is a smooth function, we can consider fidelity terms of the form $F_\lambda(u) = \sum_i \lambda_i \varphi(\langle H(u) - v, \Psi_i \rangle)$, and show that the resulting gradient in u is given by

$$\nabla_u F_\lambda(u) = H^* \left(\sum_i \lambda_i \varphi'(\langle H(u) - v, \Psi_i \rangle) \Psi_i \right).$$

This makes the diagonal approximation needed in [17] unnecessary.

3.2. Initial choice of the dual variable

As with any gradient-ascent method, a good choice of the initial point (here the multiplier λ^0) is of great importance, since it can spare many iterations. We generalize an idea proposed originally in [2]: suppose a good approximation u_0 to the restoration task is available (through traditional thresholding, or Wiener filters, or the Rudin–Osher–Fatemi algorithm etc), then one can define λ^0 to be a multiplier such that u_0 is the minimum of $L(u, \lambda^0)$. Denoting now TV_β^2 by J, this is equivalent to writing

$$\nabla J(u_0) + H^* \left(\sum_i \left(\lambda^0_{i,+} - \lambda^0_{i,-}\right) \Psi_i \right) = 0,$$

and any solution to this equation is thus a natural candidate for λ^0.

When $\mathcal{D}$ contains only one orthonormal basis. Assume for now that $\mathcal{D}$ contains a single orthonormal basis $\mathcal{D} = \{\Psi_i, i = 1, \ldots, n\}$. H might not be invertible, but if we assume that a pseudo-inverse $\tilde{H}$ is available such that $H^* \circ \tilde{H} \approx Id$, then a reasonable choice is given by

$$\lambda^0_{i,+} - \lambda^0_{i,-} = -\langle \tilde{H}(\nabla J(u_0)), \Psi_i \rangle.$$

Since the basis is orthonormal, we verify that

$$H^* \left(\sum_i \left(\lambda^0_{i,+} - \lambda^0_{i,-}\right) \Psi_i \right) = -H^* \circ \tilde{H}(\nabla J(u_0)) \approx -\nabla J(u_0).$$

This, by itself, does not yet determine $\lambda^0_{i,+}$ and $\lambda^0_{i,-}$. But, recall that the dual problem consists in maximizing the dual function $G(\lambda)$. The optimal choice for λ^0 is the one that minimizes $\lambda^0_{i,+} + \lambda^0_{i,-}$ (see (3)) and can easily be shown to be

$$\lambda^0_{i,+} = \max(0, -\langle \tilde{H}(\nabla J(u_0)), \Psi_i \rangle), \qquad \lambda^0_{i,-} = \max(0, \langle \tilde{H}(\nabla J(u_0)), \Psi_i \rangle).$$

If H is invertible, this choice for λ^0 corresponds exactly to the chosen u_0. It also works well in practice if H is not invertible.

Collection of orthonormal bases. Assume now that $\mathcal{D}$ is a collection of functions that can be organized in p orthogonal bases, which we write as $B_q = \left\{\Psi_1^q \ldots \Psi_n^q\right\}$: typical would-be wavelet packets. Then obviously, there is more than one multiplier λ that satisfies $u_\lambda = u_0$: each basis extracted from $\cup_q B_q$ gives one possible λ^0.

Note that searching for the λ that maximizes G over all λ satisfying $u_\lambda = u_0$ is a linear programming task whose dimensionality is way too large to allow any attempt at solving it.

We simply propose to compute, as above, one λ^0 per basis B_q, and then take the average over all these λ^0.

General H and $\mathcal{D}$. When $\mathcal{D}$ is not a collection of orthonormal bases, or if no pseudo-inverse of H is available, we use a completely different strategy. Let u_α be a minimum of the penalty functional

$$J(u) + \alpha \sum_i \varphi(\langle H(u) - v, \Psi_i \rangle),$$

where $\alpha > 0$ and $\varphi(t) = \sup(|t| - \tau, 0)^2$, for $t \in \mathbb{R}$. Then we know that

$$\nabla J(u_\alpha) + H^* \left(\sum_i \alpha \varphi' \left(\langle H(u_\alpha) - v, \Psi_i \rangle \right) \Psi_i \right) = 0.$$

Therefore,

$$\lambda^0_{i,+} - \lambda^0_{i,-} = \alpha \varphi' \left(\langle H(u_\alpha) - v, \Psi_i \rangle \right)$$

is such that u_α minimizes $L(u, \lambda^0)$. Moreover, if all the Ψ_i which correspond to active constraints of the true optimum are independent, it can be shown that the λ^0 defined just above tends to λ^* as α goes to infinity (see [12]). In practice, a few iterations of the penalty method thus provide a good initialization of the dual variable, even for dictionaries which are not orthonormal at all. Clearly though, this initialization requires more computations than that presented in the previous paragraph.

4. Numerical convergence

To discuss the convergence of our algorithm, we run a simple denoising experiment, in which the original image is degraded with a Gaussian noise of standard deviation $\sigma = 20$. We then solve (P_*) with a threshold $\tau = 70$, and a large dictionary $\mathcal{D}$ containing 16 bases: four fully decomposed packet trees of depths one to four, and four of their translated (shifted) versions.

The result after five iterations can be seen in figure 1 (top right), and is quite satisfactory: most of the structures are preserved, while all the noise has been removed. If we let the algorithm iterate, no notable change is introduced. For comparison purposes we also run the penalty method over the same number of iterations, with a large penalty parameter $\alpha = 1000$, and the resulting image is given at bottom left, while bottom right shows the image obtained with the 'non-squared' algorithm from [2], in the case where it converges ($\varepsilon = 0.01$, see the next paragraph).

As can be seen, the penalty method has trouble removing the noise (i.e. minimizing the total variation) efficiently, and this is confirmed by the curves in figure 2. The Uzawa method, on the other hand, is clearly not enforcing the constraints as thoroughly (see figure 2 and table 1): the maximum of the constraints decreases extremely slowly towards the threshold τ. Still, it should be added that, after a few iterations, less than 1% of the constraints remain above τ.

A very interesting experiment then consists of feeding the Uzawa image as the input for the penalty method. Since the latter acts mostly like a projection (see figure 2), a few iterations of it suffice in fact to force the remaining 1% unsatisfied constraints below the threshold. The result is visually almost identical to the first Uzawa output: the difference is only visible when rescaling the residual, which as figure 3 shows, is mostly made of lightly contrasted textures. (To give an idea, the residual has an l^∞ norm of 6 on a 255 grey scale.)

The algorithm thus produces quite satisfactory solutions to (P_*): obviously, no decimal accuracy is to be expected, but the results are of great visual quality, and as we have just

Figure 1. Top left: original noisy picture ($\sigma = 20$). Top right: result after five iterations of the Uzawa method; numerical enforcement of the constraints is not achieved. Bottom left: result of the penalty method. Bottom right: result with the non-squared Uzawa algorithm ($\varepsilon = 0.01$).

Table 1. Statistics of the end images for each method. Max: maximum of the constraints. Mean: mean of the active constraints only. Proportion: proportion of active constraints.

	Total variation	Max	Mean	Proportion (%)
Penalty method	11.01	70.030	70.005	0.010
Uzawa method	8.578	72.413	70.271	0.164
Uzawa + penalty	9.003	70.026	70.004	0.009

demonstrated, not so far from the exact minimum. Should one be really concerned with very precise convergence, the combined 'projection' proposed above provides a thoroughly minimized and constrained image. Note that in theory, augmented Lagrangians were designed to do exactly this: combine the penalty with the Uzawa method in one efficient algorithm, which, applied here, might lead to even more accurate results. (At very high computational costs though, and with no noticeable visual improvement.)

Remark. As mentioned earlier, the squaring of the total variation is decisive in making the algorithm convergent. It is true that without the squaring, the Uzawa method can still lead

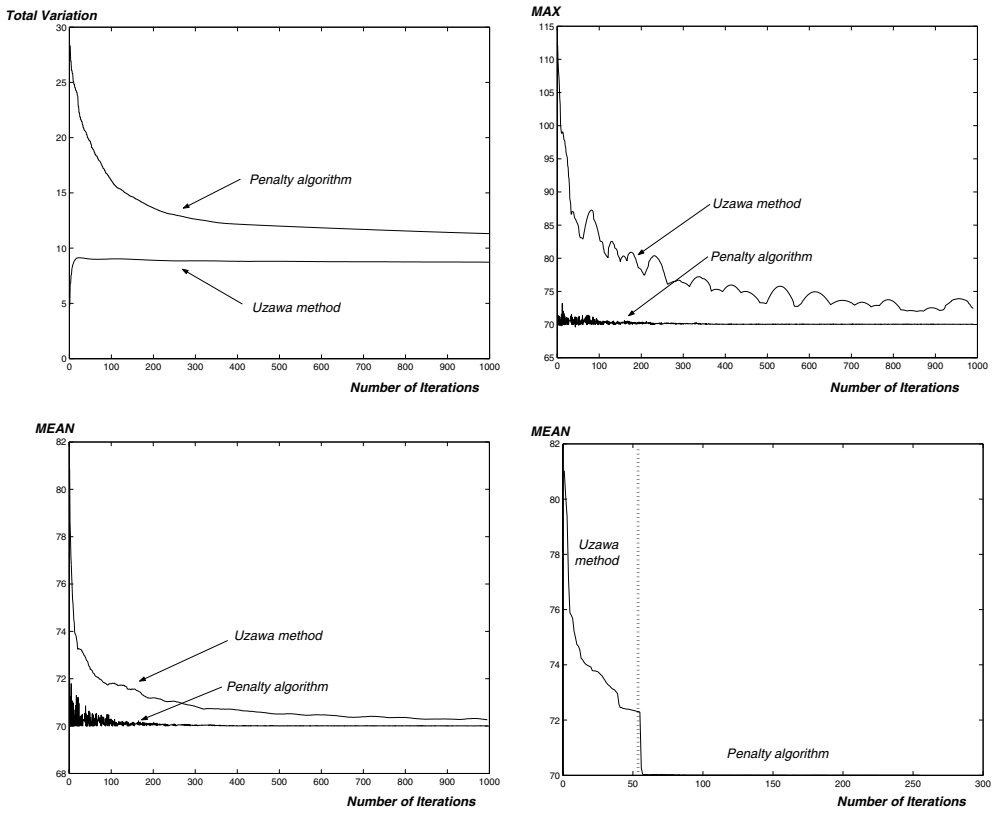

Figure 2. Iterating the penalty and the Uzawa method. Top left: total variation minimization. Top right: maximum of the constraints. Bottom left: mean of the active constraints (they usually represent less than 1% of all constraints). Bottom right: the penalty method acts like a projection. (The time scale being in number of iterations, it should be remembered that one iteration of the Uzawa algorithm requires more computations than one iteration of the penalty method.)

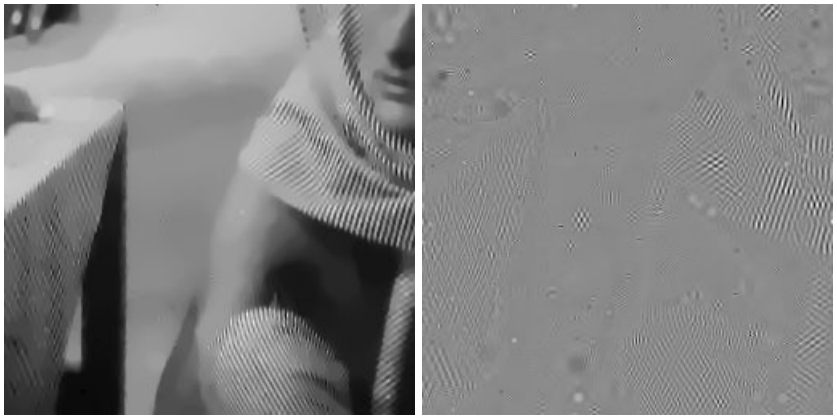

Figure 3. Left: Uzawa method followed by a penalty projection. Right: rescaled difference between the plain Uzawa method, and the projected Uzawa–penalty result.

Figure 4. Left: denoising with the Rudin–Osher–Fatemi algorithm. Right: solution to (P^*) with a complete dictionary of wavelet packets.

to good images (see [2]), provided though that it is very well initialized, and that only a few iterations are carried out: in most cases though, if this 'non-squared' algorithm is allowed to iterate long enough (10, 20 or more iterations), it is almost guaranteed to blow up at one point or another, unless the parameter ε is large enough. The bottom right of figure 1 shows the image obtained with $\varepsilon = 0.01$, which corresponds to the first value for which the algorithm behaves stably. It is clear from the picture that the total variation is not thoroughly minimized, and this is because the algorithm converges to the minimum of $J(u) = TV(u) + \varepsilon\|u\|^2$ under the l^∞ constraints. Actually, for the displayed image, the (normalized) total variation is $\|u\|_{TV} = 9.438$, but the (normalized) quadratic term takes the value $\varepsilon\|u\|^2 = 171.9$, which makes it far from being negligible.

Our 'squared' algorithm, on the other hand, is stable and converges even when it is not well initialized. We have indeed tried several random initializations, always obtaining the same results, even though the amount of time required for convergence is (quite obviously) much longer than with the initialization given in the previous section.

At this point, we might add a few words on the speed of convergence: the image in figure 1 is obtained after only five iterations, and is already of high quality. But denoising is a 'simple' case, and for more general restoration tasks where H is more difficult to invert, the algorithm typically requires around 20 iterations or sometimes even more to yield a final high-quality result. (Note that each iteration corresponds to two transforms in $\mathcal{D}$, and to one fast unconstrained minimization.) Again, the number of iterations depends also on the chosen initialization; the ones we proposed earlier are generally quite efficient.

5. Results

5.1. Denoising

To complete the previous section, we compare the images obtained for the denoising experiment to that obtained with the Rudin–Osher–Fatemi algorithm. The result can be seen in figure 4: clearly, textures are much better preserved, and the overall aspect is much sharper.

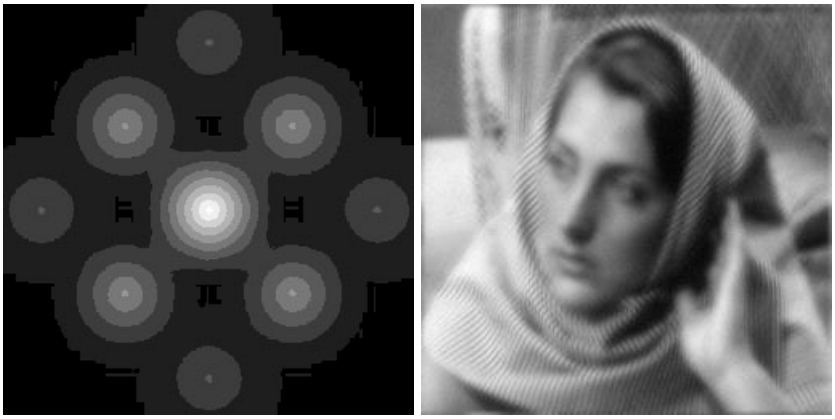

Figure 5. Left: filter with holes provided by Alcatel Space Industry. It is shown here in the frequency domain, and its values ranging in [0, 1] have been quantized for display (the black part corresponds to 0, the first grey level corresponds to the range]0, 0.12], ...). Right: blurred image.

Figure 6. Left: deblurring with the Rudin–Osher–Fatemi algorithm. Right: deblurring with our approach.

5.2. Deblurring

Next, we tackle a difficult deblurring task (with the courtesy of Alcatel Space Industry) which involves the blurring kernel displayed in figure 5, and a Gaussian noise of variance $\sigma = 2$. Besides its anisotropy, the kernel shows important holes at medium frequencies, and as a result, its inversion is quite delicate. In fact, it is hard to approximate with traditional methods: wavelet packets, for example, have dyadic supports which do not fit the holes in the filter at all.

Still, our method solves this in a straightforward manner, without requiring much more work than the denoising problems presented above: running the Rudin–Osher–Fatemi algorithm (see [20]), we compute u_0, from which we obtain the corresponding initial estimate for λ^0 in the manner described earlier. Using the same wavelet-packet dictionary as before, we then run 20 iterations of the Uzawa method with a threshold $\tau = 5$. The result is displayed in figure 6: without any noise blow-up, or edge blurring, the algorithm renders a sharp image, with most of the original texture. The Rudin–Osher–Fatemi result is also displayed, clearly showing the difference in the texture preservation.

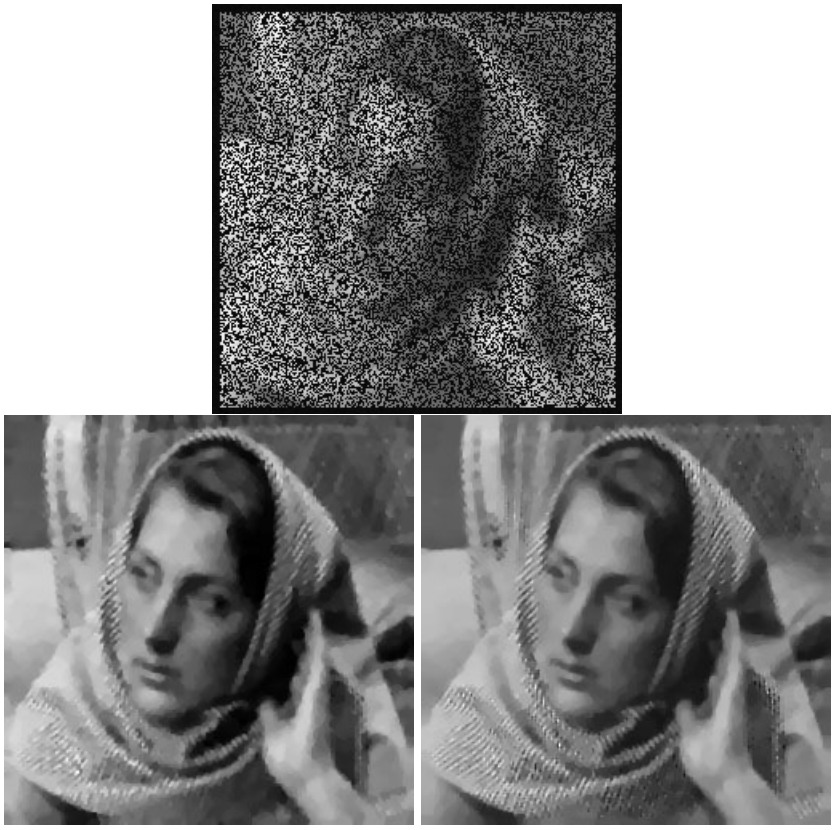

Figure 7. A mixed degradation: after the Alcatel blurring, 50% of the pixels are removed along with the borders (top). Bottom left: the Rudin–Osher–Fatemi result. Bottom right: the reconstruction with this paper's algorithm.

5.3. Limits of the model

Finally, we push the difficulty a little further, by assuming that 50% of the blurred image's pixels are lost, along with its borders. The degradation operator is thus of the form $H = H_2 \circ H_1$ where H_1 is a convolution, and H_2 an inpainting mask.

We then solve (P_*) with the same parameters as above, except that the initialization is refined by first applying a straightforward averaging on the destroyed image to fill in the holes. Figure 7 shows the result after 50 iterations, along with the degraded image, and the Rudin–Osher–Fatemi result. Note that this time, the solution to (P^*) is not so different from the classical Rudin–Osher–Fatemi approach: some textures are better captured with the L^∞ attach, but only slightly, and this is due to the fact that the dictionary is by no means adapted to the degradation operator. Recall from section 3 that the L^∞ constraints are given by the coefficients $\langle H(u), \Psi \rangle$, or equivalently, $\langle u, H^*(\Psi) \rangle$. But here, $H(u)$ and $H^*(\Psi)$ are images lacking 50% of their pixels, and because of this, they will in general be extremely irregular. Consequently, the class of images which have a sparse representation in the family $\{H^*(\Psi)\}$ should be very narrow, and the L^∞ constraint has no reason to carry much more information than the classical L^2 fidelity term.

What this last experiment shows is that even though the adaptation of $\mathcal{D}$ to H is not necessary to obtain good images, it can still play an important role. In fact, this role still needs

to be studied more thoroughly and the question remains open: given a class of images and a degradation H, how should the dictionary $\mathcal{D}$ be designed, if one is to aim at 'optimal' results?

5.4. More restoration examples

Because of limitations of space, we could only discuss a few examples here, and we invite the reader to visit the following webpage, where more results are presented and discussed: http://www.acm.caltech.edu/~'lintner/restoration

Acknowledgments

The authors would like to thank Frédéric Falzon (Alcatel Space) for his support. The work of SL was supported by Alcatel Space Industry, while studying at the Ecole Normale Supérieure de Cachan (France).

Appendix. Proof of proposition 1

The proof is based on the following lemma. This lemma states that a convex and coercive function has, at least, a linear growth towards infinity.

Lemma 1. *Let $J : \mathbb{R}^n \to \mathbb{R}$ be convex and coercive over $\mathbb{R}^n$. There exist $C_1 > 0$ and $C_2 \in \mathbb{R}$ such that, for any $u \in \mathbb{R}^n$,*

$$J(u) \geqslant C_1 \|u\|_2 + C_2.$$

Proof. Let us first remark that, since J is convex and coercive, its level sets are convex and compact. Therefore, for any $g \in \mathbb{R}^n$, satisfying $g \neq 0$, there exists v_g, in the level set $\{v, J(v) \leqslant \text{ in } f_w J(w) + 1\}$, such that g belongs to the normal cone to $\{v, J(v) \leqslant \text{ in } f_w J(w) + 1\}$, at v_g. Therefore, we know there exists $\alpha_g > 0$ such that $\alpha_g g \in \partial J(v_g)$, where $\partial J(v_g)$ denotes the sub-differential of J, at v_g (see [21], p 222). So, we have, for any $u \in \mathbb{R}^n$,

$$J(u) \geqslant \langle \alpha_g g, u - v_g \rangle + J(v_g).$$

Let $(g_j)_{1 \leqslant j \leqslant n}$ be the canonical basis of $\mathbb{R}^n$. We can repeat the above construction, for all εg_j, with $\varepsilon = \pm 1$ and $j \in \{1, \ldots, n\}$. We therefore obtain that, for any $j \in \{1, \ldots, n\}$ and any $\varepsilon = \pm 1$, there exist $v_j^\varepsilon \in \mathbb{R}^n$ and $\alpha_j^\varepsilon > 0$ such that

$$J(u) \geqslant \left\langle \alpha_j^\varepsilon \varepsilon g_j, u - v_j^\varepsilon \right\rangle + J\left(v_j^\varepsilon\right),$$

for all $u \in \mathbb{R}^n$.

So, we have for any $u \in \mathbb{R}^n$,

$$J(u) \geqslant \max_{j,\varepsilon} \left(\left\langle \alpha_j^\varepsilon \varepsilon g_j, u \right\rangle + C_j^\varepsilon \right),$$

with $C_j^\varepsilon = J\left(v_j^\varepsilon\right) - \left\langle \alpha_j^\varepsilon \varepsilon g_j, v_j^\varepsilon \right\rangle$. Denoting $C_2 = \min_{j,\varepsilon} C_j^\varepsilon$, we obtain

$$J(u) \geqslant \max_{j,\varepsilon} \alpha_j^\varepsilon \langle \varepsilon g_j, u \rangle + C_2.$$

Noting $C_1' = \min_{j,\varepsilon} \alpha_j^\varepsilon$, we have

$$J(u) \geqslant C_1' \max_{j,\varepsilon} \langle \varepsilon g_j, u \rangle + C_2 = C_1' \|u\|_\infty + C_2,$$

and $C_1' > 0$. We can then conclude. Indeed, for any $u \in \mathbb{R}^n$, $\|u\|_\infty \geqslant \frac{\|u\|_2}{\sqrt{n}}$. □

Proof of proposition 1. Since J is convex and coercive, lemma 1 shows that there exist $C_1 > 0$ and $C_2 \in \mathbb{R}$ such that, for any $u \in \mathbb{R}^n$,

$$J(u) \geqslant C_1 \|u\|_2 + C_2.$$

Now, for any $i \in \{1, \ldots, M\}$, there exists $w_i \in \mathbb{R}^n$ such that

$$\varphi_i(u) = \langle u, w_i \rangle.$$

Therefore, we have, for any $\lambda \in \mathbb{R}_+^M$, and any $u \in \mathbb{R}^n$ such that $\|u\|_2 \geqslant -\frac{C_2}{C_1}$,

$$\begin{aligned} L(u, \lambda) &\geqslant (C_1 \|u\|_2 + C_2)^2 + \left\langle u, \sum_{i=1}^{M} \lambda_i w_i \right\rangle \\ &\geqslant (C_1 \|u\|_2 + C_2)^2 - \|u\|_2 \left\| \sum_{i=1}^{M} \lambda_i w_i \right\|_2, \end{aligned}$$

which guarantees that, regardless of $\lambda \in \mathbb{R}^M$, $L(u, \lambda)$ is coercive in u. □

References

[1] Acar R and Vogel C 1994 Analysis of bounded variation methods for ill-posed problems *Inverse Problems* **10** 1217–29

[2] Candes E and Guo F 2002 New multiscale transforms, minimum total variation synthesis: application to edge-preserving image reconstruction *Signal Process.* **82** 1519–43

[3] Chan T F and Zhou H M 2000 Optimal construction of wavelet coefficients using total variation regularization in image compression *Technical Report* CAM 00-27 (Los Angeles, CA: University of California)

[4] Chen S, Donoho D and Saunders M 1995 Atomic decomposition by basis pursuit *SPIE, Int. Conf. on Wavelets (San Diego, July 1995)*

[5] Ciarlet P 1989 *Introduction to Numerical Linear Algebra and Optimisation* (Cambridge: Cambridge University Press)

[6] Coifman R R and Sowa A 2000 Combining the calculus of variations and wavelets for image enhancement *Appl. Comput. Harmon. Anal.* **9** 1–18

[7] Donoho D L and Johnstone I M 1994 Ideal spatial adaptation by wavelet shrinkage *Biometrika* **81** 425–55

[8] Durand S and Froment J 2003 Reconstruction of wavelet coefficients using total variation minimization *SIAM J. Sci. Comput.* **24** 1754–67 (a preliminary version is available at: http://www.cmla.ens-cachan.fr/Cmla/Publications/2001/index.html)

[9] Fiacco A V and McCormick G P 1990 Nonlinear programming *Classics in Applied Mathematics* (Philadelphia, PA: SIAM)

[10] Ishwar P and Moulin P 1999 Multiple-domain image modeling and restoration *Proc. IEEE Int. Conf. Image Processing (Kobe, Japan, Oct. 1999)* vol 1, pp 362–6

[11] Jalobeanu A, Blanc-féraud L and Zerubia J 2003 Satellite image deblurring using complex wavelet packets *Int. J. Comput. Vis.* **51** 205–17

[12] Luenberger D C 1984 *Linear and Nonlinear Programming* 2nd edn (Reading, MA: Addison-Wesley)

[13] Malgouyres F 2001 A noise selection approach of denoising *Technical Report* CAM 01-04 (Los Angeles, CA: University of California)

[14] Malgouyres F 2001 A noise selection approach of image restoration *Wavelet: Applications in Signal and Image Processing IX (San Diego, USA, July 2001)* vol 4478, ed M A Unser, A F Laine and A Aldroubi (Bellingham, WA: SPIE Optical Engineering Press) pp 34–41

[15] Malgouyres F 2002 A framework for image deblurring using wavelet packet bases *Appl. Comput. Harmon. Anal.* **12** 309–31

[16] Malgouyres F 2002 Mathematical analysis of a model which combines total variation and wavelets for image restoration *J. Inf. Process.* **2** 1–10

[17] Malgouyres F 2002 Minimizing the total variation under a general convex constraint for image restoration *IEEE Trans. Image Process.* **11** 1450–6

[18] Malgouyres F and Guichard F 2001 Edge direction preserving image zooming: a mathematical and numerical analysis *SIAM J. Numer. Anal.* **39** 1–37 (a preliminary version is available at: http://www.zeus.math.univ-paris13.fr/∼malgouy)

[19] Mallat S and Zhang Z 1993 Matching pursuits with time–frequency dictionaries *IEEE Trans. Signal Process.* **41** 3397–415

[20] Rudin L, Osher S and Fatemi E 1992 Nonlinear total variation based noise removal algorithms *Physica* D **60** 259–68

[21] Rockafellar R T 1970 *Convex Analysis* (Princeton, NJ: Princeton University Press)

[22] Starck J L, Donoho D and Candes E 2001 Very high quality image restoration by combining wavelets and curvelets *Wavelet and Applications in Signal and Image Processing IX (San Diego, Aug. 2001)* vol 4478 ed M A Unser, A F Laine and A Aldroubi (Bellingham, WA: SPIE Optical Engineering Press) pp 9–19

[23] Tramini S 1999 Problèmes inverses et EDP pour le décodage et la déconvolution d'images *PhD Thesis* University of Nice–Sophia-Antipolis, I3S

[24] Woog L 1996 Adapted waveform algorithms for denoising *PhD Thesis* Yale University (available at http://www.cs.yale.edu/homes/∼woog.html)