Restoration of Poissonian Images Using Alternating Direction Optimization

Mário A. T. Figueiredo, *Fellow, IEEE*, José M. Bioucas-Dias, *Member, IEEE*,

***Abstract*—Much research has been devoted to the problem of restoring Poissonian images, namely for medical and astronomical applications. However, the restoration of these images using state-of-the-art regularizers (such as those based on multiscale representations or total variation) is still an active research area, since the associated optimization problems are quite challenging. In this paper, we propose an approach to deconvolving Poissonian images, which is based on an alternating direction optimization method. The standard regularization (or maximum a posteriori) restoration criterion, which combines the Poisson log-likelihood with a (non-smooth) convex regularizer (log-prior), leads to hard optimization problems: the log-likelihood is non-quadratic and non-separable, the regularizer is non-smooth, and there is a non-negativity constraint. Using standard convex analysis tools, we present sufficient conditions for existence and uniqueness of solutions of these optimization problems, for several types of regularizers: total-variation, frame-based analysis, and frame-based synthesis. We attack these problems with an instance of the alternating direction method of multipliers (ADMM), which belongs to the family of augmented Lagrangian algorithms. We study sufficient conditions for convergence and show that these are satisfied, either under total-variation or frame-based (analysis and synthesis) regularization. The resulting algorithms are shown to outperform alternative state-of-the-art methods, both in terms of speed and restoration accuracy.**

***Index Terms*—Image restoration, image deconvolution, Poisson images, convex optimization, alternating direction methods, augmented Lagrangian.**

I. Introduction

A large fraction of (if not all) the work on image denoising, restoration, and reconstruction has been devoted to developing regularizers (priors, from a Bayesian point of view) to deal with the presence of noise and/or the ill-conditioned or ill-posed nature of the observation operator, and to devising efficient algorithms to solve the resulting optimization problems. Much of that work assumes linearity of the observation operator (*e.g.*, the convolution with some point spread function, the acquisition of tomographic projections, or simply an identity in the case of denoising) and the presence of additive Gaussian noise. For this classical scenario, recent state-of-the-art methods adopt non-smooth convex regularizers, such as total-variation or the ℓ_1 norm of frame coefficients; the resulting optimization problems are convex but non-smooth, of very high dimensionality, and have stimulated a considerable amount of research on special purpose algorithms (see [1], [2], [9], [18], [40], [45] and the many references therein).

The algorithms developed for linear operators and Gaussian noise cannot be directly applied to other observation models, such as the Poisson case considered in this paper. Poissonian image models are well studied and highly relevant in fields such as astronomical [38], biomedical [11], [16], [30], [33], [41], [43], and photographic imaging [19]. A very recent and comprehensive overview of deconvolution methods for Poissonian images can be found in [12] (where a state-of-the-art algorithm is also introduced); we refer the reader to that publication for more references on this topic.

The standard criterion for deconvolution of Poissonian images consist of a convex constrained optimization problem: the objective function includes the so-called data term, which is convex and smooth, but not quadratic, plus a convex non-smooth regularizer (the log-likelihood and log-prior, from a Bayesian inference perspective), and a constraint forcing the solution to be non-negative. Although the problem is convex, its very high dimensionality (when dealing with images) usually rules out the direct application of off-the-shelf optimization algorithms.

Furthermore, the Poisson log-likelihood, which is non-quadratic and non-separable (except in the pure denoising case) raises several difficulties to the current state-of-the-art image deconvolution algorithms. More specifically, the Poisson log-likelihood does not have a Lipschitz-continuous gradient, a sufficient condition for the applicability (with guaranteed convergence) of algorithms of the forward-backward splitting (FBS) family [9], [12], [45]. If, nevertheless, an FBS method is applied, it is well known to be slow, specially if the observation operator is severely ill-conditioned, a fact which has stimulated recent research on faster methods [1], [2], [45]; these faster algorithms also require the log-likelihood to have a Lipschitz-continuous gradient, which is not the case with Poissonian observations.

In this paper, we propose a new approach to tackle the optimization problem referred to in the previous paragraph. The proposed algorithm is based on an instance of the *alternating direction method of multipliers* (ADMM) [13], [20], [21], which belongs to the family of augmented Lagrangian methods [29]. For this reason, we call it PIDAL (Poisson image deconvolution by augmented Lagrangian). Although the proposed approach is related to the recent split-Bregman (SB) technique [22], our splitting strategy and resulting algorithm are quite different from the one in [22] (which, moreover, is not adequate for Poissonian image models). Finally, we mention that this paper is an extension of our much shorter

M. Figueiredo and J. Bioucas-Dias are with the *Instituto de Telecomunicações*, *Instituto Superior Técnico*, 1049-001 Lisboa, Portugal. Email: mario.figueiredo@lx.it.pt and bioucas@lx.it.pt.

This work was partially supported by *Fundação para a Ciência e Tecnologia* (FCT), Portuguese Ministry of Science and Higher Education, under projects POSC/EEA-CPS/61271/2004 and PTDC/EEA-TEL/104515/2008.

and preliminary work [17].

In recent work, Douglas-Rachford splitting (DRS) methods were proposed to attack problems in which log-likelihood the does not have a Lipschitz-continuous gradient [8]. In fact, the ADMM is closely related to DRS methods [13], [15], so the method proposed in this paper can also be interpreted from a DRS viewpoint.

In this paper, we will consider three types of regularization: total variation [4], [31] and both frame-based analysis and frame-based synthesis formulations [14]. In Section II, after presenting these three formulations, we derive sufficient conditions for existence and uniqueness of solutions of the corresponding optimization problems. The ADMM framework is reviewed in Section III, where we also introduce the particular variant that is suitable for a linear combination of several convex functions, which is the form of the objective function in hand. In Sections IV and V, we instantiate the proposed variant of ADMM to the three types of regularizers considered and provide sufficient conditions for convergence. Finally, the effectiveness of the resulting algorithm is illustrated in comparison with current state-of-the-art alternatives [11], [12], [19], [37], [39], via a set of experiments reported in Section VI.

II. Problem Formulation

In this section, we begin by reviewing the derivation of the standard log-likelihood resulting from assuming that the observations are Poisson distributed with a mean intensity linearly related with the underlying image to be estimated. Then, we present three different regularization/Bayesian criteria, using synthesis and analysis formulations [14], and study existence/uniqueness of the corresponding solutions.

A. The Linear/Poisson Observation Model

Let $\mathbf{y} = (y_1, ..., y_m) \in \mathbb{N}_0^m$ denote an m-vector of observed counts ($\mathbb{N}_0 = \mathbb{N} \cup \{0\}$), assumed to be a sample of a random vector $\mathbf{Y} = (Y_1, ..., Y_m)$ of m independent Poisson variables, with probability distribution

$$P[\mathbf{Y} = \mathbf{y}|\boldsymbol{\lambda}] = \prod_{i=1}^{m} \frac{\lambda_i^{y_i}\, e^{-\lambda_i}}{y_i!}, \tag{1}$$

where $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_m) \in \mathbb{R}_+^m$ ($\mathbb{R}_+$ denotes the non-negative reals) is the underlying mean (intensity) vector, assumed to be a linear observation of an unknown image $\mathbf{x}$, *i.e.*,

$$\boldsymbol{\lambda} = \mathbf{K}\,\mathbf{x}, \tag{2}$$

where $\mathbf{K}$ the observation operator, which in our finite dimensional setting is simply a matrix $\mathbf{K} \in \mathbb{R}^{m\times n}$. This matrix may model a convolution or some other linear observation mechanism, such as emission tomography. So that the underlying unknown $\mathbf{x}$ can also have the meaning of intensity, it is commonly assumed that $\mathbf{x} \in \mathbb{R}_+^n$. It is usually further assumed that all the elements of $\mathbf{K}$ are non-negative [11], [16], [38]. When dealing with images, we adopt the usual vector notation obtained by stacking the pixels into a vector, in lexicographic order.

Combining (1) and (2) and taking logarithms leads to the negative log-likelihood function [11], [38],

$$\begin{aligned} -\log P[\mathbf{Y} = \mathbf{y}|\mathbf{x}] &= \sum_{i=1}^{m} (\mathbf{K}\,\mathbf{x})_i - y_i \log\left((\mathbf{K}\,\mathbf{x})_i\right) + \log(y_i!) \\ &= \mathcal{L}(\mathbf{K}\,\mathbf{x}) \qquad (3) \\ &= \left(\mathcal{L} \circ \mathbf{K}\right)(\mathbf{x}), \qquad (4) \end{aligned}$$

where $(\mathbf{v})_i$ (or v_i) denotes the i-th component of some vector $\mathbf{v}$ and $\mathcal{L} : \mathbb{R}^m \to \bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ is the negative log-likelihood function for the case $\mathbf{K} = \mathbf{I}$, that is

$$\mathcal{L}(\mathbf{z}) = \sum_{i=1}^{m} -\, y_i\, \log\left(z_i\right) + z_i + \log(y_i!). \tag{5}$$

Dealing with the particular case $z_i = 0$ requires some care, because of the presence of the logarithm. Seen as function of $\mathbf{z}$ to be used in a minimization problem, it is convenient to write the negative log-likelihood function as

$$\mathcal{L}(\mathbf{z}) = C + \sum_{i=1}^{m} \xi(z_i, y_i), \tag{6}$$

where $C = \sum_i \log(y_i!)$ is a finite (recall that $0! = 1$) irrelevant (independent of $\mathbf{z}$) constant and $\xi : \mathbb{R} \times \mathbb{N}_0 \to \bar{\mathbb{R}}$ is defined as

$$\xi(z, y) = z + \iota_{\mathbb{R}_+}(z) - y\, \log(z_+), \tag{7}$$

where ι_S is the indicator function of set S,

$$\iota_S(z) = \begin{cases} 0 & \Leftarrow \quad z \in S \\ +\infty & \Leftarrow \quad z \notin S, \end{cases}$$

$z_+ = \max\{0, z\}$, $\log(0) = -\infty$, and $0 \log(0) = 0$.

The following two propositions characterize ξ as a function of its first argument, as well as $\mathcal{L}$ and $\mathcal{L} \circ \mathbf{K}$, in terms of the key concepts of convex analysis (see Appendix A).

Proposition 1: For any $y \in \mathbb{N}_0$ the function $\xi(\cdot, y) : \mathbb{R} \to \bar{\mathbb{R}}$ is proper, lower semi-continuous (lsc), coercive, and convex. If $y > 0$, then $\xi(\cdot, y)$ is also strictly convex.

Proof: For $y = 0$, $\xi(z, 0) = z + \iota_{\mathbb{R}_+}(z)$, thus $\xi(\cdot, 0)$ is the sum of the identity function with $\iota_{\mathbb{R}_+}$, which are both proper, lsc, coercive, and convex. For any $y > 0$, $\xi(z, y) = z + \iota_{\mathbb{R}_+}(z) - y \log(z_+)$; since $y \log((\cdot)_+)$ is also proper, lsc, coercive, and convex, so is $\xi(\cdot, y)$. Finally, if $y > 0$, $y \log((\cdot)_+)$ is strictly convex (see the definition in Appendix A), thus $\xi(\cdot, y)$ is also strictly convex. ■

Proposition 2: Function $\mathcal{L}$ is proper, lsc, coercive, and convex. If $y_i \neq 0$, for $i = 1, ..., m$, $\mathcal{L}$ is also strictly convex. Function $\mathcal{L} \circ \mathbf{K}$ is proper, lsc, and convex. Function $\mathcal{L} \circ \mathbf{K}$ is coercive if $\mathbf{K}$ is injective. Function $\mathcal{L} \circ \mathbf{K}$ is strictly convex if $\mathbf{K}$ is injective and $y_i \neq 0$, for $i = 1, ..., m$.

Proof: Function $\mathcal{L}$ is the sum of proper, lsc, coercive, convex functions. If $y_i \neq 0$, for $i = 1, ..., m$, the functions in the sum are also strictly convex, thus $\mathcal{L}$ is also strictly convex. Function $\mathcal{L} \circ \mathbf{K}$ is the composition of a proper, lsc, convex functions with a linear function, thus it is proper, lsc, and convex. If $\mathbf{K}$ is injective, its null set is the zero vector, thus $\lim_{\|\mathbf{x}\| \to +\infty} \|\mathbf{K}\,\mathbf{x}\| = +\infty$, thus $\mathcal{L} \circ \mathbf{K}$ is coercive. Finally, if $\mathbf{K}$ is injective and $y_i \neq 0$, for $i = 1, ..., m$, $\mathcal{L}$ is strictly convex, thus so is $\mathcal{L} \circ \mathbf{K}$. ■

B. Regularization Criteria: Analysis and Synthesis Formulations

Under a *maximum a posteriori* (MAP) or regularization criterion, the image estimate is obtained by solving a variational problem: minimizing an objective function, which includes the log-likelihood term plus a regularizer [17], [26], [38], under a positivity constraint. We will now describe three possible ways of building such an objective function.

1) Total Variation Regularization: A standard choice for regularization of digital image restoration/reconstruction problems is the isotropic discrete TV regularizer [4],

$$\mathrm{TV}(\mathbf{x}) = \sum_{s=1}^{n} \sqrt{(\Delta_s^h \mathbf{x})^2 + (\Delta_s^v \mathbf{x})^2}, \tag{8}$$

where $(\Delta_s^h \mathbf{x}$ and $\Delta_s^v \mathbf{x})$ denote the horizontal and vertical first order differences at pixel s, respectively. This regularizer is a discrete version of the TV regularizer proposed in [31]. The resulting optimization problem is

$$\min_{\mathbf{x}} L^{\mathrm{TV}}(\mathbf{x}) \tag{9}$$

with

$$L^{\mathrm{TV}}(\mathbf{x}) = \mathcal{L}(\mathbf{K}\,\mathbf{x}) + \tau\ \mathrm{TV}(\mathbf{x}) + \iota_{\mathbb{R}_+^n}(\mathbf{x}), \tag{10}$$

where $\tau \in \mathbb{R}_+$ is the regularization parameter and the role of $\iota_{\mathbb{R}_+^n}$, the indicator of the first orthant, is to impose the non-negativity constraint on the estimate. The next proposition concerns the existence and uniqueness of minimizers of L^{TV}.

Proposition 3: Consider the function L^{TV} defined in (10).

a) L^{TV} is proper, lsc, and convex.

b) If the intersection of the space of constant images $\{\mathbf{x} = \alpha(1, 1, ..., 1),\ \alpha \in \mathbb{R}\}$ (which is the null space of TV) with the null space of $\mathbf{K}$ is just the zero vector, then L^{TV} is coercive, and (9) has a solution.

c) If $\mathbf{K}$ is injective, then (9) has a solution.

d) If $\mathbf{K} \in \mathbb{R}_+^{m\times n}$, and at least one element of $\mathbf{K}$ is strictly positive, then (9) has a solution.

e) If $\mathbf{K}$ is injective and $y_i \neq 0$, for $i = 1, ..., m$, then L^{TV} is coercive and strictly convex thus there is a unique solution.

Proof:

a) The functions $\iota_{\mathbb{R}_+^n}$, TV, and $\mathcal{L} \circ \mathbf{K}$ (Proposition 2) are proper, lsc, and convex, thus so is their sum.

b) Similar to [5].

c) If $\mathbf{K}$ is injective, its null space is just the zero vector, thus $\mathcal{L} \circ \mathbf{K}$ and L^{TV} are coercive.

d) If all the elements of $\mathbf{K}$ are non-negative and at least one is positive, then the constant vector $(1, 1, ..., 1)$ doesn't belong to the null space of $\mathbf{K}$ and the result follows from (b).

e) If $\mathbf{K}$ is injective and $y_i \neq 0$, for $i = 1, ..., m$, $\mathcal{L} \circ \mathbf{K}$ is strictly convex (Proposition 2), thus so is L^{TV} and its minimizer is unique.

■

2) Frame Analysis Regularization: The use of a regularizer which is a direct function of the unknown image (as in (9)–(10)) corresponds to a so-called analysis-based prior/regularizer [14]. Another well-known type of analysis-based regularization penalizes the norm (typically ℓ_1) of the representation coefficients of $\mathbf{x}$ on some wavelet basis or tight frame, given by $\mathbf{P}\mathbf{x}$, where $\mathbf{P}$ is the analysis operator associated with the frame [28]. This approach leads to the following optimization problem:

$$\min_{\mathbf{x}} L^{\mathrm{FA}}(\mathbf{x}), \tag{11}$$

where FA stands for *frame analysis* and

$$L^{\mathrm{FA}}(\mathbf{x}) = \mathcal{L}(\mathbf{K}\,\mathbf{x}) + \tau\|\mathbf{P}\mathbf{x}\|_1 + \iota_{\mathbb{R}_+^n}(\mathbf{x}); \tag{12}$$

as above, τ is the regularization parameter and $\iota_{\mathbb{R}_+^n}$ imposes the non-negativity constraint on the estimate. The next proposition addresses the existence and uniqueness of minimizers of L^{FA}.

Proposition 4: Consider the function L^{FA} defined in (12).

a) L^{FA} is proper, lsc, convex, and coercive, thus has a minimizer.

b) If $\mathbf{K}$ is injective and $y_i \neq 0$, for $i = 1, ..., m$, then L^{FA} is strictly convex with a unique minimizer.

Proof:

a) The functions $\iota_{\mathbb{R}_+^n}$, $\|\cdot\|_1 \circ \mathbf{P}$, and $\mathcal{L} \circ \mathbf{K}$ (Proposition 2) are proper, lsc, and convex, thus so is their sum. Furthermore, since $\mathbf{P}$ is the analysis operator of a tight frame, its null space is simply the zero vector, thus $\|\cdot\|_1 \circ \mathbf{P}$ is coercive.

b) If $\mathbf{K}$ is injective and $y_i \neq 0$, for $i = 1, ..., m$, $\mathcal{L} \circ \mathbf{K}$ is strictly convex (Proposition 2), thus so is L^{FA} and its minimizer is unique.

■

3) Frame Synthesis Regularization: Finally, another well-known class of approaches is known as synthesis-based [14]. Here, the unknown image is represented on a frame (*e.g.*, of wavelets, curvelets, or other multi-scale system) and then the coefficients of this representation are estimated from the observed data, under some regularizer. With $\mathbf{W} \in \mathbb{R}^{n\times d}$ denoting the synthesis matrix of the frame, the image is written as $\mathbf{x} = \mathbf{W}\,\mathbf{s}$, where $\mathbf{s}$ is the vector of representation coefficients, and the resulting optimization problem is

$$\min_{\mathbf{s}} L^{\mathrm{FS}}(\mathbf{s}) \tag{13}$$

where FS stands for *frame synthesis* and

$$L^{\mathrm{FS}}(\mathbf{s}) = \mathcal{L}(\mathbf{K}\,\mathbf{W}\,\mathbf{s}) + \tau\|\mathbf{s}\|_1 + \iota_{\mathbb{R}_+^n}(\mathbf{W}\,\mathbf{s}). \tag{14}$$

Naturally, the indicator function $\iota_{\mathbb{R}_+^n}$ forcing the image estimate to be non-negative is applied to the image $\mathbf{W}\,\mathbf{s}$ and not its coefficients. The next proposition addresses the existence and uniqueness of minimizers of L^{FS}.

Proposition 5: Consider the function L^{FS} defined in (14).

a) L^{FS} is proper, lsc, convex, and coercive, thus has a minimizer.

b) If $\mathbf{K}\,\mathbf{W}$ is injective and $y_i \neq 0$, for $i = 1, ..., m$, then L^{FS} is strictly convex with a unique minimizer.

Proof:

a) The functions $\iota_{\mathbb{R}^n_+} \circ \mathbf{W}$, $\|\cdot\|_1$, and $\mathcal{L} \circ \mathbf{K}\,\mathbf{W}$ (Proposition 2) are proper, lsc, and convex, thus so is their sum. Furthermore, since $\|\cdot\|_1$ is coercive, L^{FS} is coercive.

b) Same proof as that of Proposition 4 (b).

■

III. The Alternating Direction Method of Multipliers (ADMM)

A. The Standard ADMM

The key tool in this paper is the *alternating direction method of multipliers* (ADMM) [13], [20], [21]. Consider an unconstrained problem of the form

$$\min_{\mathbf{z}\in\mathbb{R}^d} \; f_1(\mathbf{z}) + f_2(\mathbf{G}\,\mathbf{z}), \tag{15}$$

where $f_1 : \mathbb{R}^d \to \bar{\mathbb{R}}$, $f_2 : \mathbb{R}^p \to \bar{\mathbb{R}}$, and $\mathbf{G} \in \mathbb{R}^{p\times d}$. The ADMM for this problem is defined in Fig. 1.

```
Algorithm ADMM
1.  Set $k = 0$, choose $\mu > 0$, $\mathbf{u}_0$, and $\mathbf{d}_0$.
2.  repeat
3.      $\mathbf{z}_{k+1} \in \arg\min_{\mathbf{z}} f_1(\mathbf{z}) + \frac{\mu}{2}\|\mathbf{G}\,\mathbf{z} - \mathbf{u}_k - \mathbf{d}_k\|_2^2$
4.      $\mathbf{u}_{k+1} \in \arg\min_{\mathbf{u}} f_2(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u} - \mathbf{d}_k\|_2^2$
5.      $\mathbf{d}_{k+1} \leftarrow \mathbf{d}_k - (\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1})$
6.      $k \leftarrow k + 1$
7.  until stopping criterion is satisfied.
```

Fig. 1. The ADMM algorithm.

For later reference, we now recall a theorem by Eckstein and Bertsekas in which convergence of (a generalized version of) ADMM is shown.

Theorem 1 (Eckstein-Bertsekas, [13]): Consider problem (15), where $\mathbf{G} \in \mathbb{R}^{p\times d}$ has full column rank and $f_1 : \mathbb{R}^d \to \bar{\mathbb{R}}$ and $f_2 : \mathbb{R}^p \to \bar{\mathbb{R}}$ are closed, proper, convex functions. Consider arbitrary $\mu > 0$ and $\mathbf{u}_0, \mathbf{d}_0 \in \mathbb{R}^p$. Let $\{\eta_k \geq 0,\ k = 0, 1, ...\}$ and $\{\rho_k \geq 0,\ k = 0, 1, ...\}$ be two sequences such that

$$\sum_{k=0}^{\infty} \eta_k < \infty \quad \textit{and} \quad \sum_{k=0}^{\infty} \rho_k < \infty.$$

Consider three sequences $\{\mathbf{z}_k \in \mathbb{R}^d,\ k = 0, 1, ...\}$, $\{\mathbf{u}_k \in \mathbb{R}^p,\ k = 0, 1, ...\}$, and $\{\mathbf{d}_k \in \mathbb{R}^p,\ k = 0, 1, ...\}$ that satisfy

$$\begin{aligned} \left\|\mathbf{z}_{k+1} - \arg\min_{\mathbf{z}} f_1(\mathbf{z}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{z} - \mathbf{u}_k - \mathbf{d}_k\|_2^2\right\| &\leq \eta_k \\ \left\|\mathbf{u}_{k+1} - \arg\min_{\mathbf{u}} f_2(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{z}_{k+1} - \mathbf{u} - \mathbf{d}_k\|_2^2\right\| &\leq \rho_k \end{aligned}$$

and

$$\mathbf{d}_{k+1} = \mathbf{d}_k - (\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1}). \tag{16}$$

Then, if (15) has a solution, say $\mathbf{z}^*$, the sequence $\{\mathbf{z}_k\}$ converges to $\mathbf{z}^*$. If (15) does not have a solution, then at least one of the sequences $\{\mathbf{u}_k\}$ or $\{\mathbf{d}_k\}$ diverges.

According to Theorem 1, it is not necessary to exactly solve the minimizations in lines 3 and 4 of ADMM: as long as the sequences of errors are absolutely summable, convergence is not compromised. As shown in Section IV-D, this fact is quite relevant in designing instances of ADMM, when these minimizations lack closed form solutions.

The proof of Theorem 1 is based on the equivalence between ADMM and the DRS method applied to the dual of problem (15). For recent and comprehensive reviews of ADMM, DRS, and their relationship with Bregman and split-Bregman methods, see [15], [36].

B. A Variant of ADMM

Notice that the ADMM and the associated convergence theorem presented in the previous subsection apply to objective functions of the form (15), *i.e.*, which are the sum of two functions. The fact that our objective functions (9), (11), and (13) involve more than two terms raises the following question: what is the best way of mapping an objective with more than two terms into (15) so that the resulting ADMM is easily applicable and the conditions of Theorem 1 still hold. In this section, we give an answer to this question, which will constitute the core of our approach.

Consider a generalization of problem (15), where instead of two functions, we have J functions, that is,

$$\min_{\mathbf{z}\in\mathbb{R}^d} \sum_{j=1}^{J} g_j(\mathbf{H}^{(j)}\,\mathbf{z}), \tag{17}$$

where $g_j : \mathbb{R}^{p_j} \to \bar{\mathbb{R}}$ are closed, proper, convex functions, and $\mathbf{H}^{(j)} \in \mathbb{R}^{p_j\times d}$ are arbitrary matrices. The minimization problem (17) can be written as (15) using the following correspondences: $f_1 = 0$,

$$\mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \in \mathbb{R}^{p\times d}, \tag{18}$$

where $p = p_1 + \cdots + p_J$, and $f_2 : \mathbb{R}^{p\times d} \to \bar{\mathbb{R}}$ given by

$$f_2(\mathbf{u}) = \sum_{j=1}^{J} g_j(\mathbf{u}^{(j)}), \tag{19}$$

where $\mathbf{u}^{(j)} \in \mathbb{R}^{p_j}$ and $\mathbf{u} = [(\mathbf{u}^{(1)})^T, \ldots, (\mathbf{u}^{(J)})^T]^T \in \mathbb{R}^p$.

We are now in position to apply ADMM. The resulting algorithm has exactly the same structure as the one in Fig. 1 with

$$\mathbf{d}_k = \begin{bmatrix} \mathbf{d}_k^{(1)} \\ \vdots \\ \mathbf{d}_k^{(J)} \end{bmatrix}, \quad \mathbf{u}_k = \begin{bmatrix} \mathbf{u}_k^{(1)} \\ \vdots \\ \mathbf{u}_k^{(J)} \end{bmatrix}.$$

The fact that $f_1 = 0$ turns Step 3 of the algorithm into a simple quadratic minimization problem, which has a unique solution if $\mathbf{G}$ has full column rank:

$$\begin{aligned} \arg\min_{\mathbf{z}} \left\|\mathbf{G}\,\mathbf{z} - \boldsymbol{\zeta}_k\right\|_2^2 &= \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\boldsymbol{\zeta}_k, \qquad (20) \\ &= \left[\sum_{j=1}^{J}(\mathbf{H}^{(j)})^T\mathbf{H}^{(j)}\right]^{-1}\sum_{j=1}^{J}\left(\mathbf{H}^{(j)}\right)^T\boldsymbol{\zeta}_k^{(j)}, \end{aligned}$$

where $\boldsymbol{\zeta}_k = \mathbf{u}_k + \mathbf{d}_k$ (and, naturally, $\boldsymbol{\zeta}_k^{(j)} = \mathbf{u}_k^{(j)} + \mathbf{d}_k^{(j)}$) and the second equality results from the particular structure of $\mathbf{G}$ in (18).

Furthermore, our particular way of mapping problem (17) into problem (15) allows decoupling the minimization in Step 4 of ADMM into a set of J independent ones. In fact,

$$\mathbf{u}_{k+1} \leftarrow \arg\min_{\mathbf{u}} f_2(\mathbf{u}) + \frac{\mu}{2}\|\mathbf{G}\,\mathbf{z}_{k+1} - \mathbf{u} - \mathbf{d}_k\|_2^2 \tag{21}$$

which can be written as

$$\begin{bmatrix} \mathbf{u}_{k+1}^{(1)} \\ \vdots \\ \mathbf{u}_{k+1}^{(J)} \end{bmatrix} \leftarrow \arg \min_{\mathbf{u}^{(1)},\ldots,\mathbf{u}^{(J)}} \; g_1(\mathbf{u}^{(1)}) + \cdots + g_J(\mathbf{u}^{(J)})$$
$$+\frac{\mu}{2}\left\| \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \mathbf{z}_{k+1} - \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(J)} \end{bmatrix} - \begin{bmatrix} \mathbf{d}_k^{(1)} \\ \vdots \\ \mathbf{d}_k^{(J)} \end{bmatrix} \right\|_2^2 .$$

Clearly, the minimizations with respect to $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(J)}$ are decoupled, thus can be solved separately, leading to

$$\mathbf{u}_{k+1}^{(j)} \leftarrow \arg \min_{\mathbf{v} \in \mathbb{R}^{p_j}} \; g_j(\mathbf{v}) + \frac{\mu}{2}\left\|\mathbf{v} - \mathbf{s}_k^{(1)}\right\|_2^2, \tag{22}$$

for $j = 1, ..., J$, where

$$\mathbf{s}_k^{(j)} = \mathbf{H}^{(j)}\mathbf{z}_{k+1} - \mathbf{d}_k^{(j)}.$$

Eq. (22) defines the so-called *Moreau proximity operators* [9] of $g_1, ..., g_J$, applied to $\mathbf{s}_k^{(1)}, ..., \mathbf{s}_k^{(J)}$, respectively, denoted as

$$\mathbf{u}_{k+1}^{(j)} = \boldsymbol{\Psi}_{g_j/\mu}(\mathbf{s}_k^{(j)}). \tag{23}$$

Some comments on the algorithm are relevant. Firstly, being exactly an ADMM, and since all the functions g_j, for $j = 1, ..., J$, are closed, proper, and convex, convergence is guaranteed if $\mathbf{G}$ has full column rank. This full column rank condition, which is also required for the inverse in (20) to exist, will be studied in the next section for each of the specific problems considered in this paper.

For some functions, this mapping can be computed exactly in closed form. For example, if $g_j(\mathbf{x}) = \|\mathbf{x}\|_1$, the corresponding proximity operator $\boldsymbol{\Psi}_{g_j/\mu}$ is simply a soft threshold,

$$\boldsymbol{\Psi}_{g_j/\mu}(\mathbf{v}) = \text{soft}(\mathbf{v}, 1/\mu) = \text{sign}(\mathbf{v}) \odot \max\{|\mathbf{v}| - (1/\mu), 0\}, \tag{24}$$

where sign$(\cdot)$ denotes the component-wise application of the sign function, $\odot$ denotes the component-wise product, $|\mathbf{v}|$ denotes the vector of absolute values of the elements of $\mathbf{v}$, and the maximum is computed in a component-wise fashion. For other functions, the corresponding Moreau proximity operator does not have a simple close form solution and needs to be computed numerically.

IV. Poissonian Image Reconstruction with TV-Based Regularization

A. Applying ADMM

In this section, we apply the algorithmic framework presented in Section III-B to the total-variation-based criterion (9)–(10). The objective function in (10) has the form (17) with $J = 3$,

$$g_1 \equiv \mathcal{L}, \quad g_2 \equiv \tau\ \text{TV}, \quad g_3 \equiv \iota_{\mathbb{R}_+^n} \tag{25}$$

and

$$\mathbf{H}^{(1)} \equiv \mathbf{K}, \quad \mathbf{H}^{(2)} \equiv \mathbf{I}, \quad \mathbf{H}^{(3)} \equiv \mathbf{I}. \tag{26}$$

The resulting ADMM algorithm, which we call PIDAL-TV (Poisson image deconvolution by augmented Lagrangian – total variation), is shown in Fig. 2.

```
Algorithm Poisson Image Deconvolution by AL (PIDAL-TV)
1.  Choose $\mathbf{u}_0^{(1)}, \mathbf{u}_0^{(2)}, \mathbf{u}_0^{(3)}, \mathbf{d}_0^{(1)}, \mathbf{d}_0^{(2)}, \mathbf{d}_0^{(3)}$, $\mu$, and $\tau$. Set $k \leftarrow 0$.
2.  repeat
3.      $\boldsymbol{\zeta}_k^{(1)} \leftarrow \mathbf{u}_k^{(1)} + \mathbf{d}_k^{(1)}$
4.      $\boldsymbol{\zeta}_k^{(2)} \leftarrow \mathbf{u}_k^{(2)} + \mathbf{d}_k^{(2)}$
5.      $\boldsymbol{\zeta}_k^{(3)} \leftarrow \mathbf{u}_k^{(3)} + \mathbf{d}_k^{(3)}$
6.      $\boldsymbol{\gamma}_k \leftarrow \mathbf{K}^T\boldsymbol{\zeta}_k^{(1)} + \boldsymbol{\zeta}_k^{(2)} + \boldsymbol{\zeta}_k^{(3)}$
7.      $\mathbf{z}_{k+1} \leftarrow \left(\mathbf{K}^T\mathbf{K} + 2\,\mathbf{I}\right)^{-1}\boldsymbol{\gamma}_k$
8.      $\boldsymbol{\nu}_k^{(1)} \leftarrow \mathbf{K}\,\mathbf{z}_{k+1} - \mathbf{d}_k^{(1)}$
9.      $\mathbf{u}_{k+1}^{(1)} \leftarrow \arg\min_{\mathbf{v}} \frac{\mu}{2}\|\mathbf{v} - \boldsymbol{\nu}_k^{(1)}\|_2^2 + \sum_{i=1}^{m}\xi(v_i, y_i)$
10.     $\boldsymbol{\nu}_k^{(2)} \leftarrow \mathbf{z}_{k+1} - \mathbf{d}_k^{(2)}$
11.     $\mathbf{u}_{k+1}^{(2)} \leftarrow \arg\min_{\mathbf{v}} \frac{\mu}{2}\|\mathbf{v} - \boldsymbol{\nu}_k^{(2)}\|^2 + \tau\,\text{TV}(\mathbf{v})$.
12.     $\boldsymbol{\nu}_k^{(3)} \leftarrow \mathbf{z}_{k+1} - \mathbf{d}_k^{(3)}$
13.     $\mathbf{u}_{k+1}^{(3)} \leftarrow \arg\min_{\mathbf{v}} \frac{\mu}{2}\|\mathbf{v} - \boldsymbol{\nu}_k^{(3)}\|^2 + \iota_{\mathbb{R}_+^n}(\mathbf{v})$.
14.     $\mathbf{d}_{k+1}^{(1)} \leftarrow \mathbf{d}_k^{(1)} - (\mathbf{K}\,\mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(1)})$
15.     $\mathbf{d}_{k+1}^{(2)} \leftarrow \mathbf{d}_k^{(2)} - (\mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(2)})$
16.     $\mathbf{d}_{k+1}^{(3)} \leftarrow \mathbf{d}_k^{(3)} - (\mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(3)})$
17.     $k \leftarrow k + 1$
18. until some stopping criterion is satisfied.
```

Fig. 2. The PIDAL-TV algorithm.

B. Implementation Aspects and Computational Cost of PIDAL-TV

Notice that line 7 of PIDAL-TV corresponds to (20) for the particular form of matrix $\mathbf{G}$ in this problem: $\mathbf{G} = [\mathbf{K}^T\ \mathbf{I}\ \mathbf{I}]^T$ (see (25) and (26)), which is of course of full column rank. Moreover, if $\mathbf{K}$ models a periodic convolution, it is a block circulant matrix and the inversion in line 7 of the algorithm can be implemented in $O(n \log n)$ operations, via the FFT algorithm. Although this is a well-known fact, we include the derivation in the next paragraph, for the sake of completeness.

Assuming that the convolution is periodic, $\mathbf{K}$ is block-circulant with circulant blocks and can be factorized as

$$\mathbf{K} = \mathbf{U}^H\mathbf{D}\mathbf{U}, \tag{27}$$

where $\mathbf{U}$ is the matrix that represents the 2D discrete Fourier transform (DFT), $\mathbf{U}^H = \mathbf{U}^{-1}$ is its inverse ($\mathbf{U}$ is unitary, *i.e.*, $\mathbf{U}\mathbf{U}^H = \mathbf{U}^H\mathbf{U} = \mathbf{I}$), and $\mathbf{D}$ is a diagonal matrix containing the DFT coefficients of the convolution operator represented by $\mathbf{K}$. Thus (with $\mathbf{K}^T = \mathbf{K}^H$, since $\mathbf{K}$ is a real matrix)

$$\begin{aligned}\left(\mathbf{K}^T\mathbf{K} + 2\mathbf{I}\right)^{-1} &= \left(\mathbf{U}^H\mathbf{D}^*\mathbf{D}\mathbf{U} + 2\mathbf{U}^H\mathbf{U}\right)^{-1} &(28)\\ &= \mathbf{U}^H\left(|\mathbf{D}|^2 + 2\,\mathbf{I}\right)^{-1}\mathbf{U}, &(29)\end{aligned}$$

where $(\cdot)^*$ denotes complex conjugate and $|\mathbf{D}|^2$ the squared absolute values of the entries of $\mathbf{D}$. Since $|\mathbf{D}|^2 + 2\,\mathbf{I}$ is diagonal, its inversion has $O(n)$ cost. Products by $\mathbf{U}$ and $\mathbf{U}^H$ can be carried out with $O(n\log n)$ cost using the FFT algorithm.

The minimization in line 9 of PIDAL-TV is separable. With respect to each component, it has the form

$$\min_v \left\{ v + \iota_{\mathbb{R}_+}(v) - y\,\log(v_+) + \frac{\mu}{2}(v-\nu)^2 \right\}. \qquad (30)$$

It is simple to show that the solution of (30) leads to

$$u^{(1)}_{i,k+1} = \frac{1}{2}\left(\nu^{(1)}_{i,k} - \frac{1}{\mu} + \sqrt{\left(\nu^{(1)}_{i,k} - \frac{1}{\mu}\right)^2 + 4\,y_i} \right), \qquad (31)$$

where $\nu^{(1)}_{i,k}$ denotes the i-th component of $\boldsymbol{\nu}^{(1)}_k$. Notice that $u^{(1)}_{i,k+1}$ is always a non-negative quantity.

The minimization in line 11 of PIDAL-TV is, by definition, the Moreau proximity operator $\boldsymbol{\Psi}_{(\tau/\mu)\mathrm{TV}} : \mathbb{R}^n \to \mathbb{R}^n$ [9], which corresponds to applying TV-denoising to $\boldsymbol{\nu}^{(2)}_k$. Below, we address in detail the issue of how to implement this operator and its implications to the convergence of PIDAL-TV. Suffice it to say here that most TV-denoising algorithms have $O(n)$ cost.

The minimization in line 13 of PIDAL-TV corresponds to the projection of $\boldsymbol{\nu}^{(3)}_k$ onto the first orthant, thus

$$\mathbf{u}^{(3)}_{k+1} = \max\left\{\boldsymbol{\nu}^{(3)}_k, 0\right\}, \qquad (32)$$

where the maximum is to be understood in a component-wise sense; this projection has of course $O(n)$ cost.

From the observations in the previous paragraphs, the computational costs of the lines of PIDAL-TV are the following. Lines 3, 4, 5, 9, 10, 11, 12, 13, 15, and 16 have $O(n)$ cost. Lines 6, 7, 8, and 14 have $O(n\log n)$ cost. Thus the computational cost of PIDAL-TV scales as $O(n\log n)$.

C. Convergence of PIDAL-TV: Exact TV

Convergence of PIDAL-TV is addressed by the following corollary of Theorem 1, for the (ideal) case where $\boldsymbol{\Psi}_{(\tau/\mu)\mathrm{TV}}$ (line 11) is computed exactly. The minimizations in lines 9 and 13 have the closed-form solutions given in (31) and (32).

Corollary 1: If the minimizations in lines 9, 11, and 13 of PIDAL-TV are solved exactly, then the algorithm converges to a minimizer of (10), provided one exists.

Proof: PIDAL-TV is an instance of ADMM in Fig. 1, where $f_1 = 0$ and f_2 has the form (19), with $J = 3$ and the g_i functions given in (25), which are all closed, proper, and convex. Function f_2 is thus also closed, proper, and convex. Matrix $\mathbf{G} = [\mathbf{K}^T\ \mathbf{I}\ \mathbf{I}]^T$ has full column rank. The minimization in line 4 of ADMM corresponds to lines 9, 11, and 13 of PIDAL-TV; if these minimizations are solved exactly, then according to Theorem 1, convergence to a minimizer of the objective function, if one exists, is guaranteed. ■

D. Convergence of PIDAL-TV: Approximate TV

As is well known, the TV denoising problem has no closed form solution, and many iterative algorithms have been proposed to solve it (see [4], [6], [10], [31], and references therein). Here, we adopt Chambolle's algorithm [4].

Of course, in practice, Chambolle's (or any other iterative) algorithm can only run for a finite number of iterations, thus the minimization in line 11 of PIDAL-TV can only be solved approximately. However, as stated in Theorem 1, this will not compromise the convergence of ADMM/PIDAL-TV, if the corresponding error sequence is summable. To achieve this goal, we adopt a simple procedure in which the internal variables of Chambolle's algorithm (the discrete gradient, see [4]) are initialized, in each iteration of PIDAL-TV, with those obtained in the previous iteration. We will now formalize this idea and provide experimental evidence that this procedure does produce a summable error sequence.

Let us define $\beta = \tau/\mu$ and let $(\mathbf{s},\mathbf{q}) = \widetilde{\boldsymbol{\Psi}}^{(t)}_{\beta\,\mathrm{TV}}(\mathbf{r},\mathbf{p})$ be the result of running t iterations of Chambolle's algorithm with its internal variables initialized at $\mathbf{p}$, where $\mathbf{s}$ is the obtained (denoised) image (which is approximately $\boldsymbol{\Psi}_{\beta\,\mathrm{TV}}(\mathbf{r})$) and $\mathbf{q}$ the final values of the internal variables. Consider now two possible implementations of line 11 of PIDAL-TV:

$$\begin{aligned}\text{11(a).}\quad (\mathbf{u}^{(2,a)}_{k+1}, \mathbf{p}_{k+1}) &= \widetilde{\boldsymbol{\Psi}}^{(t)}_{\beta\,\mathrm{TV}}(\boldsymbol{\nu}^{(2)}_k, \mathbf{p}_k) &(33)\\ \text{11(b).}\quad (\mathbf{u}^{(2,b)}_{k+1}, \cdot) &= \widetilde{\boldsymbol{\Psi}}^{(t)}_{\beta\,\mathrm{TV}}(\boldsymbol{\nu}^{(2)}_k, \mathbf{p}). &(34)\end{aligned}$$

Implementation 11(a) uses the proposed internal variables initialization, whereas in 11(b) the internal variables are always initialized to the same values (usually zeros). Consider now the corresponding error sequences

$$\begin{aligned}\rho^{(a)}_k &= \left\|\mathbf{u}^{(2,a)}_{k+1} - \boldsymbol{\Psi}_{\beta\,\mathrm{TV}}(\boldsymbol{\nu}^{(2)}_k)\right\| &(35)\\ \rho^{(b)}_k &= \left\|\mathbf{u}^{(2,b)}_{k+1} - \boldsymbol{\Psi}_{\beta\,\mathrm{TV}}(\boldsymbol{\nu}^{(2)}_k)\right\|. &(36)\end{aligned}$$

Notice that since the two other minimizations (lines 9 and 13) are solved exactly, the sequences $\rho^{(a)}_k$ and $\rho^{(b)}_k$ correspond to the sequence ρ_k in Theorem 1.

The following experiment provides evidence that $\rho^{(a)}_k$ is summable, but $\rho^{(b)}_k$ is not. Consider the same setup as in the first experiment in [37]: the original image is a portion of the Cameraman image scaled to a maximum value of 3000 and then blurred with a Gaussian kernel of unit variance; the observed image is generated according to (1). As in [37], we set $\tau = 0.008$ and $\mu = \tau/50$. The number of iterations of Chambolle's algorithm is set to 5 or 20. To compute $\boldsymbol{\Psi}_{\beta\,\mathrm{TV}}(\boldsymbol{\nu}^{(2)}_k)$ (almost) exactly, we run 4000 iterations of Chambolle's algorithm. In Figure 3, it is clear that the $\rho^{(b)}_k$ sequences are not even decreasing, let alone summable. In contrast, the sequences $\rho^{(a)}_k$ approach zero, for both choices of t. Evidence for the summability of the $\rho^{(a)}_k$ sequences is provided by the fact that by fitting a function of the form $A\,(1/k)^{\omega}$ to the tails of these sequences (*i.e.*, for $k = 20, ..., 200$), we obtain

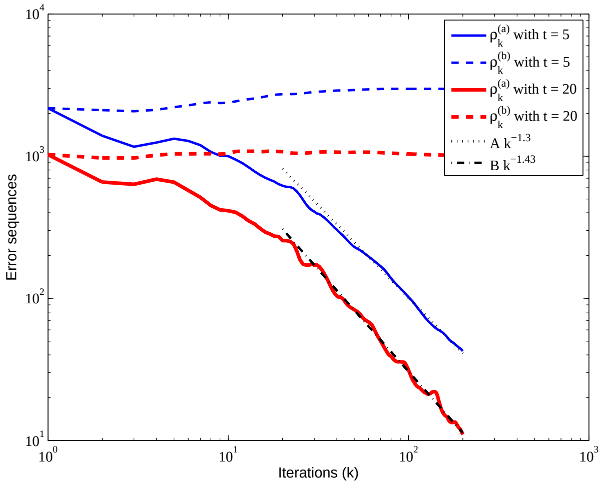

Fig. 3. Error sequences $\rho_k^{(a)}$ and $\rho_k^{(b)}$ for $t = 5$ and $t = 20$ (number of iterations of Chambolle's algorithm) and fitted functions of the form $A\,t^{\omega}$ to the sequences $\{\rho_k^{(a)}, k = 20, 21, ...200\}$.

values of ω that are larger than one ($\omega \simeq 1.3$, for $t = 5$, and $\omega \simeq 1.43$, for $t = 20$).

In conclusion, the experiment reported in the previous paragraph, though of course not a formal proof, strongly suggests that by implementing line 11 of PIDAL-TV as in (33), the corresponding error sequence (with respect to the exact minimizations) is summable, thus we can invoke Theorem 1 to state that PIDAL-TV converges. Moreover, this experiment shows that this is achieved with a quite small number of iterations in each call of Chambolle's algorithm. In all our experiments with PIDAL-TV, we thus use (33), with $t = 5$.

V. Poissonian Image Reconstruction with Frame-Based Regularization

We now consider the frame-based analysis criterion (11), and the frame-based synthesis criterion (13).

A. Analysis Criterion

In this case, the objective function is given by (12), which has the form (17) with $J = 3$,

$$g_1 \equiv \mathcal{L}, \quad g_2 \equiv \tau\|\cdot\|_1, \quad g_3 \equiv \iota_{\mathbb{R}_+^n} \tag{37}$$

and

$$\mathbf{H}^{(1)} \equiv \mathbf{K}, \quad \mathbf{H}^{(2)} \equiv \mathbf{P}, \quad \mathbf{H}^{(3)} \equiv \mathbf{I}.$$

The resulting instance of ADMM, which we call PIDAL-FA (where FA stands for "frame analysis"), is shown in Fig. 4. The matrix being inverted in line 7 results from assuming that $\mathbf{P}$ is the analysis operator of a 1-tight (Parseval) frame, thus $(\mathbf{H}^{(2)})^T\mathbf{H}^{(2)} = \mathbf{P}^T\mathbf{P} = \mathbf{I}$. Notice that line 7 of PIDAL-FA corresponds to (20) for the particular form of matrix $\mathbf{G}$ in this case, $\mathbf{G} = [\mathbf{K}^T \;\; \mathbf{P}^T \;\; \mathbf{I}]^T$, which of course has full column rank. As in PIDAL-TV, if $\mathbf{K}$ models a convolution, the inverse $(\mathbf{K}^T\mathbf{K} + 2\,\mathbf{I})^{-1}$ can be computed with $O(n \log n)$ cost, using the FFT (see (29)).

For most tight frames used in image processing, products by $\mathbf{P}$ and $\mathbf{P}^T$ correspond to the inverse and direct transforms for which fast algorithms exist. For example, in the case of translation-invariant wavelet transforms, these products can be computed using the undecimated wavelet transform with $O(n \log n)$ cost [25], [28]. Curvelets also constitute a Parseval frame for which fast $O(n \log n)$ implementations of the forward and inverse transform exist [3]. Yet another example of a redundant Parseval frame is provided by complex wavelets, with the corresponding direct and inverse transforms having $O(n)$ cost [24], [34]. In conclusion, for a large class of choices of $\mathbf{P}$, the cost of lines 6, 10, and 15 of PIDAL-FA is $O(n \log n)$.

```
Algorithm Poisson Image Deconvolution by AL (PIDAL-FA)
1.  Choose $\mathbf{u}_0^{(1)}, \mathbf{u}_0^{(2)}, \mathbf{u}_0^{(3)}, \mathbf{d}_0^{(1)}, \mathbf{d}_0^{(2)}, \mathbf{d}_0^{(3)}$, $\mu$, and $\tau$. Set $k \leftarrow 0$.
2.  repeat
3.      $\boldsymbol{\zeta}_k^{(1)} \leftarrow \mathbf{u}_k^{(1)} + \mathbf{d}_k^{(1)}$
4.      $\boldsymbol{\zeta}_k^{(2)} \leftarrow \mathbf{u}_k^{(2)} + \mathbf{d}_k^{(2)}$
5.      $\boldsymbol{\zeta}_k^{(3)} \leftarrow \mathbf{u}_k^{(3)} + \mathbf{d}_k^{(3)}$
6.      $\boldsymbol{\gamma}_k \leftarrow \mathbf{K}^T\boldsymbol{\zeta}_k^{(1)} + \mathbf{P}^T\boldsymbol{\zeta}_k^{(2)} + \boldsymbol{\zeta}_k^{(3)}$
7.      $\mathbf{z}_{k+1} \leftarrow \left(\mathbf{K}^T\mathbf{K} + 2\,\mathbf{I}\right)^{-1}\boldsymbol{\gamma}_k$
8.      $\boldsymbol{\nu}_k^{(1)} \leftarrow \mathbf{K}\,\mathbf{z}_{k+1} - \mathbf{d}_k^{(1)}$
9.      $\mathbf{u}_{k+1}^{(1)} \leftarrow \frac{1}{2}\left(\nu_{i,k}^{(1)} - \frac{1}{\mu} + \sqrt{\left(\nu_{i,k}^{(1)} - \frac{1}{\mu}\right)^2 + 4\,y_i}\right)$
10.     $\boldsymbol{\nu}_k^{(2)} \leftarrow \mathbf{P}\mathbf{z}_{k+1} - \mathbf{d}_k^{(2)}$
11.     $\mathbf{u}_{k+1}^{(2)} \leftarrow \arg\min_{\mathbf{v}} \frac{\mu}{2}\|\mathbf{v} - \boldsymbol{\nu}_k^{(2)}\|^2 + \tau\,\|\mathbf{v}\|_1.$
12.     $\boldsymbol{\nu}_k^{(3)} \leftarrow \mathbf{z}_{k+1} - \mathbf{d}_k^{(3)}$
13.     $\mathbf{u}_{k+1}^{(3)} \leftarrow \max\left\{\boldsymbol{\nu}_k^{(3)}, 0\right\}.$
14.     $\mathbf{d}_{k+1}^{(1)} \leftarrow \mathbf{d}_k^{(1)} - \mathbf{K}\,\mathbf{z}_{k+1} + \mathbf{u}_{k+1}^{(1)}$
15.     $\mathbf{d}_{k+1}^{(2)} \leftarrow \mathbf{d}_k^{(2)} - \mathbf{P}\mathbf{z}_{k+1} + \mathbf{u}_{k+1}^{(2)}$
16.     $\mathbf{d}_{k+1}^{(3)} \leftarrow \mathbf{d}_k^{(3)} - \mathbf{z}_{k+1} + \mathbf{u}_{k+1}^{(3)}$
17.     $k := k + 1$
18. until some stopping criterion is satisfied.
```

Fig. 4. The PIDAL-FA algorithm.

The expressions in lines 9 and 13 of PIDAL-FA are similar to those in lines 9 and 13 of PIDAL-TV, respectively; see also (30), (31), and (32).

The minimization in line 11 is, by definition, the Moreau proximity operator of the ℓ_1 norm [9], which corresponds to a soft-threshold (24).

In summary, from the observations in the previous paragraphs, the computational costs of the lines of PIDAL-FA are the following. Lines 3, 4, 5, 9, 11, 12, 13, and 16 have $O(n)$ cost. Lines 6, 7, 8, 10, 14, and 15 have $O(n \log n)$ cost. Thus the computational cost of PIDAL-FA scales as $O(n \log n)$.

Finally, convergence of PIDAL-FA is addressed by the following corollary of Theorem 1.

Corollary 2: The PIDAL-FA algorithm converges to a minimizer of (11), provided one exists.

Proof: The proof is similar to, but simpler than, that of

```
Algorithm Poisson Image Deconvolution by AL (PIDAL-FS)
1.  Choose $\mathbf{u}_0^{(1)}$, $\mathbf{u}_0^{(2)}$, $\mathbf{u}_0^{(3)}$, $\mathbf{d}_0^{(1)}$, $\mathbf{d}_0^{(2)}$, $\mathbf{d}_0^{(3)}$, $\mu$, and $\tau$. Set $k \leftarrow 0$.
2.  repeat
3.      $\boldsymbol{\zeta}_k^{(1)} \leftarrow \mathbf{u}_k^{(1)} + \mathbf{d}_k^{(1)}$
4.      $\boldsymbol{\zeta}_k^{(2)} \leftarrow \mathbf{u}_k^{(2)} + \mathbf{d}_k^{(2)}$
5.      $\boldsymbol{\zeta}_k^{(3)} \leftarrow \mathbf{u}_k^{(3)} + \mathbf{d}_k^{(3)}$
6.      $\boldsymbol{\gamma}_k \leftarrow \mathbf{W}^T\mathbf{K}^T\boldsymbol{\zeta}_k^{(1)} + \boldsymbol{\zeta}_k^{(2)} + \mathbf{W}^T\boldsymbol{\zeta}_k^{(3)}$
7.      $\mathbf{z}_{k+1} \leftarrow \left(\mathbf{W}^T\mathbf{K}^T\mathbf{K}\,\mathbf{W} + \mathbf{I} + \mathbf{W}^T\mathbf{W}\right)^{-1}\boldsymbol{\gamma}_k$
8.      $\boldsymbol{\nu}_k^{(1)} \leftarrow \mathbf{K}\,\mathbf{W}\,\mathbf{z}_{k+1} - \mathbf{d}_k^{(1)}$
9.      $\mathbf{u}_{k+1}^{(1)} \leftarrow \frac{1}{2}\left(\nu_{i,k}^{(1)} - \frac{1}{\mu} + \sqrt{\left(\nu_{i,k}^{(1)} - \frac{1}{\mu}\right)^2 + 4\,y_i}\right)$
10.     $\boldsymbol{\nu}_k^{(2)} \leftarrow \mathbf{z}_{k+1} - \mathbf{d}_k^{(2)}$
11.     $\mathbf{u}_{k+1}^{(2)} \leftarrow \mathrm{soft}(\boldsymbol{\nu}_k^{(2)}, \tau/\mu)$
12.     $\boldsymbol{\nu}_k^{(3)} \leftarrow \mathbf{W}\,\mathbf{z}_{k+1} - \mathbf{d}_k^{(3)}$
13.     $\mathbf{u}_{k+1}^{(3)} \leftarrow \max\left\{\boldsymbol{\nu}_k^{(3)}, 0\right\}$.
14.     $\mathbf{d}_{k+1}^{(1)} \leftarrow \mathbf{d}_k^{(1)} - \mathbf{K}\,\mathbf{W}\,\mathbf{z}_{k+1} + \mathbf{u}_{k+1}^{(1)}$
15.     $\mathbf{d}_{k+1}^{(2)} \leftarrow \mathbf{d}_k^{(2)} - \mathbf{z}_{k+1} + \mathbf{u}_{k+1}^{(2)}$
16.     $\mathbf{d}_{k+1}^{(3)} \leftarrow \mathbf{d}_k^{(3)} - \mathbf{W}\,\mathbf{z}_{k+1} + \mathbf{u}_{k+1}^{(3)}$
17.     $k := k + 1$
18. until some stopping criterion is satisfied.
```

Fig. 5. The PIDAL-FS algorithm.

TABLE I
INITIALIZATION OF THE PIDAL ALGORITHMS.

	$\mathbf{u}_0^{(1)}$	$\mathbf{u}_0^{(2)}$	$\mathbf{u}_0^{(3)}$	$\mathbf{d}_0^{(1)}$	$\mathbf{d}_0^{(2)}$	$\mathbf{d}_0^{(3)}$
PIDAL-TV	$\mathbf{y}$	$\mathbf{y}$	$\mathbf{y}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
PIDAL-FA	$\mathbf{y}$	$\mathbf{P}\mathbf{y}$	$\mathbf{y}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$
PIDAL-FS	$\mathbf{y}$	$\mathbf{W}^T\mathbf{K}^T\mathbf{y}$	$\mathbf{K}^T\mathbf{y}$	$\mathbf{0}$	$\mathbf{0}$	$\mathbf{0}$

Corollary 1, since all the minimizations involved are solved exactly in closed form. Clearly, matrix $\mathbf{G} = [\mathbf{K}^T \;\; \mathbf{P}^T \;\; \mathbf{I}]^T$ has full column rank, thus Theorem 1 guarantees convergence to a minimizer of the objective function. ■

B. Synthesis Criterion

In the synthesis formulation, the objective function is given by (13), which has the form (17) with $J = 3$,

$$g_1 \equiv \mathcal{L}, \quad g_2 \equiv \tau\|\cdot\|_1, \quad g_3 \equiv \iota_{\mathbb{R}_+^n}$$

and

$$\mathbf{H}^{(1)} \equiv \mathbf{K}\,\mathbf{W}, \quad \mathbf{H}^{(2)} \equiv \mathbf{I}, \quad \mathbf{H}^{(3)} \equiv \mathbf{W}.$$

The resulting ADMM algorithm, which we call PIDAL-FS (where FS stands for "frame synthesis"), is shown in Fig. 5.

Notice that line 7 of PIDAL-FS corresponds to (20) for the particular form of matrix $\mathbf{G}$ in this problem: $\mathbf{G} = [(\mathbf{K}\,\mathbf{W})^T \;\; \mathbf{I} \;\; \mathbf{W}^T]^T$. This matrix has of course full column rank. However, even if $\mathbf{K}$ models a periodic convolution (thus is block circulant), the question remains of how to efficiently compute the matrix inverse in line 7, since $\mathbf{K}\,\mathbf{W}$ is not block circulant. The next paragraph shows how to sidestep this difficulty.

Consider that matrix $\mathbf{W}$ corresponds to a 1-tight (Parseval) frame, *i.e.,* $\mathbf{W}\,\mathbf{W}^H = \mathbf{I}$, and start by noticing that $\mathbf{W}^T\mathbf{K}^T\mathbf{K}\,\mathbf{W} + \mathbf{I} + \mathbf{W}^T\mathbf{W} = \mathbf{W}^H(\mathbf{K}^H\mathbf{K} + \mathbf{I})\mathbf{W} + \mathbf{I}$. Applying the Sherman-Morrison-Woodbury (SMW) matrix inversion formula yields

$$\begin{aligned}\left(\mathbf{W}^H(\mathbf{K}^H\mathbf{K} + \mathbf{I})\mathbf{W} + \mathbf{I}\right)^{-1} &= \\ = \;\; \mathbf{I} - \mathbf{W}^H\Big(\mathbf{W}\,\mathbf{W}^H + \big(\mathbf{K}^H\mathbf{K} + \mathbf{I}\big)^{-1}\Big)^{-1}&\mathbf{W} \\ = \;\; \mathbf{I} - \mathbf{W}^H\Big(\mathbf{I} + \big(\mathbf{K}^H\mathbf{K} + \mathbf{I}\big)^{-1}\Big)^{-1}&\mathbf{W}. \qquad (38)\end{aligned}$$

Using the factorization (27), we have

$$\Big(\mathbf{I} + \big(\mathbf{K}^H\mathbf{K} + \mathbf{I}\big)^{-1}\Big)^{-1} = \mathbf{U}^H\big(\mathbf{I} + (|\mathbf{D}|^2 + \mathbf{I})^{-1}\big)^{-1}\mathbf{U}, \quad (39)$$

where both inversions have $O(n)$ cost since $|\mathbf{D}|^2$ and $\mathbf{I}$ are diagonal, thus products by the matrix in (39) have the $O(n\log n)$ cost associated to the FFT implementation of the products by $\mathbf{U}$ and $\mathbf{U}^H$.

The leading cost of line 7 of PIDAL-FS (given by (38)) will thus be either $O(n\log n)$ or the cost of the products by $\mathbf{W}^H$ and $\mathbf{W}$. As mentioned above, for a large class of choices of frames, matrix-vector products by $\mathbf{W}$ and $\mathbf{W}^H$ have $O(n\log n)$ cost.

From the observations in the previous paragraphs, the computational costs of the lines of PIDAL-FS are the following. Lines 3, 4, 5, 9, 10, 11, 12, 13, 15, and 16 have $O(n)$ cost. Lines 6, 7, 8, and 14 have $O(n\log n)$ cost. Thus the computational cost of PIDAL-FS scales as $O(n\log n)$.

Finally, convergence of PIDAL-FS is addressed by the following corollary of Theorem 1.

Corollary 3: The PIDAL-FS algorithm converges to a minimizer of (13), provided one exists.

Proof: The proof is similar to that of Corollary 2, since all the minimizations involved are solved exactly in closed form. Clearly, matrix $\mathbf{G} = [\mathbf{W}^T\mathbf{K}^T \;\; \mathbf{I} \;\; \mathbf{W}^T]^T$ has full column rank, thus Theorem 1 guarantees convergence to a minimizer of the objective function. ■

VI. EXPERIMENTS

We now report experiments where PIDAL is compared with other state-of-the-art methods, namely those proposed in [12], [19], [37]. All the algorithms are implemented in MATLAB and the experiments are carried out on a PC with a 3.0GHz Intel Core2Extreme CPU, with 4Gb of RAM, under Microsoft Windows Vista. Unless otherwise indicated, we adjust the regularization parameter τ to achieve the highest *improvement in signal-noise-ratio* (ISNR $= 10\log_{10}\left(\|\mathbf{y} - \mathbf{x}\|_2^2/\|\widehat{\mathbf{x}} - \mathbf{x}\|_2^2\right)$). The PIDAL algorithms are initialized as shown in Table I.

According to Theorem 1, ADMM (thus PIDAL) converges for any choice of $\mu > 0$. However, this parameter does influence the speed of the algorithms. To our knowledge, there is work on methods to choose this parameter for optimal speed; in our experiments, we use the following rule of thumb, found to achieve satisfactory results: $\mu = 60\,\tau/M$, where M is the maximum intensity of the original image. We have observed that the results do not change significantly if this parameter is changed to one order of magnitude below or above this choice.

A. *Comparison with [37]*

We begin by comparing with the algorithms (PIDSplit and PIDSplit+) proposed in [37], which (as acknowledged by the authors of [37]) is based on the earlier version of PIDAL-TV [17]. The setup was already described in Section IV-D: the original image is a portion (84×84) of the Cameraman image, scaled to a maximum value of 3000 and blurred with a Gaussian kernel of unit variance; the observed image is generated according to (1); the regularization parameter is set to $\tau = 0.008$. In the experiments reported in [37], the TV denoising step of PIDAL-TV is implemented by an inner iterative algorithm with a tight stopping criterion based on the change between two consecutive images. Our implementation of PIDAL-TV, as explained in Section IV-D, uses a small and fixed number of iterations (just 5) of Chambolle's algorithm, which is initialized as explained in that section. Because PIDSplit and PIDSplit+ have no inner loop, each of its iterations is roughly equivalent to one iteration of PIDAL-TV with just one iteration of Chambolle's algorithm. In [37], PIDSplit and PIDSplit+ were run for 2150 iterations; we thus run PIDAL-TV for $2150/5 = 430$ iterations, corresponding to roughly the same amount of computation. Fig. 6 shows the evolution of the *mean absolute error* (MAE $= \|\widehat{\mathbf{x}} - \mathbf{x}\|_1/n$) and ISNR along the first 160 iterations of PIDAL-TV (as well as elapsed time); it is clear that convergence is achieved after less than 140 iterations (4.3 seconds, in our computer). This is dramatically less than what is reported in [37] for PIDAL-TV; in terms of iterations of PIDSplit and PIDSplit+, this corresponds to approximately $150 * 5 = 750$ iterations, thus also much less than the 2150 iterations (11 seconds) reported in that work. Finally, Fig. 7 shows the original, observed, and restored images; as expected, the image estimates produced by PIDSplit and PIDAL-TV are very similar.

Finally, we also tested PIDAL-FA and PIDAL-FS on the same example, using a fully redundant Haar frame. The plots of ISNR and MAE are presented in Figs. 8 and 9, while the estimated images are shown in Fig. 10.

These results show that, in this example, PIDAL-FA performs slightly better than PIDAL-TV in terms of ISNR and similarly in terms of MAE, with PIDAL-FA achieving its best estimate faster than PIDAL-TV. The synthesis-based criterion implemented by PIDAL-FS is a little worse in terms of both ISNR and MAE, and PIDAL-FS also takes longer than PIDAL-FA to achieve its best estimate. This poorer performance of the synthesis formulation (in line with recent results in [35]) was also found in all the experiments reported below, so we will only present results from PIDAL-TV and PIDAL-FA.

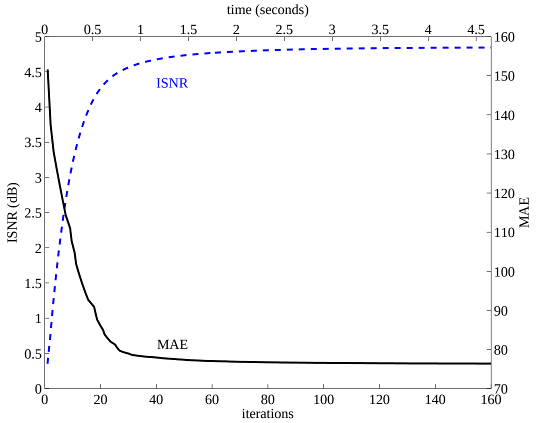

Fig. 6. Evolution of the *mean absolute error* (MAE) and *improvement in signal-noise-ratio* (ISNR) along the iterations and elapsed time of PIDAL-TV, for the experiment of Section VI-A.

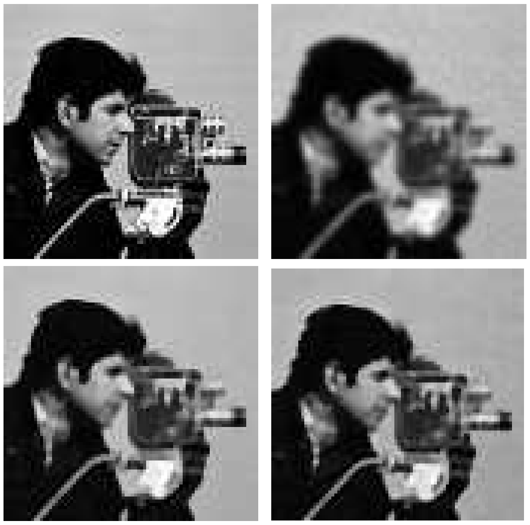

Fig. 7. Experiment of Section VI-A. Top row: original (left); blurred and noisy image (right). Bottom row: estimate from [37]; estimate by PIDAL-TV (ISNR=4.8dB).

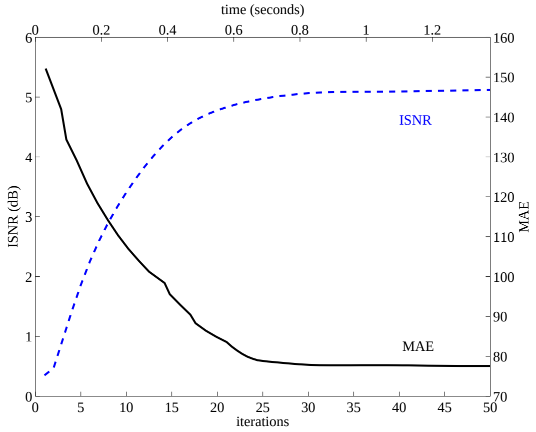

Fig. 8. Experiment of Section VI-A. Evolution of the *mean absolute error* (MAE) and *improvement in signal-noise-ratio* (ISNR) along the iterations and elapsed time of PIDAL-FA.

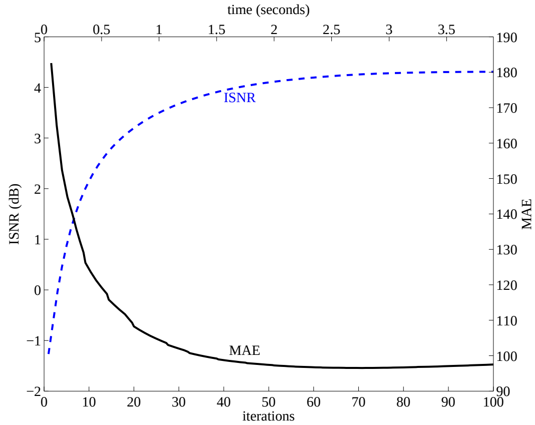

Fig. 9. Experiment of Section VI-A. Evolution of the *mean absolute error* (MAE) and *improvement in signal-noise-ratio* (ISNR) along the iterations and elapsed time of PIDAL-FS.

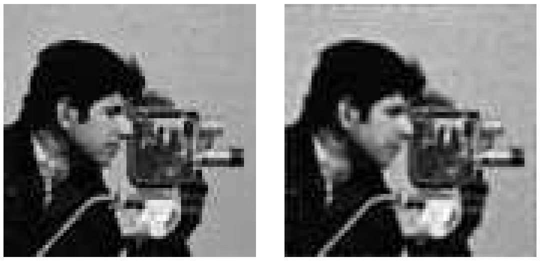

Fig. 10. Experiment of Section VI-A. Left: PIDAL-FA estimate (ISNR = 5.3dB). Right: PIDAL-FS estimate (ISNR = 4.3dB).

B. Comparison with [19]

The next experiment follows [19]: the original image is the complete (256×256) Cameraman, scaled to a maximum value of 17600, the blur is 9×9 uniform. As in the experiment reported in the previous subsection, this is a high SNR situation. Fig. 11 shows the evolution of the MAE and ISNR along the execution of PIDAL-TV; it is clear that convergence is achieved after about 160 iterations (25 seconds, in our computer). A detail of the blurred, and estimated images (from [19] and using PIDAL-TV and PIDAL-FA) are shown in Fig. 12. Although the TV and FA regularizers are considerably simpler than the locally adaptive approximation techniques used in [19], both PIDAL-TV and PIDAL-FA achieve higher ISNR values (7.0dB and 6.95dB, respectively) than that reported in [19] (6.61dB).

C. Comparison with [12]

In the last set of experiments we compare our approach with another recent state-of-the-art algorithm (herein referred to as DFS), proposed in [12], for which the MATLAB implementation is available at `www.greyc.ensicaen.fr/∼fdupe`. That work includes comparisons with other methods, namely: Richardson-Lucy with multi-resolution support wavelet regularization (RL-MRS) [39]; fast translation invariant tree-pruning reconstruction (FTITPR) [44]; Richardson-Lucy with total variation regularization (RL-TV) [11]. The results in [12] show that the algorithm therein proposed generally achieves better performance (*i.e.*, lower MAE) than the others, except for one of the images (a microscopy cell image) where RL-MRS outperforms DFS. For this reason, we will report results comparing PIDAL-TV and PIDAL-FA versus DFS and RL-MRS. For PIDAL-FA, we use a redundant Haar frame for the Cameraman image and Daubechies-4 for the other images. As in [12], the original images are scaled to a maximum value M, belonging to $\{5, 30, 100, 255\}$, and then blurred by a 7×7 uniform filter.

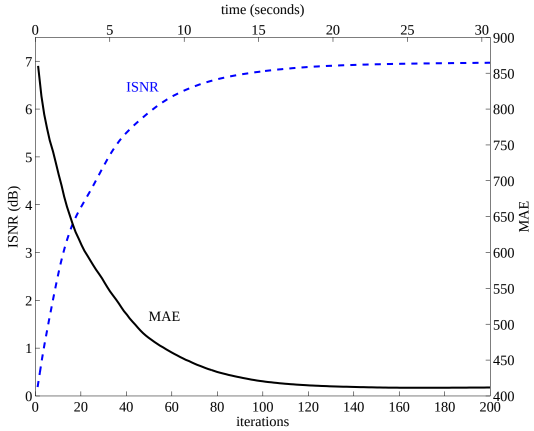

Fig. 11. Experiment of Section VI-B. Evolution of the *mean absolute error* (MAE) and *improvement in signal-noise-ratio* (ISNR) along the iterations and elapsed time of PIDAL-TV.

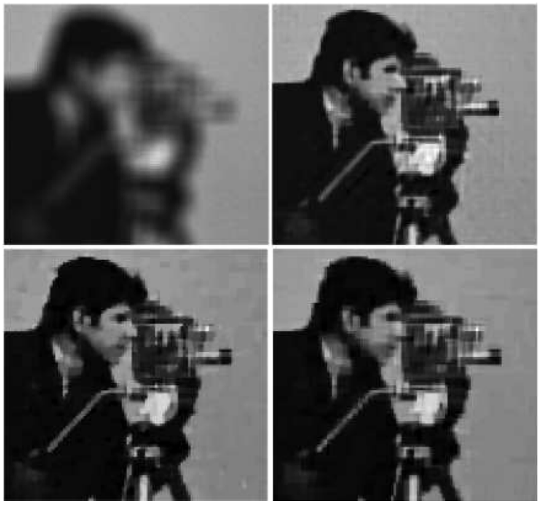

Fig. 12. Experiment of Section VI-B. Top row: blurred noisy image (left) and estimate from [19] (ISNR=6.61dB). Bottom row: PIDAL-TV estimate (left, ISNR = 7.0dB); PIDAL-FA estimate (right, ISNR=6.95dB).

The DFS algorithm does not include a stopping criterion, with the results reported in [12] having been obtained by running a fixed number (200) of iterations. In order to compare the running times of PIDAL-TV, PIDAL-FA, and DFS, we run DFS until the MAE decreases less than 0.01% between two consecutive iterations. Our algorithms are stopped when the following condition is met:

$$\frac{\|\mathbf{z}_k - \mathbf{z}_{k-1}\|_2}{\|\mathbf{z}_{k-1}\|_2} \leq \delta,$$

with $\delta = 0.005$ if $M = 5$ and $\delta = 0.001$ in all the other cases.

Notice that this favors DFS, since a stopping criterion based on MAE is not applicable in practice due to the absence of the original image.

The results reported in Table II show that, in 9 out of the 12 experiments, either PIDAL-TV or PIDAL-FA achieves the lowest MAE. Notice however, that the main goal of this paper was not to introduce a new restoration criterion aiming at obtaining the lowest possible MAE (or any other performance measure), but rather to introduce algorithms to solve the optimization problems resulting from variational formulations of Poissonian image restoration. In terms of computational efficiency, PIDAL-TV and PIDAL-FA are clearly faster than the DRS algorithm, except in the very low SNR situations ($M = 5$) for two of the images (Cameraman and Cell).

VII. Concluding Remarks

We have propose new algorithms to handle the optimization problems resulting from regularization approaches to the restoration of Poissonian images. These optimization problems include several difficulties: the Poisson log-likelihood is non-quadratic and its gradient is not Lipschitz; the state-of-the-art regularizers are non-smooth; there is a non-negativity constraint. We have started by presenting sufficient conditions for existence and uniqueness of solutions of these optimization problems, for the following regularizers: total-variation, frame-based analysis, and frame-based synthesis. These problems were handled by adapting the alternating direction method of multipliers (ADMM) to their particular forms. This adaptation is based on a new way of using ADMM to deal with problems in which the objective function is a linear combination of convex terms, which can be used in many other problems. We gave sufficient conditions for convergence and proved that these are met in the considered cases. Finally, we have experimentally compared the proposed algorithms against competing techniques, showing that they achieve state-of-the-art performance both in terms of speed and restoration accuracy.

Appendix A: Convex Analysis

We very briefly review some basic convex analysis results used in this paper. For more details see [9], [47].

Consider a function $f : \mathcal{X} \to \bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$, where $\bar{\mathbb{R}}$ is called the extended real line, and $\mathcal{X}$ is a real Hilbert space. The domain of function f is the set $\text{dom}(f) = \{\mathbf{x} : f(\mathbf{x}) < +\infty\}$.

The function f is *convex* if $f(\alpha\mathbf{u} + (1-\alpha)\mathbf{v}) \leq \alpha f(\mathbf{u}) + (1-\alpha) f(\mathbf{v})$, for any $\mathbf{u}, \mathbf{v} \in \mathcal{X}$ and any $\alpha \in [0, 1]$. Convexity is said to be strict if the inequality holds strictly ($<$) for any $\mathbf{u}, \mathbf{v} \in \text{dom}(f)$ and $\alpha \in]0, 1[$.

The function is called *proper* if it is not equal to $+\infty$ everywhere and is never equal to $-\infty$.

The function f is *lower semi-continuous* (lsc) at $\mathbf{v}$ if

$$\lim_{\delta \searrow 0} \inf_{\mathbf{x} \in B(\mathbf{v},\delta)} f(\mathbf{x}) \geq f(\mathbf{v}),$$

where $B(\mathbf{v}, \delta) = \{\mathbf{x} : \|\mathbf{x} - \mathbf{v}\| \leq \delta\}$ is the δ-ball around $\mathbf{v}$, and $\|\cdot\|$ is the norm in the Hilbert space $\mathcal{X}$. A function is called lsc if it is lsc at every point of its domain.

A function f is called *coercive* if it verifies $\lim_{\|\mathbf{x}\| \to \infty} f(\mathbf{x}) = +\infty$. Proper, lsc, coercive functions play a key role in optimization via the following theorem [9]:

Theorem 2: If f is a proper, lsc, coercive, convex function, then the set $\arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ is nonempty.

The next theorem concerns strictly convex functions.

Theorem 3: If f is a strictly convex function, the set $\arg\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$ possesses at most one element.

References

[1] A. Beck, M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”, *SIAM Journal on Imaging Sciences*, vol. 2, pp. 183–202, 2009.

[2] J. Bioucas-Dias, M. Figueiredo, “A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration”, *IEEE Trans. on Image Proc.*, vol. 16, pp. 2992-3004, 2007.

[3] E. Candès, L. Demanet, D. Donoho, and L. Ying, “Fast discrete curvelet transforms”, *Multiscale Modelling and Simulation*, vol. 5, pp. 861–899, 2005.

[4] A. Chambolle, “An algorithm for total variation minimization and applications,” *Jour. Math. Imaging and Vision*, vol. 20, pp. 89-97, 2004.

[5] A. Chambolle, P.-L. Lions, “Image recovery via total variation minimization and related problems,” *Numerische Mathematik*, vol. 76, pp. 167–188, 1997.

[6] T. Chan, S. Esedoglu, F. Park, A. Yip, “Recent developments in total variation image restoration,” in *Mathematical Models of Computer Vision*, Springer, 2005.

[7] T. Chan, G. Golub, P. Mulet, “A nonlinear primal-dual method for total variation-based image restoration”, *SIAM Jour. Sci. Comput.*, vol. 20, pp. 1964–1977, 1999.

[8] P. Combettes, J.-C. Pesquet, “A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery,” *IEEE Jour. of Selec. Topics in Signal Proc.*, vol. 1, pp. 564-574, 2007.

[9] P. Combettes, V. Wajs, “Signal recovery by proximal forward-backward splitting,” *SIAM Jour. Multiscale Modeling & Simulation*, vol. 4, pp. 1168–1200, 2005.

[10] J. Dahl, P. Hansen, S. Jensen, T. Jensen, “Algorithms and software for total variation image reconstruction via first-order methods”, *Numerical Algorithms*, 2009.

[11] N. Dey, L. Blanc-Feraud, C. Zimmer, P. Roux, Z. Kam, J. Olivo-Marin, J. Zerubia, “Richardson-Lucy algorithm with TV regularization for 3D confocal microscope deconvolution”, *Microscopy Research and Tech.*, vol. 69, pp. 260-266, 2006.

[12] F.-X. Dupé, J. Fadili, J.-L. Starck, “A proximal iteration for deconvolving Poisson images using sparse representations”, *IEEE Trans. on Image Proc.*, vol. 18, pp. 310–321, 2009.

[13] J. Eckstein, D. Bertsekas, “On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators”, *Math. Program.*, vol. 5, pp. 293-318, 1992.

[14] M. Elad, P. Milanfar, and R. Rubinstein, “Analysis versus synthesis in signal priors”, *Inverse Problems*, vol. 23, pp. 947–968, 2007.

[15] E. Esser, “Applications of Lagrangian-based alternating direction methods and connections to split Bregman”, Tech. Rep. 09-31, Computational and Applied Math., Univ. of California, Los Angeles, 2009.

[16] J. Fessler, A. Hero III, “Penalized maximum-likelihood image reconstruction using space-alternating generalized EM algorithms”, *IEEE Trans. Image Proc.*, vol. 4, pp. 1417–29, 1995.

[17] M. Figueiredo, J. Bioucas-Dias, “Deconvolution of Poissonian images using variable splitting and augmented Lagrangian optimization”, *IEEE Workshop on Statistical Signal Processing*, Cardiff, UK, 2009.

TABLE II

RESULTS OF THE COMPARISON OF PIDAL-TV AND PIDAL-FA WITH THE ALGORITHMS PROPOSED IN [12] AND [39] (AVERAGE OVER 10 RUNS); THE REPORTED TIMES ARE IN SECONDS.

		PIDAL-TV			PIDAL-FA			DFS [12]			RL-MRS [39]
Image	M	MAE	iterations	time	MAE	iterations	time	MAE	iterations	time	MAE
Cameraman	5	0.27	120	22	**0.26**	70	13	0.35	6	4.5	0.37
Cameraman	30	1.29	51	9.1	**1.22**	39	7.4	1.47	98	75	2.06
Cameraman	100	3.99	33	6.0	**3.63**	36	6.8	4.31	426	318	5.58
Cameraman	255	8.99	32	5.8	**8.45**	37	7.0	10.26	480	358	12.3
Neuron	5	**0.17**	117	3.6	0.18	66	2.9	0.19	6	3.9	0.19
Neuron	30	**0.68**	54	1.8	0.77	44	2.0	0.82	161	77	0.95
Neuron	100	**1.75**	43	1.4	2.04	41	1.8	2.32	427	199	2.88
Neuron	255	3.52	43	1.4	**3.47**	42	1.9	5.25	202	97	6.31
Cell	5	0.12	56	10	**0.11**	36	7.6	0.12	6	4.5	0.12
Cell	30	0.57	31	6.5	0.54	39	8.2	0.56	85	64	**0.47**
Cell	100	1.71	85	15	1.46	31	6.4	1.72	215	162	**1.37**
Cell	255	3.77	89	17	3.33	34	7.0	5.45	410	308	**3.10**

[18] M. Figueiredo, R. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. on Image Proc.*, vol. 12, pp. 906–916, 2003.

[19] A. Foi, S. Alenius, M. Trimeche, V. Katkovnik, K. Egiazarian, "A spatially adaptive Poissonian image deblurring, *Proc. IEEE Int. Conf. Image Proc.*, Genova, 2005.

[20] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite-element approximations", *Computers and Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[21] R. Glowinski and A. Marroco, "Sur l'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité d'une classe de problemes de Dirichlet non lineares", *Revue Française d'Automatique, Informatique et Recherche Opérationelle*, vol. 9, pp. 41–76, 1975.

[22] T. Goldstein, S. Osher, "The split Bregman method for L1 regularized problems", Tech. Rep. 08-29, Computational and Applied Math., Univ. of California, Los Angeles, 2008.

[23] M. Hestenes, "Multiplier and Gradient Methods", *Journal of Optimization Theory and Applications*, vol. 4, pp. 303-320, 1969.

[24] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals", *Journal of Applied and Computational Harmonic Analysis,* vol. 10, pp. 234-253, 2001.

[25] M. Lang, H. Guo, J. Odegard, C. Burrus, and R. Wells. "Noise reduction using an undecimated discrete wavelet transform," *IEEE Signal Processing Letters*, vol. 3, pp. 10–12, 1996.

[26] T. Le, R. Chartrand, and T. Asaki. "A variational approach to reconstructing images corrupted by Poisson noise", *Journal of Mathematical Imaging and Vision*, vol. 27, pp. 257–263, 2007.

[27] M. Lustig, D. Donoho, and J. Pauly. "Sparse MRI: the application of compressed sensing for rapid MR imaging," *Magnetic Resonance in Medicine*, vol. 58, pp. 1182–1195, 2007.

[28] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 2009.

[29] J. Nocedal, S. Wright, *Numerical Optimization*, Springer, 2006.

[30] R. Nowak and E. Kolaczyk. "A Bayesian multiscale framework for Poisson inverse problems", *IEEE Transactions on Information Theory,* vol. 46, pp. 1811–1825, 2000.

[31] S. Osher, L. Rudin, E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[32] M. Powell, "A Method for nonlinear Constraints in Minimization Problems", in *Optimization*, R. Fletcher, (editor), Academic Press, pp. 283–298, New York, 1969.

[33] P. Sarder and A. Nehorai. "Deconvolution method for 3-D fluorescence microscopy images", *IEEE Signal Processing Magazine,* vol. 23, pp. 32-45, 2006.

[34] I. Selesnick, "Hilbert transform pairs of wavelet bases", *IEEE Signal Processing Letters,* vol. 8, pp. 170-173, 2001.

[35] I. Selesnick and M. Figueiredo, "Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors", *Proceedings of SPIE*, vol. 7446 (Wavelets XIII), 2009.

[36] S. Setzer, "Split Bregman algorithm, Douglas-Rachford splitting, and frame shrinkage", *Proc. 2nd Intern. Conf. on Scale Space and Variat. Meth. in Computer Vision,* Springer, 2009.

[37] S. Setzer, G. Steidl, and T. Teuber, "Deblurring Poissonian images by split Bregman techniques", *Journal of Visual Communication and Image Representation,* 2009 (accepted).

[38] J.-L. Starck, F. Murtagh, *Astronomical Image and Data Analysis*, Springer, 2006.

[39] J.-L. Starck, A. Bijaoui, F. Murtagh, "Multiresolution support applied to image filtering and deconvolution", *Computer Vision, Graphics, and Image Processing: Graphical Models and Image Processing*, vol. 57, pp. 420–431, 1995.

[40] J.-L. Starck, M. Nguyen, F. Murtagh, "Wavelets and curvelets for image deconvolution: a combined approach", *Signal Proc.*, vol. 83, pp. 2279–2283, 2003.

[41] K. Timmermann and R. Nowak. "Multiscale modeling and estimation of Poisson processes with application to photon-limited imaging", *IEEE Transactions on Information Theory,* vol. 45, pp. 846–862, 1999.

[42] Y. Wang, J. Yang, W. Yin, Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction", *SIAM Jour. Imaging Sciences*, vol. 1, pp. 248–272, 2008.

[43] R. Willett and R. Nowak. "Platelets: a multiscale approach for recovering edges and surfaces in photon-limited medical imaging", *IEEE Transactions on Medical Imaging,* vol. 22, pp. 332–350, 2003.

[44] R. Willett and R. Nowak. "Fast multiresolution photon-limited image reconstruction, *IEEE International Symposium on Biomedical Imaging,* 2004.

[45] S. Wright, R. Nowak, M. Figueiredo, "Sparse reconstruction by separable approximation, *IEEE Trans. Signal Proc.*, vol. 57, pp. 2479-2493, 2009.

[46] W. Yin, S. Osher, D. Goldfarb, J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing", *SIAM Jour. Imaging Sciences*, vol. 1, pp. 143–168, 2008.

[47] C. Zalinescu, *Convex Analysis in General Vector Spaces,* World Scientific Publishing, Singapore, 2002.