A Stable Method Solving the Total Variation Dictionary Model with L-infinity Constraints

Liyan Ma, Lionel Moisan, Jian Yu, Tieyong Zeng

To cite this version:

Liyan Ma, Lionel Moisan, Jian Yu, Tieyong Zeng. A Stable Method Solving the Total Variation Dictionary Model with L-infinity Constraints. 2013. hal-00826615v1

HAL Id: hal-00826615

https://hal.science/hal-00826615v1

Preprint submitted on 27 May 2013 (v1), last revised 30 Dec 2014 (v2)

A STABLE METHOD SOLVING THE TOTAL VARIATION DICTIONARY MODEL WITH L^∞ CONSTRAINTS

Liyan Ma
School of Computer and Information Technology, Beijing Jiaotong University
Beijing, China

Lionel Moisan
MAP5, Université Paris Descartes
Paris, France

Jian Yu
School of Computer and Information Technology, Beijing Jiaotong University
Beijing, China

Tieyong Zeng
Department of Mathematics, Hong Kong Baptist University
Kowloon Tong, Hong Kong, China

Abstract. Image restoration plays an important role in image processing, and numerous approaches have been proposed to tackle this problem. This paper presents a modified model for image restoration, that is based on a combination of Total Variation (TV) and Dictionary approaches. Since the well-known TV regularization is non-differentiable, the proposed method utilizes its dual formulation instead of its approximation in order to exactly preserve its properties. The data-fidelity term combines the one commonly used in image restoration and a wavelet thresholding based term. Then, the resulting optimization problem is solved via a first-order primal-dual algorithm. Numerical experiments demonstrate the good performance of the proposed model. In a last variant, we replace the classical TV by the nonlocal TV regularization, which results in a much higher quality of restoration.

1. Introduction

A classical inverse problem in modern imaging sciences is image restoration, the aim of which is to recover an ideal image $u \in L^2(\Omega)$ from a degraded observation $g \in L^2(\Omega)$, knowing that

$$g = Au + b, \tag{1}$$

where Ω is a connected bounded open subset of $\mathbb{R}^2$ with compact Lipschitz boundary, and $b \in L^2(\Omega)$ is a realization of a white Gaussian noise with standard deviation σ. Here A is a known linear operator from $L^2(\Omega)$ to $L^2(\Omega)$, that depends on the considered inverse problem: $A = I$ (identity) for denoising, A is convolution operator for deblurring, and A is the canonical projection on known pixels in the case of inpainting.

2000 *Mathematics Subject Classification:* Primary: 68U10; Secondary: 49N45, 94A08, 65T60.

Key words and phrases: Image restoration, total variation, nonlocal total variation, proximal gradient method, wavelet packet decomposition.

Image restoration is an ill-posed problem. Directly solving (1) may not yield meaningful solutions. To overcome this difficulty, many regularization methods have been proposed in the literature, e.g. Tikhonov's regularization [44], the Total Variation (TV) criterion [40], non-local graph regularization [39], and wavelet-based sparsity assumptions [23]. In this paper, we mainly focus on the TV approach, first proposed by Rudin, Osher and Fatemi [40]:

$$\min_u \ TV(u) + \frac{\lambda}{2} \|Au - g\|_2^2. \tag{2}$$

Here the Total Variation of an image $u \in L^2(\Omega)$ is defined by

$$TV(u) = \int_\Omega |Du|,$$

the gradient Du being understood in the distribution sense.

TV regularization has been extensively used in image processing applications, for it is a simple term that manages to preserve sharp features in images. However, due to the non-differentiability of the TV operator, the functional (2) is difficult to minimize. In the past decades, many methods have been developed to address this problem. To name a few, let us mention variable splitting methods [1, 28, 41, 45], the alternating direction method of multipliers (ADMM) [26, 38], dual methods [13] and primal-dual methods [15, 16, 51]. Some of these methods have strong connections, which are studied in [41, 43].

Besides TV regularization, another famous approach for image restoration is the wavelet-based method. Since wavelet bases can provide a multiscale and sparse representation for images (see [21]), they have been widely adopted in image restoration [22, 24, 30, 31, 32] after their introduction by Donoho and Johnstone [23]. Later, Chambolle et al. [14] and Steidl et al. [42] established the relationship between wavelet-based and TV-based methods. Eventually, by combining TV and wavelet methods, Candes and Guo [12] and Malgouyres [31] proposed a hybrid approach for image restoration. The optimization model in [31] is

$$\begin{cases} \min\limits_u TV(u) \\ s.t. \ \|Au - g\|_{\mathcal{D},\infty} \leq \tau, \end{cases} \tag{3}$$

where $\mathcal{D} \subset L^2(\Omega)$ is a finite dictionary, the parameter $\tau > 0$ depends on the noise level, and $\|u\|_{\mathcal{D},\infty} = \sup\limits_{\Psi \in \mathcal{D}} |\langle u, \Psi \rangle|$. As pointed out by Malgouyres in [31], taking the (infinite) dictionary $\mathcal{D} = \left\{ w \in L^2(\Omega), \|w\|_2 = 1 \right\}$ would reduce (3) to the ROF model. In the same paper, he showed that when the dictionary contains wavelet/wavelet packet bases and their opposites, the new model preserves textures better than the ROF model.

The choice of the dictionary is very important, since a carefully selected dictionary can improve the quality of restoration. In order to obtain optimal results through (3), Zeng and Malgouyres [48] investigated twelve Gabor dictionaries for (3), and the experimental results clearly demonstrated that the choice of the dictionary deeply affects the performance of this model. Later, Zeng and Ng [47, 49] provided a theoretical study on the model (3) and the choice of the dictionary. They proved the existence of solutions of (3), and showed that the dictionary should represent sparsely the curvatures of the ideal image in order to achieve a better denoising performance.

The minimization problem (3) is evidently a rather difficult task, and it has been rarely addressed in the literature. In order to approximate the solution of (3), Lintner and Malgouyres [29] proposed the following LM model:

$$\left\{\begin{array}{l} \min\limits_{u} \ TV_\beta(u)^2 := \left(\displaystyle\int_\Omega \varphi_\beta\left(|\nabla u|\right) + \beta\langle u, \vec{1}\rangle^2\right)^2 \\ s.t. \ \forall\Psi \in \mathcal{D}, \quad \langle Au - g, \Psi\rangle \leq \tau \end{array}\right. \tag{4}$$

where $\vec{1}$ denotes a constant vector with all coordinates equal to 1, β is a small positive number, and $\varphi_\beta\left(t\right) = \sqrt{t^2+\beta^2}$. As the objective function in (4) is differentiable, convex and coercive, a dual Uzawa method was used in [29] to solve this optimization task. Note that in (4), as well as in all this paper, the dictionary $\mathcal{D}$ is assumed to be symmetric, which means that

$$\forall\psi \in \mathcal{D}, \quad -\psi \in \mathcal{D}. \tag{5}$$

Hence, the constraints in (4) could as well be written $|\,\langle Au - g, \Psi\rangle\,| \leq \tau$.

Although improving the quality of the recovered image compared to ROF Model, the approach proposed in (4) still has some drawbacks. First, the use of $TV_\beta(u)^2$ to replace $TV(u)$ causes a well-known unpleasant smear effect; second, the term $\beta\langle u, \vec{1}\rangle$ introduced in (4) does make the objective function coercive (this is critical to make the Uzawa method work), but it also imposes an extra assumption on the mean value of the ideal image u which might affect the quality of the restored image; third, the numerical scheme adopted in [29] may be unstable in some situations, and imposes a careful (small enough) choice of the gradient-ascent time step used in Uzawa method.

The goal of this paper is to fix these issues. We propose a modified model which integrates the ROF and the dictionary model (3) seamlessly. Moreover, inspired by the recent works of Chambolle and Pock [15], we give two stable and effective algorithms to solve the associated optimization problem, one for the general case and one in the special case of image denoising ($A = I$).

The outline of the paper is as follows. In Section 2, we introduce the proposed modified model and study some basic properties. In Section 3, some notations and preliminaries (especially in the discrete settings) are presented, which will be used in the rest of the paper. In Section 4, we briefly review Chambolle-Pock Algorithm, then we describe how it can be applied to the proposed model (Section 5). We propose two algorithms that solve the corresponding unconstrained and constrained models respectively. In Section 6, we report various numerical results to demonstrate the good performance of the proposed approach. Finally, we briefly investigate in Section 7 an interesting variant of the proposed model, where the TV regularization term is replaced by the nonlocal TV [27], before we conclude in Section 8.

2. Proposed model

This section presents the primal formulation of the proposed model and a study of its basic properties. In Section 5, we will solve the proposed model in its primal-dual formulation to exactly preserve its properties.

2.1. MODEL. Using TV regularization rather than the $TV_\beta(u)^2$ term, the minimization problem (4) becomes

$$\left\{\begin{array}{l}\min\limits_u \ TV(u)\\ s.t.\ \ \forall\Psi\in\mathcal{D},\quad \langle Au-g,\Psi\rangle-\tau\le 0.\end{array}\right. \tag{6}$$

From a statistical point of view, since we are dealing with additive white Gaussian noise, the probabilistic formulation of (1) is

$$p\,(g|u)=\prod_{x\in\Omega}\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(Au(x)-g(x))^2}{2\sigma^2}\right).$$

The corresponding energy function, $-\log p(g|u)$, is, up to a constant additive and multiplicative term, the data-fidelity term $\|Au-g\|^2$ of the ROF model. We propose to add this natural term to the energy used in (6) in order to control the coercivity and to enforce the data-fidelity constraint on the restored image. Thus, we obtain a modified model defined by the primal formulation

$$\left\{\begin{array}{l}\min\limits_u \ TV(u)+\dfrac{\lambda}{2}\,\|Au-g\|_2^2\\ s.t.\ \ \forall\Psi\in\mathcal{D},\quad \langle Au-g,\Psi\rangle-\tau\le 0.\end{array}\right. \tag{7}$$

Both parameters λ and τ control the regularization induced by the model, but in different ways. For fixed τ, a small value of λ makes the restored image less noisy, but may provide a solution that is too smooth; on the contrary, a large value of λ tends to produce sharper but also more noisy results. We could probably use an estimation of the noise level to automatically set the value of λ (as it is done in some other works), but this is beyond the scope of this paper. In this work, we decide to tune it empirically.

Model (7) seamlessly integrates the ROF model and the original TV dictionary model. Indeed, if $\lambda=0$, it is exactly (3); if the dictionary is empty, then it reduces to the classical ROF model. We also avoid the extra assumption needed in (4). Experiments will later demonstrate that the least-squares term is very important to stabilize the algorithm and to provide a high-quality restored image.

2.2. EXISTENCE OF A SOLUTION. We here prove the existence of a solution for the proposed model (7) in the continuous settings. First, we recall the definition of the space of functions with bounded variation (BV), which plays an important role in image modeling. Note that a rather complete discussion on this topic can be found in [2] and [3].

Definition 2.1. The space of functions of bounded variation is

$$BV\,(\Omega)=\left\{u\in L^1\,(\Omega)\,;\int_\Omega|Du|<\infty\right\},$$

$$\text{where}\quad\int_\Omega|Du|=\sup\left\{\int_\Omega u\,\mathrm{div}\,\varphi\,dx;\ \varphi=(\varphi_1,\varphi_2)\in C_0^\infty\,(\Omega)^2,\|\varphi\|_{L^\infty(\Omega)}\le 1\right\}.$$

Endowed with the norm $\|u\|_{BV}=\int_\Omega|u|+\int_\Omega|Du|$, $BV(\Omega)$ is a Banach space, and $TV(\cdot)$ is a proper convex lower semi-continuous functional from $L^2\,(\Omega)$ to $(-\infty,+\infty]$ (see Lemma 8 in [49]).

Proposition 1. *Assume that Ω is a connected bounded open subset of $\mathbb{R}^2$ with compact Lipschitz boundary. Assume that the set*

$$S_\tau := \{u \in L^2(\Omega);\ \|Au - g\|_2 < \tau\}. \tag{8}$$

is nonempty and $A\vec{1} = \vec{1}$. If $\mathcal{D}$ is a finite symmetric subset of the closed unit ball of $L^2(\Omega)$, then for any $\lambda \geq 0$, Problem (7) admits at least one solution.

Proof. • We first consider the (simpler) case $\lambda > 0$. The ROF functional

$$u \ \mapsto\ TV(u) + \frac{\lambda}{2}\|Au - g\|_2^2 \tag{9}$$

is convex, closed (proper and lower semicontinuous), and coercive (because $A\vec{1} = \vec{1}$, see [18]) on $L^2(\Omega)$, so it admits a minimizer (see [25] p.35). Now let us consider the feasible set

$$C_\tau = \left\{u \in L^2(\Omega);\ \forall \Psi \in \mathcal{D},\ \langle Au - g, \Psi\rangle \leq \tau\right\}. \tag{10}$$

Obviously, C_τ is a nonempty closed convex set, and we can rewrite (7) as

$$\min_u\ TV(u) + \frac{\lambda}{2}\|Au - g\|_2^2 + \delta_{C_\tau}(u),$$

where $\delta_{C_\tau}(u)$ is the indicator function of C_τ (which takes the value 0 if $u \in C_\tau$, and the value $+\infty$ otherwise). The functional above is, like the ROF functional (9), convex, closed, and coercive on $L^2(\Omega)$. Therefore, (7) admits at least one solution.

• Now let us consider the case $\lambda = 0$. In [49], Zeng and Ng proved that there exists a solution of (7) when $A = I$. Here, we prove that this remains true when A is a linear operator such that S_τ is nonempty. Denote as in (10) by C_τ the feasible set of (7), and take $v^* \in S_\tau$. As C_0^∞ is dense in $L^2(\Omega)$ and A is bounded, we can choose a $v_0 \in C_0^\infty$ such that $\|A(v_0 - v^*)\|_2 < \tau - \|Av^* - g\|_2$. Then we have

$$\|Av_0 - g\|_2 \leq \|A(v_0 - v^*)\|_2 + \|Av^* - g\|_2 < \tau$$

and by Cauchy-Schwarz inequality, $\langle Av_0 - g, \Psi\rangle < \tau$ for any $\Psi \in \mathcal{D}$ (because $\|\Psi\|_2 \leq 1$). Hence $v_0 \in \mathrm{int}(C_\tau)$ and since Ω is bounded, we also have $TV(v_0) < +\infty$. Hence, Slater condition [46] holds and there exists a minimizing sequence (u_j) in C_τ such that $\lim_{j\to\infty} TV(u_j) = M$, where M is the minimum value in (7) (recall that $\lambda = 0$ here). Note that we can assume without loss of generality that $TV(u_j) \leq M + 1$ for all j.

Denote by $m_\Omega(f)$ the mean of f on Ω. Since Ω is a connected bounded open subset of $\mathbb{R}^2$ with compact Lipschitz boundary, it is an extension domain (using Prop. 3.21 of [2]) and by Poincaré inequality (see [2], Remark 3.50]), there exists a constant $c_0 > 0$ such that

$$\forall j, \quad \|u_j - m_\Omega(u_j)\|_{L^2(\Omega)} \leq c_0 \cdot TV(u_j) \leq c_0(M+1). \tag{11}$$

From (11), it is easy to deduce that $\|u_j - m_\Omega(u_j)\|_{\mathrm{BV}}$ is bounded. As every bounded sequence in $\mathrm{BV}(\Omega)$ is relatively compact in $L^1(\Omega)$ (see [3]), there exists a subsequence $\left(u_j^{(1)} - m_\Omega\left(u_j^{(1)}\right)\right)$ which converges to some $w^* \in \mathrm{BV}(\Omega)$ both in $L^1(\Omega)$, and weakly in $L^2(\Omega)$ (thanks to (11)). By lower semicontinuity of TV with respect to the $L^1(\Omega)$ topology (see [3]), one obtains

$$TV(w^*) \leq \liminf_{j\to\infty} TV\left(u_j^{(1)} - m_\Omega\left(u_j^{(1)}\right)\right) = \liminf_{j\to\infty} TV\left(u_j^{(1)}\right) = M.$$

Now let us prove that for some constant c, the image w^*+c is in the feasible set C_τ. As $u_j^{(1)} \in C_\tau$, we know that for any $\Psi \in \mathcal{D}$, we have

$$\left\langle u_j^{(1)} - m_\Omega\left(u_j^{(1)}\right), A^*\Psi \right\rangle + m_\Omega\left(u_j^{(1)}\right) \int_\Omega A^*\Psi \leq \tau + \langle g, \Psi \rangle . \tag{12}$$

If $\int_\Omega A^*\Psi = 0$ for any $\Psi \in \mathcal{D}$, then taking the limit in (12) yields

$$\langle w^*, A^*\Psi \rangle \leq \tau + \langle g, \Psi \rangle ,$$

so that w^* is in the feasible set and it is a solution of (7).

Otherwise, we have $\alpha := \int_\Omega A^*\Psi_0 \neq 0$ for some $\Psi_0 \in \mathcal{D}$, and using (12) with $\Psi = \varepsilon\Psi_0$ yields, thanks to (11),

$$\forall j, \forall \varepsilon \in \{-1, 1\}, \quad \varepsilon\alpha \cdot m_\Omega\left(u_j^{(1)}\right) \leq \tau + \langle g, \varepsilon\Psi_0 \rangle + c_0(M+1)\|A^*\|_2,$$

which proves that $m_\Omega\left(u_j^{(1)}\right)$ is bounded. Hence, we can choose a subsequence $u_j^{(2)}$ such that $m_\Omega\left(u_j^{(2)}\right)$ converges to some constant c. Taking the limit for this subsequence in (12), we obtain

$$\forall \Psi \in \mathcal{D}, \quad \langle w^*, A^*\Psi \rangle + c \int_\Omega A^*\Psi \leq \tau + \langle g, \Psi \rangle ,$$

so that w^*+c belongs to C_τ ($\langle A(w^*+c) - g, \Psi \rangle \leq \tau$) and it is a solution of (7). $\square$

3. NOTATIONS AND PRELIMINARIES

Now let us go to the discrete settings. Before giving details of the algorithms, we need to introduce some notations and preliminaries which will be used in the following sections. In order to simplify the notations, we here focus on gray-level images. Corresponding notions for color images, that will be considered in the experiments, can be found in [9].

3.1. NOTATIONS.

Discrete images will be represented by elements of the Euclidean space $E = \mathbb{R}^{m\times n}$. Let $\langle \cdot, \cdot \rangle$ denote the standard inner product, and $\|\cdot\|_p$ be the standard l^p-norm. The transpose (dual operator) A^* of a linear transform A is defined as usual by $\langle Au, g \rangle = \langle u, A^*g \rangle$. I denotes the identity matrix.

3.1.1. *The discrete gradient and divergence operators.*

Denote $U = E \times E$. The gradient operator is a map $\nabla : E \to U$ defined by $\nabla u = (\partial_x u, \partial_y u)^{\mathrm{T}}$ as indicated below. The adjoint of the gradient (∇^*) is the opposite of the divergence operator ($-\mathrm{div}$), so that for any $z = (p\ q)^{\mathrm{T}} \in U$, $\mathrm{div}(z) = -(\partial_x^* p + \partial_y^* q)$. The maximum norm of z is defined by $\|z\|_\infty = \max_{i,j} \sqrt{p_{i,j}^2 + q_{i,j}^2}$.

We shall consider two different boundary conditions for the discretization of ∇ and $\mathrm{div} = -\nabla^*$, namely:

1) **Symmetric Boundary Conditions.** The discrete gradient operators ∂_x and ∂_y are defined as

$$(\partial_x u)_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if} \quad 1 \leq i < m, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$(\partial_y u)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if} \quad 1 \leq j < n, \\ 0 & \text{otherwise.} \end{cases}$$

The discrete operators ∂_x^* and ∂_y^* are defined accordingly as

$$(\partial_x^* p)_{i,j} = \begin{cases} p_{i,j} & \text{if} \quad i = 1, \\ p_{i,j} - p_{i-1,j} & \text{if} \quad 1 < i < m, \\ -p_{i-1,j} & \text{if} \quad i = m, \end{cases}$$

and

$$(\partial_y^* q)_{i,j} = \begin{cases} q_{i,j} & \text{if} \quad j = 1, \\ q_{i,j} - q_{i,j-1} & \text{if} \quad 1 < j < n, \\ -q_{i,j-1} & \text{if} \quad j = n. \end{cases}$$

2) **Periodic Boundary Conditions.** In that case, all operators are defined using a circular convention for the coordinates, that is, $m+1 := 1$ for the i coordinate, and $n+1 := 1$ for the j coordinate:

$$(\partial_x u)_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if} \quad 1 \leq i < m, \\ u_{1,j} - u_{i,j} & \text{if} \quad i = m, \end{cases}$$

$$(\partial_y u)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if} \quad 1 \leq j < n, \\ u_{i,1} - u_{i,j} & \text{if} \quad j = n. \end{cases}$$

$$(\partial_x^* p)_{i,j} = \begin{cases} p_{i,j} - p_{m,j} & \text{if} \quad i = 1, \\ p_{i,j} - p_{i-1,j} & \text{if} \quad 1 < i \leq m, \end{cases}$$

$$(\partial_y^* q)_{i,j} = \begin{cases} q_{i,j} - q_{i,n} & \text{if} \quad \mathrm{j} = 1, \\ q_{i,j} - q_{i,j-1} & \text{if} \quad 1 < j \leq m. \end{cases}$$

3.2. Proximal operators. The proximal operator, introduced by Moreau [35, 36], is an important concept in convex optimization and has been widely used recently [15, 20, 34].

Definition 3.1 (Proximal Operator)**.** Let φ be a convex, proper and closed (hence lower semi-continuous) function on $\mathbb{R}^d$. The proximal operator of φ is defined by

$$\forall x \in \mathbb{R}^d, \quad \operatorname{prox}_\varphi(x) = \arg\min_y \ \varphi(y) + \frac{1}{2} \|x - y\|_2^2 .$$

We recall the following useful result for the calculation of $\operatorname{prox}_{f+\delta_C}$, that plays a key role in solving the constrained formulation of the model we propose.

Proposition 2. *(see* [19]*, Proposition 2.1) Let* $f : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ *be a convex, proper and closed function, and let* C *be a closed convex subset of* $\mathbb{R}$ *such that* $C \cap \operatorname{dom} f \neq \emptyset$*. Then* $\operatorname{prox}_{f+\delta_C} = P_C \circ \operatorname{prox}_f$*, where* P_C *is the projection on* C*.*

3.3. The primal-dual problem. Most recent papers in the literature [13, 16, 51] use the dual or the primal-dual formulation for TV-related convex optimization problems, because they can reach better performances than the primal formulation for this kind of non-differentiable convex functionals. This section shortly reviews the primal-dual framework.

Let us consider the primal minimization problem

$$\min_u F(Ku) + G(u), \tag{13}$$

where K is a continuous linear map from $E = \mathbb{R}^{m \times n}$ to $U = E \times E$ with norm

$$\|K\|_2 = \max\left\{\|Ku\|_2 ;\ u \in E, \|u\|_2 \leq 1\right\},$$

and $F : U \to [0, +\infty)$ and $G : E \to [0, +\infty)$ are proper, convex, lower semi-continuous functions. The convex conjugate F^* of F [6] is defined by

$$\forall p \in U, \quad F^*(p) = \sup_{w \in U} \langle p, w \rangle - F(w).$$

Since $F^{**} = F$, we have

$$\forall u \in E, \quad F(Ku) = \sup_{p \in U} \langle p, Ku \rangle - F^*(p).$$

Thus the saddle-point optimization problem or the primal-dual formulation of (13) can be written

$$\min_{u \in E} \max_{p \in U} \langle Ku, p \rangle + G(u) - F^*(p). \tag{14}$$

4. REVIEW OF CHAMBOLLE-POCK ALGORITHM

In this section, we briefly review the first-order primal-dual algorithm proposed by Chambolle and Pock [15]. This algorithm is designed to solve the convex optimization problem (14). Let $(\hat{u}, \hat{p}) \in E \times U$ be a solution of the problem (14), then we have the optimality conditions:

$$\begin{cases} K\hat{u} & \in & \partial F^*(\hat{p}), \\ -K^*\hat{p} & \in & \partial G(\hat{u}). \end{cases}$$

Assuming that prox_{F^*} and prox_G are easily computed, Chambolle and Pock solve the optimization problem (14) with respect to u and p respectively. To accelerate the algorithm, they use a multi-step scheme adopted in many algorithms [4, 37] to replace the classical one-step scheme. Algorithm 1 below summarizes the general scheme proposed by Chambolle and Pock [15], when the parameter θ they introduce is set to 1.

Algorithm 1 General scheme of Chambolle-Pock Algorithm

```
Initialization: $\bar{u}^0 = u^0$, $p = 0$, $t_p > 0$, $t_u > 0$, and $t_p t_u \|K\|_2^2 < 1$.
Output: $u^N$.
Iterate for $n = 0, 1, \dots N-1$:
Step 1. $p^{n+1} = \text{prox}_{t_p \cdot F^*}\left(p^n + t_p \cdot K\bar{u}^n\right)$
Step 2. $u^{n+1} = \text{prox}_{t_u \cdot G}\left(u^n - t_u \cdot K^* p^{n+1}\right)$
Step 3. $\bar{u}^{n+1} = 2u^{n+1} - u^n$
```

Step 3 in Algorithm 1 can be seen as an approximate intergradient step. The authors proved that Algorithm 1 converges at the rate $O(1/n)$ on the primal iterates.

Remark 1. We only presented here the basic algorithmic framework proposed by Chambolle and Pock in [15]. Their paper also describes some variants which can achieve $O(1/n^2)$ convergence under certain conditions.

5. APPLICATION OF THE CHAMBOLLE-POCK ALGORITHM FOR OUR MODEL

In this section, after providing the discrete form of the proposed model, we propose two algorithms for solving it. One solves the corresponding unconstrained formulation of the model, while the other solves the constrained formulation.

5.1. Discrete form. The discrete TV is defined as $TV(u) = \|\nabla u\|_1$, that is, $TV(u) = F(Ku)$ with $K = \nabla$ and $F(\cdot) = \|\cdot\|_1$ using the previous notations. Following the general scheme of Section 4, we obtain

$$TV(u) = \sup_{p \in U} \langle \nabla u, p \rangle - F^*(p) = \sup_{p \in U} - \langle u, \operatorname{div} p \rangle - F^*(p),$$

where $F^* = \delta_P$ is the indicator function of $P = \{p \in U;\ \|p\|_\infty \leq 1\}$. Rewriting the TV term of (7) in its dual formulation, the primal-dual formulation of the proposed model becomes

$$\left\{ \begin{array}{l} \min\limits_u \max\limits_p - \langle u, \operatorname{div} p \rangle + \frac{\lambda}{2} \|Au - g\|_2^2 - \delta_P(p) \\ s.t.\ \forall \Psi \in \mathcal{D}, \quad \langle Au - g, \Psi \rangle - \tau \leq 0 \end{array} \right. \tag{15}$$

We now present two algorithms to solve this optimization problem.

5.2. Algorithm for solving the unconstrained formulation. We first derive an unconstrained formulation of (15), then give an algorithm to solve it. Using the method of Lagrange multipliers, we can rewrite (15) as

$$\begin{aligned} \min_u \max_{p, \lambda_{\mathcal{D}}} L(u, p, \lambda_{\mathcal{D}}) = & - \langle u, \operatorname{div} p \rangle + \frac{\lambda}{2} \|Au - g\|_2^2 - \delta_P(p) \\ & + \sum_{\psi \in \mathcal{D}} \Big(\lambda_\Psi \left(\langle Au - g, \Psi \rangle - \tau \right) - \delta_Q(\lambda_\Psi) \Big), \end{aligned} \tag{16}$$

where $\lambda_{\mathcal{D}} = (\lambda_\Psi)_{\Psi \in \mathcal{D}} \in \mathbb{R}^{\mathcal{D}}$ is the (finite dimensional) vector of Lagrange multipliers associated to the constraints $\langle Au - g, \Psi \rangle - \tau \leq 0$, and $Q = \{q \in \mathbb{R};\ q \geq 0\}$.

Obviously, the objective function in (16) is convex, and this is a primal-dual problem (the primal variable is u, the dual variable is $(p, \lambda_{\mathcal{D}})$). The form (14) can be obtained with $Ku = (\nabla u, (\langle Au, \Psi \rangle)_{\psi \in \mathcal{D}})$, $G(u) = \frac{\lambda}{2} \|Au - g\|^2$ and $F^*(p, \lambda_{\mathcal{D}}) = \delta_P(p) + \sum_{\Psi \in \mathcal{D}} \delta_Q(\lambda_\Psi) + \lambda_\Psi(\langle g, \Psi \rangle + \tau)$. Thus, applying Algorithm 1 to (16) yields the following steps:

$$\left\{ \begin{array}{lcl} p^{n+1} & = & \arg\min\limits_p -L\left(\bar{u}^n, p, \lambda_{\mathcal{D}}^n\right) + \frac{1}{2t_p} \|p - p^n\|_2^2, \\ {\lambda_{\mathcal{D}}}^{n+1} & = & \arg\min\limits_{\lambda_{\mathcal{D}}} -L\left(\bar{u}^n, p^n, \lambda_{\mathcal{D}}\right) + \frac{1}{2t_\lambda} \|\lambda_{\mathcal{D}} - \lambda_{\mathcal{D}}^n\|_2^2, \\ u^{n+1} & = & \arg\min\limits_u L\left(u, p^{n+1}, \lambda_{\mathcal{D}}^{n+1}\right) + \frac{1}{2t_u} \|u - u^n\|_2^2. \\ \bar{u}^{n+1} & = & 2u^{n+1} - u^n. \end{array} \right. \tag{17}$$

Each sub-problem is solved as follows:

A. Update p: This sub-optimization problem is equivalent to

$$\min_p - \langle \nabla \bar{u}^n, p \rangle + \delta_P(p) + \frac{1}{2t_p} \|p - p^n\|_2^2.$$

Let $h = \delta_P$, then the update rule for p is

$$p^{n+1} = \operatorname{prox}_{t_p \cdot h}(p^n + t_p \cdot \nabla \bar{u}^n) = \frac{p^n + t_p \cdot \nabla u^n}{\max(1, |p^n + t_p \cdot \nabla \bar{u}^n|)}.$$

B. Update $\lambda_{\mathcal{D}}$: Because of the separable structure of the second optimization problem in (17), we can update individually each component λ_Ψ of $\lambda_{\mathcal{D}}$ by solving the corresponding minimization problem

$$\lambda_\Psi^{n+1} = \arg\min_{\lambda_\Psi} -\lambda_\Psi \left(\langle A\bar{u}^n - g, \Psi \rangle - \tau \right) + \delta_Q(\lambda_\Psi) + \frac{1}{2t_\lambda} \left(\lambda_\Psi - \lambda_\Psi^n \right)^2.$$

Here $h=\delta_Q$ so the update rule for λ_Ψ is

$$\begin{aligned}{\lambda_\Psi}^{n+1} &= \operatorname{prox}_{t_\lambda\cdot h}\Big(\lambda_\Psi^n + t_\lambda\cdot\left(\langle A\bar{u}^n - g,\Psi\rangle - \tau\right)\Big)\\ &= \max\Big(\lambda_\Psi^n + t_\lambda\cdot\left(\langle A\bar{u}^n - g,\Psi\rangle - \tau\right),0\Big).\end{aligned}$$

C. Update u: For this sub-problem, $h=\frac{\lambda}{2}\left\|Au-g\right\|_2^2$, so the update rule for u is

$$u^{n+1} = \operatorname{prox}_{t_u\cdot h}(\tilde{u}) = (I+t_u\lambda A^*A)^{-1}(\tilde{u}+t_u\lambda A^*g), \tag{18}$$

where $\tilde{u} = u^n - t_u\cdot\left(-\operatorname{div} p^{n+1} + A^*\sum\limits_{\Psi\in\mathcal{D}}\lambda_\Psi\Psi\right)$. We use a multi-step procedure to accelerate the convergence, which results in Algorithm 2 below.

The following theorem gives the convergence rate of Algorithm 2.

Theorem 5.1. *Choose $t_u>0$, $t_\lambda>0$ and $t_p>0$ such that $t_\lambda=t_p$. Consider the sequences $\{u^n\}$, $\{p^n\}$ and $\{\lambda_{\mathcal{D}}^n\}$ generated by Algorithm 2. Let $u_N=\frac{1}{N}\sum\limits_{n=1}^{N}u^n$, $p_N=\frac{1}{N}\sum\limits_{n=1}^{N}p^n$, $\lambda_{\mathcal{D},N}=\frac{1}{N}\sum\limits_{n=1}^{N}\lambda_{\mathcal{D}}^n$ and the restricted primal-dual gap*

$$\mathcal{G}(u,p,\lambda_{\mathcal{D}}) = \max_{p',\lambda'_{\mathcal{D}}} L(u,p',\lambda'_{\mathcal{D}}) - \min_{u'} L(u',p,\lambda_{\mathcal{D}}). \tag{19}$$

If t_pt_u is small enough, then there exists a constant $C>0$ such that for all N,

$$\mathcal{G}(u_N,p_N,\lambda_{\mathcal{D},N}) \le \frac{C}{N}.$$

One can easily prove this result by following the proof of Theorem 1 in [15]. In the present case, the operator K that permits to write (16) under the general form (14) satisfies

$$\|Ku\|^2 = \|\nabla u\|^2 + \sum_{\Psi\in\mathcal{D}}\langle Au,\Psi\rangle^2,$$

so that $\|K\|^2\le\|\nabla\|^2+\|A\|^2\|\mathcal{D}\|^2$ with $\|\mathcal{D}\|^2=\sup\limits_{\|u\|\le 1}\sum\limits_{\Psi\in\mathcal{D}}\langle u,\Psi\rangle^2$ and $\|\nabla\|^2\le 8$ (see [13]). Thus, the announced property holds as soon as $t_pt_u(8+\|A\|^2\|\mathcal{D}\|^2)<1$.

Algorithm 2 The proposed algorithm for the unconstrained model

```
**Initialization**: $\bar{u}^0 = u^0$, $p = 0$, $\lambda_{\mathcal{D}} = 0$, $t_p$, $t_\lambda$, $t_u$.
**Output**: $u^N$.
**Iterate for** $n = 0, 1, \ldots N-1$:
    Step 1. $p^{n+1} = \dfrac{p^n + t_p\cdot\nabla\bar{u}^n}{\max(1, |p^n + t_p\cdot\nabla\bar{u}^n|)}$
    Step 2. $\forall\Psi\in\mathcal{D}$,  $\lambda_\Psi^{n+1} = \max\Big(\lambda_\Psi^n + t_\lambda\cdot\left(\langle A\bar{u}^n - g,\Psi\rangle - \tau\right),\ 0\Big)$
    Step 3. $\tilde{u} = u^n - t_u\cdot\left(-\operatorname{div} p^{n+1} + A^*\sum_{\Psi\in\mathcal{D}}\lambda_\Psi^{n+1}\Psi\right)$
    Step 4. $u^{n+1} = (I + t_u\lambda A^*A)^{-1}(\tilde{u} + t_u\lambda A^*g)$
    Step 5. $\bar{u}^{n+1} = 2u^{n+1} - u^n$
```

5.3. Algorithm for solving the constrained formulation. In this subsection, we propose an algorithm to solve directly the constrained optimization problem (15) instead of introducing the Lagrange multipliers $(\lambda_\Psi)_{\Psi\in\mathcal{D}}$. Model (15) can be written as

$$\min_{u\in E_\tau}\max_{p} - \langle u, \operatorname{div} p\rangle + \frac{\lambda}{2}\left\|Au-g\right\|_2^2 - \delta_P(p), \tag{20}$$

where $E_\tau = \{u \in E;\ \forall\Psi\in\mathcal{D},\ \langle Au-g,\Psi\rangle - \tau \le 0\}$. Minimizing the right-hand term of (20) with respect to u, we get the following sub-problem:

$$\min_{u\in E_\tau} - \langle u, \operatorname{div} p\rangle + \frac{\lambda}{2}\left\|Au-g\right\|_2^2. \tag{21}$$

It is equivalent to

$$\min_{u\in E} - \langle u, \operatorname{div} p\rangle + \frac{\lambda}{2}\left\|Au-g\right\|_2^2 + \delta_{E_\tau}(u). \tag{22}$$

With $h(u) = \frac{\lambda}{2}\left\|Au-f\right\|_2^2$, the update rule for u writes

$$u^{n+1} = \operatorname{prox}_{h+\delta_{E_\tau}}\left(u^n - t_u\cdot\left(-\operatorname{div} p^{n+1}\right)\right),$$

and using Proposition 2, we obtain

$$u^{n+1} = P_{E_\tau}\left(\operatorname{prox}_h\left(u^n - t_u\cdot\left(-\operatorname{div} p^{n+1}\right)\right)\right).$$

Unfortunately, the projection P_{E_τ} is not easy to compute when A is not the identity matrix. This is why we propose to use this operator only for the denoising case ($A = I$), which yields Algorithm 3 below. Note that in the case when $\mathcal{D}$ is made of orthogonal vectors (e.g. an orthogonal wavelet basis) plus their opposites, P_{E_τ} can be computed by

$$P_{E_\tau}(u) = g + \sum_{\Psi\in\mathcal{D}} P_{[0,\tau]}\left(\langle u-g,\Psi\rangle\right)\Psi,$$

where $P_{[0,\tau]}(t) = \min(\tau, \max(0,t))$.

Remark 2. There are many works about the projection onto a convex set. For example, we could compute P_{E_τ} by the method proposed in [7], but the computational cost would be much higher than with Algorithm 2. This is why we systematically use Algorithm 2 when $A \ne I$. When $A = I$, it is better to use Algorithm 3 since it exactly enforces the constraints of the model at each iteration, while showing the same convergence rate as Algorithm 2 (as will be shown on Fig. 3).

Algorithm 3 The proposed algorithm for the constrained model when $A = I$

```
Initialization: $\bar{u}^0 = u^0$, $p = 0$, $t_p$, $t_u$.
Output: $u^N$.
Iterate for $n = 0, 1, \dots N-1$:
    Step 1. $p^{n+1} = \frac{p^n + t_p\cdot\nabla\bar{u}^n}{\max\left(1, |p^n + t_p\cdot\nabla\bar{u}^n|\right)}$
    Step 2. $\tilde{u} = u^n - t_u\cdot\left(-\operatorname{div} p^{n+1}\right)$
    Step 3. $u^{n+1} = P_{E_\tau}\left(\frac{\tilde{u} + t_u\lambda g}{1 + t_u\lambda}\right)$
    Step 4. $\bar{u}^{n+1} = 2u^{n+1} - u^n$
```

6. APPLICATIONS AND EXPERIMENTS

In this section, we use the proposed algorithms in three image applications, namely denoising, deblurring and simultaneous deblurring and inpainting. These problems can be addressed with TV regularization (see, e.g., [1, 51] for deblurring, [3] for inpainting, [17] for simultaneous deblurring and inpainting), hence we will be able in each case to compare the performance of the ROF model (solved by Algorithm 1), the LM model, and our model. For the LM model and our model, we use a dictionary made of wavelet packets (see details below). Note that the ROF model has a much lower computational cost since it does not involve wavelet packet decomposition/reconstruction as in the two other models. Most of the experiments we discuss are realized with the two gray-level images (with size 256×256) shown in Fig. 1.

FIGURE 1. Original images used for image restoration. The left image is used for denoising, and the right one is used for deblurring and inpainting. In both cases (as well as for the other images considered later in Fig. 5), the pixel values are in the range $[0, 255]$.

Designing a dictionary for a given class of images is still an open problem. Therefore, in each case we used the dictionary obtained by the full wavelet packet decomposition up to level four using the freeware Wavelab850[1] running under MATLAB. This choice seems to provide an acceptable representation for most images. The choice of the wavelet filter also affects the performances of the model. A large compact support can achieve better performance, because a large compact support in "time" allows more narrowness in "frequency", which is usually more efficient for denoising. However, a very large compact support would introduce a heavy computational cost, so we chose the "sym8" filter to achieve a reasonable trade-off.

The peak signal-to-noise ratio (PSNR) was used to measure the quality of the recovered images. It is defined (in dB) by

$$\mathrm{PSNR}(\hat{u}, u) = 10 \log_{10} \frac{255^2}{\frac{1}{256\times 256} \sum_{ij} |\hat{u}_{ij} - u_{ij}|^2},$$

where $\hat{u}$ denotes the restored image and u is the ground-truth image. As a stopping criterion we used the relative difference between two successive iterates of the

---

[1] This software is available on http://www-stat.stanford.edu/∼wavelab

restoration scheme (RDS) which is defined by

$$\mathrm{RDS}\left(u^{n+1}, u^{n}\right) = \frac{\left\|u^{n+1} - u^{n}\right\|_2}{\left\|u^{n+1}\right\|_2}.$$

We stop the algorithm when $\mathrm{RDS}\left(u^{n+1}, u^{n}\right) < \varepsilon$. We used $\varepsilon = 10^{-6}$ for denoising, and $\varepsilon = 10^{-5}$ in all other experiments (which do not converge as fast as the denoising case). When specified, the "number of iterations" or "CPU time" of an algorithm means the number of iterations or the CPU time that was needed to drop the RDS below the error tolerance ε.

It is not easy to tune the parameters for the LM-Uzawa algorithm. A steepest gradient descent was used to compute the minimization for u in [29], with the bisection method to search a suitable time step at each iteration (it usually requires about 30 sub-iterations to find the time step). We used the same algorithm, but significantly accelerated the code to avoid unnecessarily repeated computations. We observed that with this acceleration, the CPU time for one iteration was approximately the same for the LM-Uzawa and the proposed algorithms, but as we shall see later the convergence of the LM-Uzawa algorithm remains significantly slower.

The choice of ρ_0 (the initial time step for updating $\lambda_{\mathcal{D}}$, see [29] for details) also has a strong impact on the performance of the LM-Uzawa algorithm. Large ρ_0 results in a good control of the constraints, but may also cause bad convergence behavior. The authors pointed out that one can adjust ρ_0 by ensuring that the dual function does not decrease significantly. Unfortunately, this is also not easy to control.

The parameter in $TV_\beta(u)$ for the LM model was set to be $\beta = 10^{-5}$. In all the following experiments, we used the same constant time step settings: $t_u = 1$, $t_p = 0.1$, $t_\lambda = 0.1$ for the proposed model, and the experiments demonstrate that this simple choice yields good performances.

6.1. Image denoising. The proposed method can be easily applied to image denoising. In this case we have $A = I$, $A^* = I$ and $A^*A = I$. We give two examples: gray-level image denoising and color image denoising. Symmetric boundary conditions are used in this section.

The first example is gray-level image denoising. The original image is corrupted by an additive white Gaussian noise with zero mean and standard deviation $\sigma = 20$. We set the threshold $\tau = 75$ for the LM model and the proposed model, $\rho_0 = 20$ for the LM model, $\lambda = 0.04$ for the ROF model, and $\lambda = 0.03$ for the proposed model. Fig. 2 displays the noisy image and the restored images. We can observe that the LM model and the proposed model preserve textures much better than the ROF model. As expected, this indicates that the wavelet packet approach is very helpful to recover textures. Furthermore, the proposed model converges much faster (for the required precision) than the LM model. Note that the CPU time for one iteration of LM-Uzawa algorithm is 0.98s, and that of the proposed algorithm is 0.86s, so the large speed difference is essentially due to the faster convergence rate of the proposed method (281 iterations to meet the RDS criterion, versus 1456 for the LM-Uzawa model).

We first discuss the convergence behavior of the LM model and the proposed model. We get a solution u^* for the LM model by running the LM-Uzawa algorithm for 20,000 iterations, and get a solution u^* for the proposed model by running Algorithm 3 for 20,000 iterations (since Algorithm 3 can exactly achieve the constraints of the proposed model at each iteration, its solution can be seen as the

FIGURE 2. Denoising of a gray-level image. Top, left: noisy image; Top, right: ROF Model (787 iterations, CPU time = 3.9s, PSNR = 24.75dB); Bottom, left: LM Model (1456 iterations, CPU time = 1434s, PSNR = 26.09dB); Bottom, right: our model (281 iterations, CPU time = 241s, PSNR = 26.19dB).

true solution of the proposed model). Fig. 3 gives some comparisons with respect to the CPU time. The RDS of the LM-Uzawa algorithm is not very stable, so we stopped the algorithm when the RDS was less than ε for the first time. Table 1 gives the statistics of the restored images obtained from running the algorithms for 5000 iterations. In particular, we compute the *Maximum of the violated constraints*

$$\text{Max} = \max_{\Psi \in \mathcal{D}} \left[\langle u - g, \Psi \rangle - \tau \right]^+ , \tag{23}$$

(with as usual $[t]^+ = \max(0, t)$), the *Proportion of violated constraints*

$$\text{Pro.} = \frac{|V|}{|\mathcal{D}|}, \quad \text{where} \quad V = \left\{ \Psi \in \mathcal{D},\, \langle u - g, \Psi \rangle - \tau > 10^{-5} \right\}, \tag{24}$$

and, when $V \neq \emptyset$, the *Mean of the violated constraints*

$$\text{Mean} = \frac{1}{|V|} \sum_{\Psi \in V} \langle u - g, \Psi \rangle - \tau. \tag{25}$$

One can clearly see from Fig. 3 and Table 1 that the proposed model is more stable than the LM model, and that the proposed algorithms (2 and 3) manage to

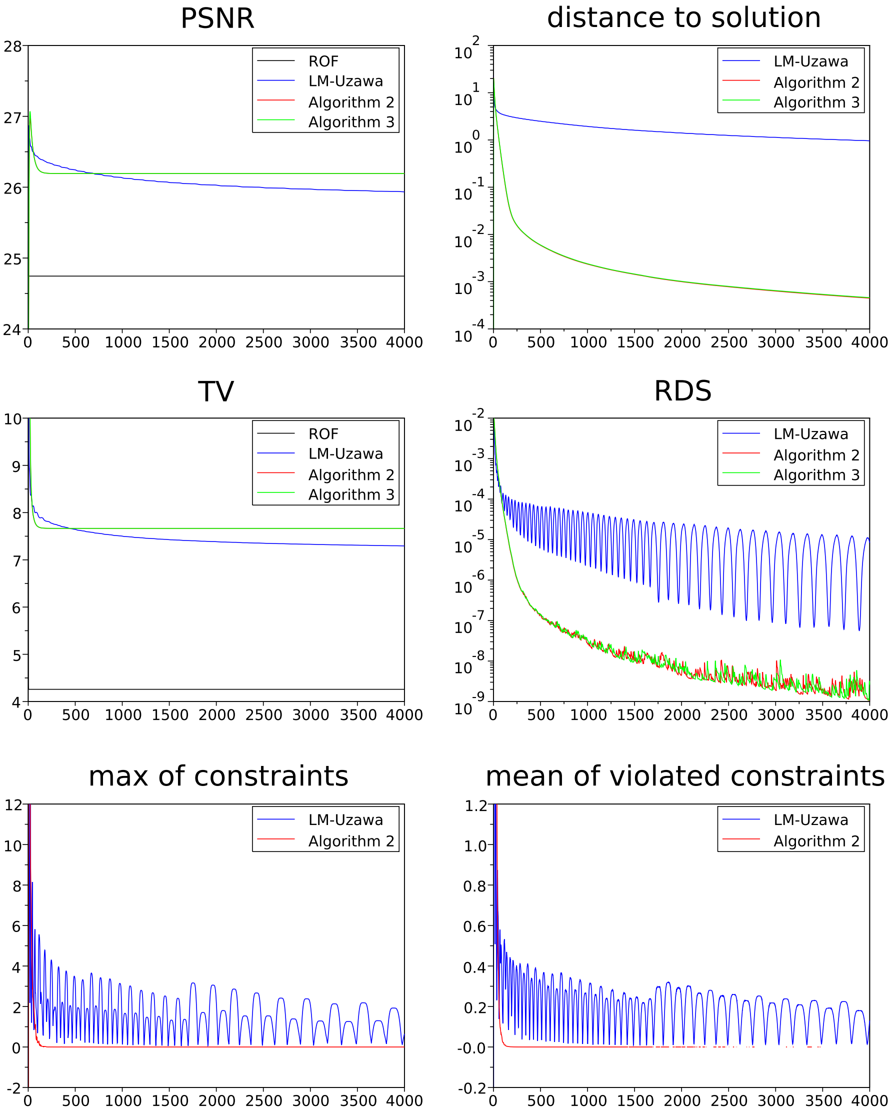

Figure 3. Evolution in function of the CPU time (in seconds) of several quantities during gray-level denoising with ROF, LM-Uzawa, and the proposed Algorithms 2 and 3 (for which the curves often overlap). From left to right, then top to bottom: PSNR; distance to solution $\|u^n - u^*\|_2$ (log scale); TV; RDS (log scale); Maximum value of the constraints (Equation 23); Mean value of the violated constraints only (Equation 25). Notice that after 2500s, the set of violated constraints for Algorithm 2 is empty most of the time, which explains the dotted line.

solve it precisely and much faster than the LM-Uzawa algorithm does for the LM model. Algorithm 2, which corresponds to the unconstrained formulation, exhibits

	TV	Max	Mean	Pro.
LM-Uzawa	7.2722	1.9385	0.1658	0.40%
Algorithm 2	7.6661	0.0000	-	0%
Algorithm 3	7.6661	0.0000	-	0%

TABLE 1. Experimental statistics of the result images obtained after running the algorithms for 5000 iterations (denoising case). Max: Maximum of the constraints (Equation 23). Mean: Mean of the violated constraints only (Equation 25). Pro.: proportion of violated constraints (Equation 24).

very good performances in terms of constraint enforcement: the maximum of the constraints for Algorithm 2 decreases quickly toward 0, and after 2500s, no more constraint is violated most of the time. In comparison, the LM-Uzawa algorithm clearly does not enforce the constraints as well as the proposed algorithm.

Next, we discuss the choice of the parameter λ weighting the data-fidelity term in the ROF and in the proposed model. It is well-known that the performance of the ROF model is highly dependent on this parameter. Table 2 lists the PSNR values of the restored images obtained for different values of λ in the denoising case (Fig. 2) and in the Gaussian deblurring case (Fig. 7). We also present some of the corresponding resulting images in Fig. 4 for the denoising problem. From these Table and Figures, one can clearly observe that, unlike the ROF model, the performance of the proposed model is not very sensitive to the parameter λ. This indicates that the wavelet-based term can restore much information, by contributing to a trade-off with the TV regularization.

	denoising				deblurring			
λ	0.02	0.03	0.04	0.05	2	3	4	5
ROF model	23.59	24.15	24.75	25.38	24.68	24.92	25.22	25.52
Proposed model	25.69	26.19	26.57	26.85	27.10	27.27	27.37	27.43

TABLE 2. PSNR values (dB) of the restored images obtained with the ROF model and the proposed model for different values of λ (the reference images are shown in Fig. 1).

To illustrate further the general behavior of the proposed model, we perform experiments on four other test images, shown in Fig. 5: Boat (size 512 × 512), Cameraman (256×256), Goldhill (512×512), and Lena (512×512). The degradation process is the same as before. Table 3 presents the resulting PSNR values. We can observe that the LM model and our model do not reach much better results than the ROF model for these relatively poorly textured images. In fact, the LM model (and consequently ours), with the standard dictionary used, are in favor of textured images. Note incidentally that the proposed model achieves slightly higher PSNR values than the LM model. To further improve the performance of the LM model and our model on images with less texture, one may consider choosing a better dictionary, as indicated in [49].

The proposed model can also be easily generalized to color image denoising. We adopt the scheme proposed by Bresson and Chan [9] to compute the dual formulation of the vectorial TV regularization. The top-left image in Fig. 6 is the original

Figure 4. Image denoising with different values of λ (top line: $\lambda = 0.02$, bottom line: $\lambda = 0.05$). Left column: ROF Model; Right column: our model.

Figure 5. Other images. From left to right: Boat, Cameraman, Goldhill, Lena.

color image with resolution $256 \times 256 \times 3$ (3 is the number of color channels). The noisy image is obtained by adding independent noise (with the same Gaussian distribution as before) on each color channel. On Fig. 6, we compare the proposed model with the vectorial ROF model, solved by Bresson and Chan [9][2] via a dual minimization algorithm. We can observe that the proposed method clearly achieves a better denoising, which is confirmed by the higher PSNR value.

---

[2] The MATLAB code is available on `http://www.cs.cityu.edu.hk/~xbresson/ucla/code.html`

		$\tau = 75$		$\tau = 70$		$\tau = 65$	
Images	ROF	LM	Ours	LM	Ours	LM	Ours
Boat	28.61	28.51	28.72	28.58	28.76	28.66	28.81
Cameraman	28.21	27.99	28.36	28.05	28.41	28.12	28.46
Goldhill	29.06	29.03	29.16	29.08	29.19	29.15	29.23
Lena	31.04	31.03	31.05	31.06	31.06	31.11	31.06
Average	29.23	29.14	29.32	29.19	29.36	29.26	29.39

TABLE 3. The resulting PSNR(dB) on the test images of Fig. 5.

FIGURE 6. Color image denoising. Top, left: original color image; Top, right: noisy image; Bottom, left: vectorial ROF model (480 iterations, CPU time=10.7s, PSNR=26.54dB); Bottom, right: Proposed model (238 iterations, CPU time=626.2s, PSNR=26.92dB).

6.2. IMAGE DEBLURRING. In this section, we apply the proposed model to a more difficult task: image deblurring. We will consider three blur kernels: a Gaussian blur kernel, a uniform blur kernel and a motion blur kernel. In these experiments, periodic boundary conditions are adopted and the FFT is used to compute the convolution operator. The initial images of the algorithms are the blurred and noisy images. We set $\tau = 5$ for the first two experiments, $\tau = 2$ for the last

Figure 7. Restoration from Gaussian blur. Top, left: blurred image; Top, right: ROF model (452 iterations, CPU time=4.0s, PSNR=25.52dB); Bottom, left: LM model (1602 iterations, CPU time=1639s, PSNR=27.16dB); Bottom, right: Proposed model (966 iterations, CPU time=861s, PSNR= 27.43dB).

experiment since the degraded image has less noise, and $\rho_0 = 5$ for the LM model in all the experiments.

In the first experiment, the corrupted image is obtained by convolving the original image with a Gaussian blur of size 5 × 5 with mean zero and standard deviation 2 (produced by the MATLAB function ”fspecial”) and then adding a Gaussian white noise with mean zero and standard deviation 2. We set $\lambda = 5$ for the ROF model and the proposed model. Fig. 7 displays the restored images.

Then, we choose a uniform 7 × 7 blur to corrupt the original image, and add as before a Gaussian white noise with mean zero and standard deviation 2 to it. Again, the parameter λ is set to 5. The restored images are shown in Fig. 8.

Finally, we apply the proposed algorithm to a motion-blurred image. The kernel is generated by the MATLAB function ”fspecial” with approximately 30 pixels and an angle of 45°. Then the blurred image is corrupted with an additive white Gaussian noise with mean zero and standard deviation 1. With the parameter $\lambda = 20$, the restored images are shown in Fig. 9.

Fig. 10 shows the resulting RDS of the three experiments mentioned above. One can observe that the proposed algorithm achieves better performance than the

FIGURE 8. Restoration from uniform blur. Top, left: blurred image; Top, right: ROF model (522 iterations, CPU time=4.6s, PSNR=24.73dB); Bottom, left: LM model (1978 iterations, CPU time=1991s, PSNR=25.49dB); Bottom, right: Proposed model (1083 iterations, CPU time=968s, PSNR= 25.75dB).

LM-Uzawa algorithm. Table 4 gives some statistics of the restored images obtained after running the algorithms for 5000 iterations. The Max, Mean and Pro. statistics correspond to the definitions given in Equations (23), (25) and (24), except that u is replaced with Au to take into account the blur operator A. Although the maximum of the violated constraints is a little larger than that of the LM-Uzawa algorithm in the case of a uniform blur kernel, the proportion of violated constraints is always much smaller for the proposed model.

6.3. SIMULTANEOUS IMAGE DEBLURRING AND INPAINTING. In several image processing applications, some entire image regions are damaged or lost. The goal of *image inpainting* is to restore the missing and/or damaged areas using the valid surrounding information. Since the pioneering work of [33] and [5], this problem has been extensively studied [11]. In this subsection, we consider a more complicated image degradation problem: images are blurred and pixels are missing in some regions.

Figure 9. Restoration from motion blur. Top, left: blurred image; Top, right: ROF model (806 iterations, CPU time=8.1s, PSNR=25.58dB); Bottom, left: LM model (3348 iterations, CPU time=3402s, PSNR=26.21dB); Bottom, right: Proposed model (1546 iterations, CPU time=1382s, PSNR= 27.14dB).

	Algorithms	TV	Max	Pro.	Mean
Gaussian blur	LM-Uzawa	10.0958	7.9479	3.93%	0.0038
$\tau = 5$	Algorithm 2	11.2513	7.9154	0.90%	0.0027
Uniform blur	LM-Uzawa	8.8571	8.0361	2.27%	0.0031
$\tau = 5$	Algorithm 2	9.3452	8.0533	0.96%	0.0029
Motion blur	LM-Uzawa	11.7072	3.8542	5.79%	0.0038
$\tau = 2$	Algorithm 2	12.7043	3.0598	2.50%	0.0024

Table 4. Experimental statistics obtained for the deblurring case after 5000 iterations: Total Variation, maximum of the violated constraints, proportion of violated constraints, mean of the violated constraints (see Equations (23) to (25) with u replaced by Au).

Let Ω_1 denote the set of indices corresponding to missing pixels. The linear transform operator for inpainting is defined as

$$(A_2u)_{i,j} = \begin{cases} u_{i,j}, & \text{if} \quad u_{i,j} \notin \Omega_1, \\ 0, & \text{if} \quad u_{i,j} \in \Omega_1. \end{cases}$$

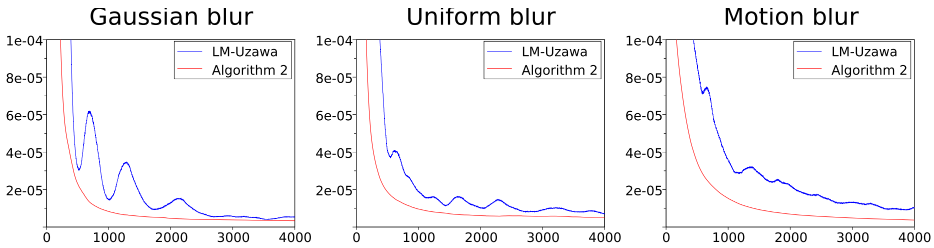

FIGURE 10. RDS in function of CPU time (in seconds) for the deblurring case. From left to right: Gaussian blur, uniform blur, motion blur.

FIGURE 11. Image inpainting with randomly missing pixels. Top, left: corrupted image; Top, right: ROF model (993 iterations, CPU time=142s, PSNR=24.29dB); Bottom, left: LM model (2449 iterations, CPU time=2477s, PSNR=24.63dB); Bottom, right: Proposed model (1856 iterations, CPU time=2084s, PSNR=25.06dB).

Let A_1 denote the convolution operator. Then the degraded operator considered in this subsection is $A = A_2 \circ A_1$. In this case, because the operator $A_1^* \circ A_2 \circ A_1$ can not be diagonalized by the FFT, the update step 4 in Algorithm 2 can not be solved as above. Fortunately, thanks to the I term, the matrix $I + t_u \lambda A^* A$ is always

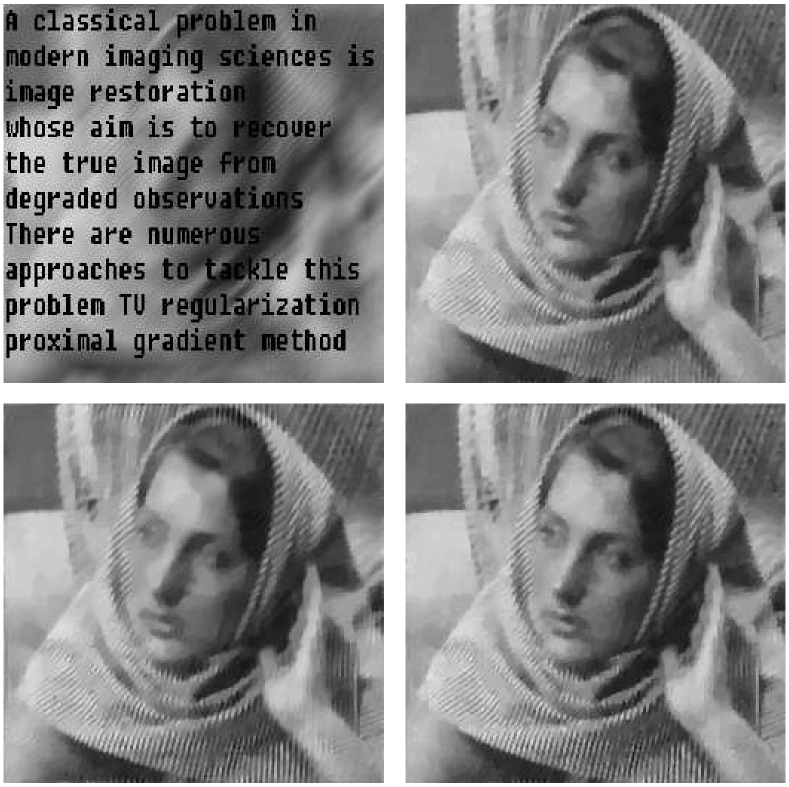

Figure 12. Image inpainting with structured missing pixels (superimposed text). Top, left: corrupted image; Top, right: ROF model (925 iterations, CPU time=151s, PSNR=25.06dB); Bottom, left: LM model (2721 iterations, CPU time=2803s, PSNR=25.30dB); Bottom, right: Proposed model (1388 iterations, CPU time=1546s, PSNR=26.20dB).

invertible. Thus we can use the preconditioned conjugate gradient method to solve this sub-problem.

We study two situations for missing pixels: randomly missing pixels and structured missing regions. In these experiments, we use the same parameter values as in the motion deblurring experiments, except for the randomly missing pixels case for which we take $\tau = 1.2$ (because there are lots of missing pixels, the coefficients of the wavelet packet decomposition are smaller). To accelerate the convergence of the algorithms, we use a initial image obtained by performing the nearest neighbor interpolation on the corrupted image.

In the first experiment, the corrupted image is obtained by randomly removing 50% pixels on the motion-blurred image used in the previous section. On Fig. 11, we can see that the visual quality of the image restored by the proposed model is better. In the second experiment, we still use the same motion-blurred image but the missing pixels now correspond to a superimposed text. The results are displayed on Fig. 12. Table 5 gives some statistics of the restored images obtained

	Algorithms	TV	Max	Pro.	Mean
random pixels inpainting	LM-Uzawa	10.3346	1.4823	7.13%	0.0031
$\tau = 1.2$	Algorithm 2	10.3569	1.5168	5.17%	0.0023
text inpainting	LM-Uzawa	10.0482	2.7645	4.57%	0.0025
$\tau = 2$	Algorithm 2	11.1518	2.9219	1.79%	0.0013

TABLE 5. Experimental statistics of the resulting images obtained after running the algorithms for 5000 iterations (inpainting case).

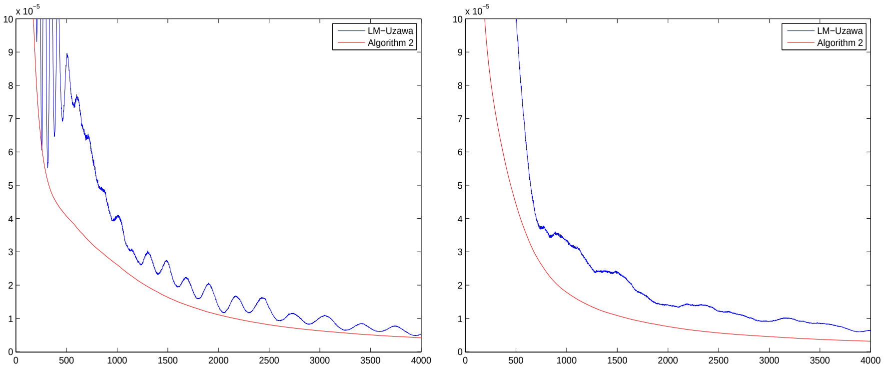

FIGURE 13. RDS in function of CPU time (in seconds) for the inpainting case. Left: randomly missing pixels; Right: structured (text) missing regions.

by running the algorithms for 5000 iterations, and Fig. 13 shows the resulting RDS. One can observe that the proposed model still exhibits better performances on the simultaneous deblurring and inpainting problem.

7. COMBINING NONLOCAL TOTAL VARIATION (NLTV)

Recently, much literature has focused on exploiting nonlocal similarities of textures and has reached higher restoration quality than methods only based on the similarity between pixels. Buades et al. [10] proposed the nonlocal means algorithm, which exhibits good performances for image denoising. Gilboa and Osher [27] proposed the NLTV approach, based on a variational framework. Zhang et al. [50] used this regularization for image deconvolution and sparse reconstruction.

We use the definitions of the NL functionals presented in [8, 27, 50]. The general NLTV framework for image restoration can be defined as

$$\min_u \|\nabla_w u\|_1 + \frac{\lambda}{2}\|Au - g\|_2^2, \tag{26}$$

where $(\nabla_w u)(x, y) = (u(y) - u(x))\sqrt{w(x, y)}$. The weight function w is a symmetric function that can be defined from the initial image g and h as

$$w(x, y) = \exp\left(-\frac{1}{h^2}\int_\Omega G_a * \|g(x + z) - g(y + z)\|_2^2\, dz\right),$$

where $\int_\Omega G_a * \|g(x + z) - g(y + z)\|_2^2\, dz$ is the distance between patches located at x and y on the given degraded image, G_a is a Gaussian function with standard

Figure 14. Image denoising based on NLTV regularization. Left: NLTV restoration model (PSNR=28.11dB); Right: Proposed model (PSNR=28.68dB).

deviation a and h is a filtering parameter that has to be set according to the noise level. The divergence of the dual variable is defined by

$$\left(\operatorname{div}_w p\right)(x) = \int_{\Omega} \left(p(x,y) - p(y,x)\right) \sqrt{w(x,y)}dy,$$

and we have

$$\langle \nabla_w u, p \rangle = - \langle u, \operatorname{div}_w p \rangle .$$

We can easily replace TV regularization by the NLTV regularization in the proposed algorithm. In the following, we give some experimental results of the proposed algorithm and the NLTV restoration model which can be solved by Algorithm 1.

To compute the weights (the values of the all pixels of the degraded image are scaled in the range $[0,1]$ when computing the weights), the MATLAB code provided by Bresson[3] was used. We set the patch size to 9×9, the searching window size to 11×11 and the number of neighbors to 10. The filtering parameter was set to $h = 0.45$ for image denoising, $h = 0.01$ for restoration from Gaussian blur and uniform blur, and $h = 0.005$ for motion deblurring. Since NLTV regularization uses more neighbors than TV regularization, the parameter λ has to be increased, so we took $\lambda = 0.05$ for denoising, $\lambda = 10$ for restoration from Gaussian blur and uniform blur, and $\lambda = 30$ for motion deblurring. We included the 4 nearest neighbors in image deblurring cases. Fig. 14 and Fig. 15 display the recovered images, and show the clear improvement brought by NLTV regularization.

8. Conclusion

We presented a modified model for image restoration based on Total Variation and wavelet packets. In this model, the original TV regularization term is replaced by its dual formulation, and a first-order primal-dual algorithm is used to solve the corresponding optimization problems. The performance of the proposed model is demonstrated by experiments on several standard image restoration problems (denoising, deblurring and inpainting), and the results show that it manages to recover textures better than previous comparable approaches. To further improve

---

[3] http://www.cs.cityu.edu.hk/∼xbresson/ucla/code.html

FIGURE 15. Results of image deblurring based on NLTV regularization. Left column: classical NLTV deblurring; Right column: proposed model combined with NLTV. From top to bottom: Gaussian blur (PSNR=26.44dB, 28.11dB), uniform blur (PSNR=25.48dB, 26.16dB), and motion blur (PSNR=25.79dB, 27.50dB).

the performance of the proposed model, we then replaced TV regularization with NLTV regularization, which resulted in a clear improvement of the quality of the restored images. Future work could focus on the choice of an adaptive dictionary to increase further the performance of the proposed model.

References

[1] M. Afonso, J. Bioucas-Dias and M. Figueiredo, *Fast image recovery using variable splitting and constrained optimization*, IEEE Trans. Image Process., **19** (2010), 2345–2356.

[2] L. Ambrosio, N. Fusco and D. Pallara, “Functions of bounded variation and free discontinuity problem,” Oxford, U.K.: Oxford Univ. Press (2000).

[3] G. Aubert and P. Kornprobst, “Mathematical problems in image processing: partial differential equations and the calculus of variations,” Applied Mathematical Sciences, 2nd ed. New York: Springer, vol. 147 (2006).

[4] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imag. Sci., **2** (2009), 183–202.

[5] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, *Image inpainting*, in Proc. SIGGRAPH, New York, (2000), 417–424.

[6] S. Boyd and L. Vandenberghe, “Convex Optimization,” Cambridge University Press (2004).

[7] J.P. Boyle and R.L. Dykstra, *A method for finding projections onto the intersection of convex sets in Hilbert spaces*, Lecture Notes in Statistics, **37** (1986), 28–47.

[8] X. Bresson, *A short note for nonlocal TV minimization*, Technical Report (2009).

[9] X. Bresson and T. F. Chan, *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, Inverse Problems and Imaging, **2** (2008), 455–484.

[10] A. Buades, B. Coll and J.M. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Model. Simul., **4** (2005), 490–530.

[11] A. Bugeau, M. Bertalmio, V. Caselles and G. Sapiro, *A Comprehensive Framework for Image Inpainting*, IEEE Trans. Image Process., **19** (2010), 2634–2645.

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

[13] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imag. Vis., **20**(2004), 89–97.

[14] A. Chambolle, R. DeVore, N.-Y. Lee and B. Lucier, *Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage*, IEEE Trans. Image Process., **7**(1998), 319–335.

[15] A. Chambolle and T. Pock, *A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging*, J. Math. Imaging Vis., **40** (2011), 120–145.

[16] T. F. Chan, G. H. Golub and P. Mulet, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comp., **20** (1999), 1964–1977.

[17] T. F. Chan, A. M. Yip and F. E. Park, *Simultaneous total variation image inpainting and blind deconvolution*, Int. J. of Imaging Systems and Technology **15:1** (2005), 92–102.

[18] T. Chan and J. Shen, “Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods,” SIAM (2005).

[19] C. Chaux, J.C. Pesquet and N. Pustelnik, *Nested iterative algorithms for convex constrained image recovery problems*, SIAM J. Imaging Sciences, **2** (2009), 730–762.

[20] P. L. Combettes and V. R.Wajs, *Signal recovery by proximal forward backward splitting*, Multiscale Modeling & Simulation, **4** (2005), 1168–1200.

[21] I. Daubechies, “Ten Lectures on Wavelets,” SIAM Publ., Philadelphia, 1992.

[22] I. Daubechies, M. Defriese and C.D. Mol, *An iterative thresholding algorithm for linear inverse problems with a sparsity constraint*, Commun.Pure Appl. Math., **57** (2004), 1413–1457.

[23] D.L. Donoho and I.M. Johnstone, *Ideal spatial adaptation by wavelet shrinkage*, Biometrika, **81** (1994), 425–455.

[24] D.L. Donoho and I.M. Johnstone, *Adapting to unknown smoothness via wavelet shrinkage*, J. Amer. Statist. Assoc., **90** (1995), 1200-1224.

[25] I. Ekeland and R. Temam, “Convex Analysis and Variational Problems,” Studies Math. Appl., American Elsevier, Amsterdam, New York, 1976.

[26] D. Gabay, *Applications of the method of multipliers to variational inequalities*, in “Augmented Lagrangian Methods: Applications to the Solution of Boundary-Valued Problems”, M. Fortin and R. Glowinski, eds., North-Holland, Amsterdam, 1983, 299–331.

[27] G. Gilboa and S. Osher, *Nonlocal operators with applications to image processing*, Multiscale Model. Simul., **7** (2008), 1005–1028.

[28] T. Goldstein and S. Osher, *The split Bregman method for l1 regularized problems*, SIAM J. Imaging Sciences, **2** (2009), 323–343.

[29] S. Lintner and F. Malgouyres, *Solving a variational image restoration model which involves contraints*, Inverse. Probl., **20** (2004), 815–831.

[30] F. Malgouyres, *A framework for image deblurring using wavelet packet bases*, Appl. and Comp. Harmonic Analysis, **12** (2002), 309–331.

[31] F. Malgouyres, *Mathematical analysis of a model which combines total variation and wavelets for image restoration*, Journal of Information Processes, **2** (2002), 1–10.

[32] F. Malgouyres, *Minimizing the total variation under a general convex constraint for image restoration*, IEEE trans. on Image Process., **11** (2002), 1450–1456.

[33] S. Masnou and J.-M. Morel, *Level lines based disocclusion*, Int. Conf. on Image Processing (2008), 259–263.

[34] C. A. Micchelli, L. Shen and Y. Xu, *Proximity Algorithms for Image Models: Denoising*, Inverse Probl., **27:4** (2011), 45009–45038.

[35] J.-J. Moreau, *Fonctions convexes duales et points proximaux dans un espace hilbertien*, C.R. Acad. Sci. Paris Ser. A Math, **255** (1962), 2897-2899.

[36] J.-J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France, **93** (1965), 273–299.

[37] Y. Nesterov, *A method of solving a convex programming problem with convergence rate* $O\left(1/k^2\right)$, Soviet Mathematics Doklady, **27** (1983), 372–376.

[38] M. Ng, W. Fan and X. Yuan, *Inexact alternating direction methods for image recovery*, SIAM J. Sci. Comput., **33** (2011), 1643–1668.

[39] G. Peyre, S. Bougleux and L. Cohen, *Non-local regularization of inverse problems*, Inverse Problems and Imaging, **5** (2011), 511–530.

[40] L. Rudin S. Osher and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, **60** (1992), 259–268.

[41] S. Setzer, *Operator splittings, Bregman methods and frame shrinkage in image processing*, Int. J. Comput. Vis., **92** (2011), 265–280.

[42] G. Steidl, J.Weickert, T. Brox, P. Mrázek and M.Welk, *On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides*, SIAM J. Numer. Anal., **42** (2004), 686–713.

[43] X.-C.Tai and C.Wu, *Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model*, SSVM 2009, LNCS 5567, Springer, **42** (2009), 502–513.

[44] A. Tikhonov and V. Arsenin, "Solution of Ill-Posed Problems," Winston and Sons, Washington, DC, 1977.

[45] C. Wu, J. Zhang and X.-C. Tai, *Augmented Lagrangian method for total variation restoration with non-quadratic fidelity*, Inverse problems and imaging, **5** (2011), 237–261.

[46] C. Zalinescu, "Convex Analysis in General Vector Spaces," Singapore: World Scientific, 2002.

[47] T. Zeng, *Incorporating known features into a total variation dictionary model for source separation*, Int. Conf. on Image Processing (2008), 577–580.

[48] T. Zeng and F. Malgouyres, *Using Gabor dictionaries in a TV-*L^∞ *model for denoising*, Int. Conf. on Acoust. Speech and Signal Proc. (2006), 865–868.

[49] T. Zeng and M.K. Ng, *On the total variation dictionary model*, IEEE trans. on Image Process., **19** (2010), 821–825.

[50] X. Zhang, M. Burger, X. Bresson and S. Osher, *Bregmanized Nonlocal Regularization for Deconvolution and Sparse Reconstruction*, SIAM J. Imaging Sciences, **3** (2010), 253–76.

[51] M. Zhu and T. Chan, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration*, UCLA CAM Report 08-34 (2008).

E-mail address: 08112068@bjtu.edu.cn
E-mail address: lionel.moisan@parisdescartes.fr
E-mail address: jianyu@bjtu.edu.cn
E-mail address: zeng@hkbu.edu.hk