

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/260709570>

Deconvolving Images With Unknown Boundaries Using the Alternating Direction Method of Multipliers

Article in IEEE Transactions on Image Processing · August 2013

DOI: 10.1109/TIP.2013.2258354

CITATIONS

142

READS

229

2 authors:



Mariana Almeida

Institute of Telecommunications

13 PUBLICATIONS 612 CITATIONS

SEE PROFILE



Mário A T Figueiredo

University of Lisbon

351 PUBLICATIONS 24,778 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Conference Chair [View project](#)



Image Restoration with Class-Adapted Priors [View project](#)

Deconvolving Images with Unknown Boundaries Using the Alternating Direction Method of Multipliers

Mariana S. C. Almeida and Mário A. T. Figueiredo, *Fellow, IEEE*

Abstract—The *alternating direction method of multipliers* (ADMM) has recently sparked interest as a flexible and efficient optimization tool for imaging inverse problems, namely deconvolution and reconstruction under non-smooth convex regularization. ADMM achieves state-of-the-art speed by adopting a *divide and conquer* strategy, wherein a hard problem is split into simpler, efficiently solvable sub-problems (e.g., using fast Fourier or wavelet transforms, or simple proximity operators). In deconvolution, one of these sub-problems involves a matrix inversion (i.e., solving a linear system), which can be done efficiently (in the discrete Fourier domain) if the observation operator is circulant, i.e., under periodic boundary conditions. This paper extends ADMM-based image deconvolution to the more realistic scenario of unknown boundary, where the observation operator is modeled as the composition of a convolution (with arbitrary boundary conditions) with a spatial mask that keeps only pixels that do not depend on the unknown boundary. The proposed approach also handles, at no extra cost, problems that combine the recovery of missing pixels (i.e., inpainting) with deconvolution. We show that the resulting algorithms inherit the convergence guarantees of ADMM and illustrate its performance on non-periodic deblurring (with and without inpainting of interior pixels) under total-variation and frame-based regularization.

Index Terms—Image deconvolution, alternating direction method of multipliers (ADMM), boundary conditions, non-periodic deconvolution, inpainting, total variation, frames.

I. INTRODUCTION

The *alternating direction method of multipliers* (ADMM), originally proposed in the 1970's [25], emerged recently as a flexible and efficient tool for several imaging inverse problems, such as denoising [23], [38], deblurring [2], inpainting [2], reconstruction [34], motion segmentation [44], to mention only a few classical problems (for a comprehensive review, see [9]). ADMM-based approaches make use of variable splitting, which allows a straightforward treatment of various priors/regularizers [1], such as those based on frames or on *total-variation* (TV) [36], as well as the seamless inclusion of several types of constraints (e.g., positivity) [23], [38]. ADMM is closely related to other techniques, namely the so-called *Bregman* and *split Bregman* methods [10], [26], [43] and *Douglas-Rachford splitting* [9], [14], [19], [21].

Several ADMM-based algorithms for imaging inverse problems require, at each iteration, solving a linear system (equivalently, inverting a matrix) [2], [10], [23], [33], [39]. This fact is simultaneously a blessing and a curse. On the one hand, the matrix to be inverted is related to the Hessian of

the objective function, thus carrying second order information; arguably, this fact justifies the excellent speed of these methods, which have been shown (see, e.g., [2]) to be considerably faster than the classical *iterative shrinkage-thresholding* (IST) algorithms [16], [22] and even than their accelerated versions [7], [8], [42]. On the other hand, this inversion (due to its typically huge size) limits its applicability to problems where it can be efficiently computed (by exploiting some particular structure). For ADMM-based image deconvolution algorithms [2], [10], [23], [39], this inversion can be efficiently carried out using the *fast Fourier transform* (FFT), if the convolution is cyclic/periodic (or assumed to be so), thus diagonal in the discrete Fourier domain. However, as explained next, periodicity is an unnatural assumption, inadequate for most real imaging problems.

In deconvolution, the pixels located near the boundary of the observed image depend on pixels (of the unknown image) located outside of its domain. The typical way to formalize this issue is to adopt a so-called *boundary condition* (BC).

- The **periodic** BC (the use of which, in image deconvolution, dates back to the 1970s [6]) assumes a periodic convolution; its matrix representation is circulant¹, diagonalized by the DFT¹, which can be implemented via the FFT. This computational convenience makes it, arguably, the most commonly adopted BC.
- The **zero** BC assumes that all the external pixels have zero value, thus the matrix representing the convolution is block-Toeplitz, with Toeplitz blocks [31]. By analogy with the BC for ordinary or partial differential equations that assumes fixed values at the domain boundary, this is commonly referred to Dirichlet BC [31].
- In the **reflexive** and **anti-reflexive** BCs, the pixels outside the image domain are a reflection of those near the boundary, using even or odd symmetry, respectively [13], [18], [32]. In the reflexive BC, the discrete derivative at the boundary is zero; thus, by analogy with the BC for ordinary or partial differential equations that assumes fixed values of the derivative at the boundary, the reflexive BC is often referred to as Neumann BC [31].

As illustrated in Figure 1, all these standard BCs are quite unnatural, as they do not correspond to any realistic imaging system, being motivated merely by computational convenience. Namely, assuming a periodic boundary condition has the advantage of allowing a very fast implementation of the convolution as a point-wise multiplication in the DFT

Both authors are with the Instituto de Telecomunicações (IT), Instituto Superior Técnico (IST), 1049-001 Lisboa, Portugal. Email: {mariana.almeida,mario.figueiredo}@lx.it.pt

This work was partially supported by *Fundação para a Ciência e Tecnologia* (FCT), under grants PTDC/EEA-TEL/104515/2008 and PEst-OE/EEI/LA0008/2011, and the fellowship SFRH/BPD/69344/2010.

¹In fact, when dealing with 2-dimensional (2D) images, these matrices are block-circulant with circulant blocks, thus diagonalized by the 2D *discrete Fourier transform* (DFT). For simplicity, we refer to such matrices simply as circulant and to the 2D DFT simply as DFT.

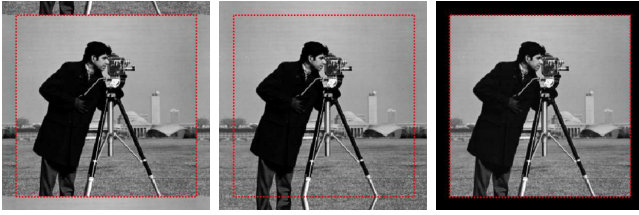


Fig. 1. Illustration of the (unnatural) assumptions underlying the periodic, reflexive, and zero boundary conditions.

domain, efficiently implementable using the FFT. However, in real problems, there is no reason for the external (unobserved) pixels to follow periodic (or any other) boundary conditions. A well known consequence of this mismatch is a degradation of the deconvolved images, such as the appearance of ringing artifacts emanating from the boundaries (see Figs. 2 and 3). These artifacts can be reduced by pre-processing the image to reduce the spurious discontinuities at the boundaries, created by the (wrong) periodicity assumption; this is what is done by the “edgetaper” function in the *MATLAB Image Processing Toolbox*. In a more sophisticated version of this idea that was recently proposed, the observed image is extrapolated to create a larger image with smooth BCs [27].

Convolutions under zero BC can still be efficiently performed in the DFT domain, by embedding (using zero-padding) the non-periodic convolution into a larger periodic one [31]. However, in addition to the mismatch problem pointed out in the previous paragraph (*i.e.*, there is usually no reason to assume that the boundary is zero), this choice has another drawback: in the context of ADMM-based methods, the zero-padding technique prevents the required matrix inversion from being efficiently computed via the FFT [4].

A. Related Work and Contributions

Instead of adopting a standard BC or a boundary smoothing scheme, a more realistic model of actual imaging systems treats the external boundary pixels as unknown; *i.e.*, the problem is seen as one of simultaneous deconvolution and inpainting, where the unobserved boundary pixels are estimated together with the deconvolved image. The corresponding observation model is the composition of a spatial mask with a convolution under arbitrary BC [13], [29], [35], [40]; addressing this formulation using ADMM is the central theme of this paper. While we were concluding this manuscript, we became aware of very recent related work in [30].

Under quadratic (Tikhonov) regularization, image deconvolution with periodic BC corresponds to a linear system, where the corresponding matrix can be efficiently inverted in the DFT domain using the FFT. In the unknown boundary case, it was shown in [35] how this inversion can be reduced to an FFT-based inversion followed by the solution (via, *e.g.*, *conjugate gradient* – CG) of a much smaller system (same dimension as the unknown boundary). Very recently, [40] adapted this technique to deconvolution under TV and frame-based analysis non-smooth regularization; that work proposes an algorithm based on variable splitting and quadratic penalization, using the method of [35] to solve the linear system at each iteration.

That method is related to, but it is not ADMM, thus has no guarantees of converge to a minimizer of the original objective function. Although [40] mentions the possibility of using the method of [35] within ADMM, that option was not explored.

In this paper, we address image deconvolution with unknown boundaries, under TV-based and frame-based (synthesis and analysis) non-smooth regularization, using ADMM. Our first and main approach exploits the fact that the observation model involves the composition of a spatial mask with a periodic convolution, and uses ADMM in a way that decouples these two operations. The resulting algorithms inherit all the desirable properties of previous ADMM-based deconvolution methods: all the update equations (including the matrix inversion) can be computed efficiently without using inner iterations; convergence is formally guaranteed. The second approach considered is the direct extension of the methods from [2], [10], [23], [39] to the unknown boundary case; here (unlike with periodic BC), the linear system at each iteration cannot be solved efficiently using the FFT, thus we adopt the technique of [35], [40]. However, since the resulting algorithms are instances of ADMM, they have convergence guarantees, which are missing in [40]. Furthermore, we show how the method of [35] can also be used in ADMM for frame-based synthesis regularization, whereas [40] only considers analysis formulations.

The proposed methods are experimentally illustrated using frame-based and TV-based regularization; the results show the advantage of our approach over the use of the “edgetaper” function (in terms of improvement in SNR) and over an adaptation of the recent strategy of [40] (in terms of speed). Finally, our approach is also tested on inverse problems where the observation model consists of a (non-cyclic) deblurring followed by a generic loss of pixels (inpainting problems). Arguably due to its complexity, this problems has not been often addressed in the literature, although it is a relevant one: consider the problem of deblurring an image in which some pixel values are inaccessible, *e.g.*, because they are saturated, thus should not be taken into account, or because they correspond to so-called *dead pixels* in the image sensor.

B. Outline and Notation

Sections II and III review the ADMM and its use for image deconvolution with periodic BC, using frame-based and TV-based regularization, setting the stage for the proposed approaches for the unknown boundary case, which are introduced in Section IV. The experimental evaluation of our methods is reported in Section V, which also illustrates its use for simultaneous deblurring and inpainting. Section VI concludes the manuscript.

We use the following notation: $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ is the extended real line; bold lower case denotes vectors (*e.g.*, \mathbf{x} , \mathbf{y}), and bold upper case (Roman or Greek) are matrices (*e.g.*, \mathbf{A} , \mathbf{Y}); the superscript $(\cdot)^*$ denotes vector/matrix transpose, or conjugate transpose in the complex case; \mathbf{I} , $\mathbf{1}$, and $\mathbf{0}$ are the identity matrix and vectors with all elements equal to one and zero, respectively; as usual, \circ denotes function composition (*e.g.*, $(f \circ \mathbf{A})(\mathbf{x}) = f(\mathbf{Ax})$), and \odot the component-wise product between vectors $((\mathbf{v} \odot \mathbf{u})_i = v_i u_i)$.

II. ALTERNATING DIRECTION METHOD OF MULTIPLIERS

A. The Standard ADMM Algorithm

Consider an unconstrained optimization problem

$$\min_{\mathbf{z} \in \mathbb{R}^n} f(\mathbf{z}) + g(\mathbf{G}\mathbf{z}), \quad (1)$$

where $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ and $g : \mathbb{R}^p \rightarrow \bar{\mathbb{R}}$ are convex functions, and $\mathbf{G} \in \mathbb{R}^{p \times n}$ is a matrix. The ADMM algorithm for this problem (which can be derived from different perspectives, namely, augmented Lagrangian and Douglas-Rachford splitting) is presented in Algorithm 1; for a recent comprehensive review, see [9]. Convergence of ADMM was shown in [19], where the following theorem was proved.

Theorem 1. Consider problem (1), where $\mathbf{G} \in \mathbb{R}^{p \times n}$ has full column rank and $f : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$ and $g : \mathbb{R}^p \rightarrow \bar{\mathbb{R}}$ are closed, proper, convex functions. Consider arbitrary $\mu > 0$, $\mathbf{u}_0, \mathbf{d}_0 \in \mathbb{R}^p$. Let $\eta_k \geq 0$ and $\rho_k \geq 0$, for $k = 0, 1, \dots$, be two sequences such that $\sum_{k=0}^{\infty} \eta_k < \infty$ and $\sum_{k=0}^{\infty} \rho_k < \infty$. Consider three sequences $\mathbf{z}_k \in \mathbb{R}^n$, $\mathbf{u}_k \in \mathbb{R}^p$, and $\mathbf{d}_k \in \mathbb{R}^p$, for $k = 0, 1, \dots$, satisfying

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

Then, if (1) has a solution, say \mathbf{z}^* , the sequence $\{\mathbf{z}_k\}$ converges to \mathbf{z}^* . If (1) does not have a solution, at least one of the sequences $(\mathbf{u}_1, \mathbf{u}_2, \dots)$ or $(\mathbf{d}_1, \mathbf{d}_2, \dots)$ diverges.

Algorithm 1: ADMM

1 Initialization: set $k = 0$, choose $\mu > 0$, \mathbf{u}_0 , and \mathbf{d}_0 .

2 **repeat**

3 $\mathbf{z}_{k+1} \leftarrow \arg \min_{\mathbf{z}} f(\mathbf{z}) + \frac{\mu}{2} \|\mathbf{G}\mathbf{z} - \mathbf{u}_k - \mathbf{d}_k\|_2^2$

4 $\mathbf{u}_{k+1} \leftarrow \arg \min_{\mathbf{u}} g(\mathbf{u}) + \frac{\mu}{2} \|\mathbf{G}\mathbf{z}_{k+1} - \mathbf{u} - \mathbf{d}_k\|_2^2$

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

There are more recent results on ADMM, namely establishing linear convergence [17]; however, those results make stronger assumptions (such as strong convexity) which are not generally satisfied in deconvolution problems. Under the conditions of *Theorem 1*, convergence holds for any $\mu > 0$; however, the choice of μ may strongly affect the convergence speed [9]. It is also possible to replace the scalar μ by a positive diagonal matrix $\Upsilon = \text{diag}(\mu_1, \dots, \mu_p)$, i.e., to replace the quadratic terms of the form $\mu \|\mathbf{G}\mathbf{z} - \mathbf{u} - \mathbf{d}\|_2^2$ by $(\mathbf{G}\mathbf{z} - \mathbf{u} - \mathbf{d})^* \Upsilon (\mathbf{G}\mathbf{z} - \mathbf{u} - \mathbf{d})$.

B. The ADMM For Two or More Functions

Following the formulation proposed in [3], [23], consider a generalization of (1), with $J \geq 2$ functions,

$$\min_{\mathbf{z} \in \mathbb{R}^n} \sum_{j=1}^J g_j(\mathbf{H}^{(j)}\mathbf{z}), \quad (2)$$

where $g_j : \mathbb{R}^{p_j} \rightarrow \bar{\mathbb{R}}$ are proper, closed, convex functions, and $\mathbf{H}^{(j)} \in \mathbb{R}^{p_j \times n}$ are arbitrary matrices. We map this problem into form (1) as follows: let $f(\mathbf{z}) = 0$; define matrix \mathbf{G} as

$$\mathbf{G} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(J)} \end{bmatrix} \in \mathbb{R}^{p \times n}, \quad (3)$$

where $p = p_1 + \dots + p_J$, and let $g : \mathbb{R}^p \rightarrow \bar{\mathbb{R}}$ be defined as

$$g(\mathbf{u}) = \sum_{j=1}^J g_j(\mathbf{u}^{(j)}), \quad (4)$$

where each $\mathbf{u}^{(j)} \in \mathbb{R}^{p_j}$ is a p_j -dimensional sub-vector of \mathbf{u} , that is, $\mathbf{u} = [(\mathbf{u}^{(1)})^*, \dots, (\mathbf{u}^{(J)})^*]^*$.

The definitions in the previous paragraph lead to an ADMM for problem (2) with the following two key features.

- 1) The fact that $f(\mathbf{z}) = 0$ turns line 3 of Algorithm 1 into an unconstrained quadratic problem. Letting Υ be a p -dimensional positive block diagonal matrix (associating a possibly different parameter μ_j to each function g_j)

$$\Upsilon = \text{diag}(\underbrace{\mu_1, \dots, \mu_1}_{p_1 \text{ elements}}, \dots, \underbrace{\mu_j, \dots, \mu_j}_{p_j \text{ elements}}, \dots, \underbrace{\mu_J, \dots, \mu_J}_{p_J \text{ elements}}), \quad (5)$$

the solution of this quadratic problem is given by (denoting $\zeta = \mathbf{u}_k + \mathbf{d}_k$)

$$\begin{aligned} \arg \min_{\mathbf{z}} (\mathbf{G}\mathbf{z} - \zeta)^* \Upsilon (\mathbf{G}\mathbf{z} - \zeta) &= (\mathbf{G}^* \Upsilon \mathbf{G})^{-1} \mathbf{G}^* \Upsilon \zeta \\ &= \left[\sum_{j=1}^J \mu_j (\mathbf{H}^{(j)})^* \mathbf{H}^{(j)} \right]^{-1} \sum_{j=1}^J \mu_j (\mathbf{H}^{(j)})^* \zeta^{(j)}, \quad (6) \end{aligned}$$

with $\zeta^{(j)} = \mathbf{u}_k^{(j)} + \mathbf{d}_k^{(j)}$, where $\zeta^{(j)}$, $\mathbf{u}_k^{(j)}$, and $\mathbf{d}_k^{(j)}$, for $j = 1, \dots, J$, are the sub-vectors of ζ , \mathbf{u}_k , and \mathbf{d}_k , respectively, corresponding to the partition in (3), and the second equality results from the block structure of matrices \mathbf{G} (in (3)) and Υ (in (5)).

- 2) The separable structure of g (as defined in (4)) allows decoupling the minimization in line 4 of Algorithm 1 into J independent minimizations, each of the form

$$\mathbf{u}_{k+1}^{(j)} \leftarrow \arg \min_{\mathbf{v} \in \mathbb{R}^{p_j}} g_j(\mathbf{v}) + \frac{\mu_j}{2} \|\mathbf{v} - \mathbf{s}^{(j)}\|_2^2, \quad (7)$$

for $j = 1, \dots, J$, where $\mathbf{s}^{(j)} = \mathbf{H}^{(j)}\mathbf{z}_{k+1} - \mathbf{d}_k^{(j)}$. This minimization defines to the so-called *Moreau proximity operator* of g_j/μ (denoted as prox_{g_j/μ_j}) (see [14], [15] and references therein) applied to $\mathbf{s}^{(j)}$, thus

$$\mathbf{u}_{k+1}^{(j)} \leftarrow \text{prox}_{g_j/\mu_j}(\mathbf{s}^{(j)}) \equiv \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{x} - \mathbf{s}^{(j)}\|_2^2 + \frac{g_j(\mathbf{x})}{\mu_j}.$$

For some functions, the Moreau proximity operators are

known in closed form [14]; a well-known case is the ℓ_1 norm ($\tau\|\mathbf{x}\|_1 = \tau \sum_i |x_i|$), for which the proximity operator is the soft-threshold function:

$$\text{soft}(v, \gamma) = \text{sign}(v) \odot \max\{|v| - \gamma, 0\}, \quad (8)$$

where the sign, max, and absolute value functions are applied in component-wise fashion.

The convergence of the resulting instance of ADMM (shown in Algorithm 2) is the subject of the following proposition.

Proposition 1. *Consider problem (2), where $g_j : \mathbb{R}^{p_j} \rightarrow \bar{\mathbb{R}}$ are proper, closed, convex functions, and $\mathbf{H}^{(j)} \in \mathbb{R}^{p_j \times n}$. Consider positive constants $\mu_1, \dots, \mu_J > 0$ and arbitrary $\mathbf{u}_0, \mathbf{d}_0 \in \mathbb{R}^p$. If matrix \mathbf{G} has full column rank and (2) has a solution, say \mathbf{z}^* , then, the sequence $(\mathbf{z}_1, \mathbf{z}_2, \dots)$ generated by Algorithm 2 converges to \mathbf{z}^* .*

Proof: We simply need to show that the conditions of Theorem 1 are satisfied. Problem (2) has the form (1), with $f(\mathbf{z}) = 0$ and g as given by (4); if all the functions g_1, \dots, g_J are closed, proper, and convex, so are g and f . Furthermore, if \mathbf{G} has full column rank, a sufficient condition for the inverse in (6) to exist is that $\mu_1, \dots, \mu_J > 0$. Finally, the minimization in line 3 of Algorithm 1 is solved exactly in line 4 of Algorithm 2, and the minimization in line 4 of Algorithm 1 is solved exactly in line 6 of Algorithm 2. Thus, we can take the sequences η_k and ρ_k in Theorem 1 to be identically zero. ■

Algorithm 2: ADMM-2

```

1 Initialization: set  $k = 0$ , choose  $\mu_1, \dots, \mu_J > 0$ ,  $\mathbf{u}_0, \mathbf{d}_0$ .
2 repeat
3    $\boldsymbol{\zeta} \leftarrow \mathbf{u}_k + \mathbf{d}_k$ 
4    $\mathbf{z}_{k+1} \leftarrow \left( \sum_{j=1}^J \mu_j (\mathbf{H}^{(j)})^* \mathbf{H}^{(j)} \right)^{-1} \sum_{j=1}^J \mu_j (\mathbf{H}^{(j)})^* \boldsymbol{\zeta}^{(j)}$ 
5   for  $j = 1$  to  $J$  do
6      $\mathbf{u}_{k+1}^{(j)} \leftarrow \text{prox}_{g_j/\mu_j}(\mathbf{H}^{(j)} \mathbf{z}_{k+1} - \mathbf{d}_k^{(j)})$ 
7      $\mathbf{d}_{k+1}^{(j)} \leftarrow \mathbf{d}_k^{(j)} - (\mathbf{H}^{(j)} \mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(j)})$ 
8   end
9    $k \leftarrow k + 1$ 
10 until stopping criterion is satisfied
```

If the proximity operators of the functions g_j are simple, the computational bottleneck of Algorithm 2 is the matrix inversion in line 4. As shown in [39], [41], inversions of this form can be efficiently tackled using the FFT, if all the matrices $\mathbf{H}^{(j)}$ are circulant, thus jointly diagonalized by the DFT (more details in Section III). Due to this condition, the work in [39], [41] was limited to deconvolution with periodic BCs, under TV regularization. More recent work in [2], [3], [23] has shown how these inversions can still be efficiently handled via the FFT (and other fast transforms) in problems such as image reconstruction from partial Fourier observations and inpainting, and with other regularizers, such as those based on tight frames. This paper extends that work, showing how to handle deconvolution problems with unknown boundaries.

III. IMAGE DECONVOLUTION WITH PERIODIC BC

This section reviews the ADMM-based approach to image deconvolution with periodic BC, using the frame-based formulations and TV-based regularization [2], [3], [23], [36], [39], the standard regularizers for this class of imaging inverse problems. The next section will then show how this approach can be extended to deal with unknown boundaries.

A. Frame-Based Synthesis Formulation

There are two standard ways to formulate frame-based regularization for image deconvolution, both exploiting the fact that natural images have sparse representations on wavelet² frames: the synthesis and analysis formulations [2], [20], [37].

The synthesis formulation expresses the estimated image $\hat{\mathbf{x}}$ as a linear combination of the elements of some wavelet frame (an orthogonal basis or an overcomplete dictionary), i.e., $\hat{\mathbf{x}} = \mathbf{W}\hat{\mathbf{z}}$, with $\hat{\mathbf{z}}$ given by

$$\hat{\mathbf{z}} = \arg \min_{\mathbf{z} \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{W}\mathbf{z}\|_2^2 + \lambda \phi(\mathbf{z}), \quad (9)$$

where $\mathbf{y} \in \mathbb{R}^n$ is a vector containing all the pixels (in lexicographic order) of the observed image, $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a matrix representation of the (periodic) convolution, the columns of matrix $\mathbf{W} \in \mathbb{R}^{n \times d}$ ($d \geq n$) are the elements of the adopted frame, ϕ is a regularizer encouraging $\hat{\mathbf{z}}$ to be sparse (a typical choice, herein adopted, is $\phi(\mathbf{z}) = \|\mathbf{z}\|_1$), and $\lambda > 0$ is the regularization parameter [2], [20], [24]. The first term in (9) results from assuming the usual observation model

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n} \quad (10)$$

where $\mathbf{x} = \mathbf{W}\mathbf{z}$, and \mathbf{n} is a sample of white Gaussian noise with unit variance (any other value may be absorbed by λ).

Clearly, problem (9) has the form (2), with $J = 2$ and

$$g_1 : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}, \quad g_1(\mathbf{v}) = \frac{1}{2} \|\mathbf{y} - \mathbf{v}\|_2^2 \quad (11)$$

$$g_2 : \mathbb{R}^d \rightarrow \bar{\mathbb{R}}, \quad g_2(\mathbf{z}) = \lambda \|\mathbf{z}\|_1 \quad (12)$$

$$\mathbf{H}^{(1)} \in \mathbb{R}^{n \times d}, \quad \mathbf{H}^{(1)} = \mathbf{A}\mathbf{W} \quad (13)$$

$$\mathbf{H}^{(2)} \in \mathbb{R}^{d \times d}, \quad \mathbf{H}^{(2)} = \mathbf{I}. \quad (14)$$

The proximity operators of g_1 and g_2 , key components of Algorithm 2 for solving (9), have simple expressions,

$$\text{prox}_{g_1/\mu_1}(\mathbf{v}) = \frac{\mathbf{y} + \mu_1 \mathbf{v}}{1 + \mu_1}, \quad (15)$$

$$\text{prox}_{g_2/\mu_2}(\mathbf{z}) = \text{soft}(\mathbf{z}, \lambda/\mu_2), \quad (16)$$

where “soft” denotes the soft-threshold function in (8). Line 4 of Algorithm 2 (the other key component) has the form

$$\mathbf{z}_{k+1} \leftarrow \left(\mathbf{W}^* \mathbf{A}^* \mathbf{A} \mathbf{W} + \frac{\mu_2}{\mu_1} \mathbf{I} \right)^{-1} \left(\mathbf{W}^* \mathbf{A}^* \boldsymbol{\zeta}^{(1)} + \frac{\mu_2}{\mu_1} \boldsymbol{\zeta}^{(2)} \right). \quad (17)$$

As shown in [2], if \mathbf{W} corresponds to a Parseval frame [28], i.e., if³ $\mathbf{W}\mathbf{W}^* = \mathbf{I}$ (although possibly $\mathbf{W}^*\mathbf{W} \neq \mathbf{I}$), the

²We use the generic term “wavelet” to mean any wavelet-like multiscale representation, such as “curvelets,” “beamlets,” or “ridgelets” [28].

³In fact, the frame only needs to be tight [28], i.e., $\mathbf{W}\mathbf{W}^* = \omega \mathbf{I}$; without loss of generality, we assume $\omega = 1$, which yields lighter notation. Recently, a method for relaxing the tight frame condition was proposed in [33].

Sherman-Morrison-Woodbury inversion formula can be used to write the matrix inverse in (17) as

$$\frac{\mu_1}{\mu_2} \left(\mathbf{I} - \mathbf{W}^* \underbrace{\mathbf{A}^* (\mathbf{A} \mathbf{A}^* + (\mu_2/\mu_1) \mathbf{I})^{-1} \mathbf{A}}_{\mathbf{F}} \mathbf{W} \right). \quad (18)$$

Assuming periodic BC, matrix \mathbf{A} is circulant, thus factors into

$$\mathbf{A} = \mathbf{U}^* \mathbf{\Lambda} \mathbf{U}, \quad (19)$$

where \mathbf{U} and \mathbf{U}^* are the unitary matrices representing the DFT and its inverse, and $\mathbf{\Lambda}$ is the diagonal matrix of the DFT coefficients of the convolution kernel. Using (19), matrix \mathbf{F} defined in (18) can be written as

$$\mathbf{F} = \mathbf{U}^* \mathbf{\Lambda}^* (|\mathbf{\Lambda}|^2 + (\mu_2/\mu_1) \mathbf{I})^{-1} \mathbf{\Lambda} \mathbf{U}, \quad (20)$$

where $|\mathbf{\Lambda}|^2$ denotes the squared absolute values of the entries of $\mathbf{\Lambda}$. Since the product $\mathbf{\Lambda}^* (|\mathbf{\Lambda}|^2 + (\mu_2/\mu_1) \mathbf{I})^{-1} \mathbf{\Lambda}$ involves only diagonal matrices, it can be computed with only $O(n)$ cost, while the products by \mathbf{U} and \mathbf{U}^* (DFT and its inverse) can be computed with $O(n \log n)$ cost, using the FFT; thus, computing \mathbf{F} and multiplying by it has $O(n \log n)$ cost.

The leading cost of each application of (17) (line 4 of Algorithm 2) is either the $O(n \log n)$ cost associated with \mathbf{F} or the cost of the products by \mathbf{W}^* and \mathbf{W} . For most tight frames used in image restoration (e.g. undecimated wavelet transforms, curvelets, complex wavelets), these products correspond to direct and inverse transforms, for which fast $O(n \log n)$ algorithms exist [28]. In conclusion, under periodic BC and for a large class of frames, each iteration of Algorithm 2 for solving (9) has $O(n \log n)$ cost. Finally, convergence of this instance of Algorithm 2 is established in the following proposition.

Proposition 2. *The instance of Algorithm 2, with the definitions in (11)–(14), with line 4 computed as in (17), and the proximity operators in line 6 as given in (15) and (16), converges to a solution of (9).*

Proof: Since g_1 and g_2 are proper, closed, convex functions, so is the objective function in (9). Because g_2 is coercive, so is the objective function in (9), thus its set of minimizers is not empty [15]. Matrix $\mathbf{H}^{(2)} = \mathbf{I}$ obviously has full column rank, which implies that \mathbf{G} also has full column rank; that fact combined with the fact that g_1 and g_2 are proper, closed, convex functions, and that a minimizer exists, allows invoking Proposition 1 which concludes the proof. ■

B. Frame-Based Analysis Formulation

In the frame-based analysis approach, the problem is formulated directly with respect to the unknown image (rather than its synthesis coefficients),

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{A} \mathbf{x}\|_2^2 + \lambda \|\mathbf{P} \mathbf{x}\|_1, \quad (21)$$

where \mathbf{A} is as in (9) and $\mathbf{P} \in \mathbb{R}^{m \times n}$ ($m \geq n$) is the analysis operator of a Parseval frame, i.e., satisfying⁴ $\mathbf{P}^* \mathbf{P} = \mathbf{I}$ (although possibly $\mathbf{P} \mathbf{P}^* \neq \mathbf{I}$, unless the frame is an orthonormal

⁴As above, the frame is only required to be tight, i.e., $\mathbf{P}^* \mathbf{P} = \omega \mathbf{I}$; since $\omega = 1$ can be obtained simply by normalization, there is no loss of generality.

basis) [28]. Problem (21) has the form (2), with $J = 2$, g_1 and g_2 as given in (11) and (12), respectively, and

$$\mathbf{H}^{(1)} \in \mathbb{R}^{n \times n}, \quad \mathbf{H}^{(1)} = \mathbf{A} \quad (22)$$

$$\mathbf{H}^{(2)} \in \mathbb{R}^{m \times n}, \quad \mathbf{H}^{(2)} = \mathbf{P}. \quad (23)$$

Since the proximity operators of g_1 and g_2 are the same as in the synthesis case (see (15) and (16)), only line 4 of Algorithm 2 has a different form:

$$\mathbf{z}_{k+1} \leftarrow \left(\mathbf{A}^* \mathbf{A} + \frac{\mu_2}{\mu_1} \mathbf{P}^* \mathbf{P} \right)^{-1} \left(\mathbf{A}^* \boldsymbol{\zeta}^{(1)} + \frac{\mu_2}{\mu_1} \mathbf{P}^* \boldsymbol{\zeta}^{(2)} \right). \quad (24)$$

Since $\mathbf{P}^* \mathbf{P} = \mathbf{I}$ and $\mathbf{A} = \mathbf{U}^* \mathbf{\Lambda} \mathbf{U}$, the matrix inverse is simply

$$\left(\mathbf{A}^* \mathbf{A} + \frac{\mu_2}{\mu_1} \mathbf{I} \right)^{-1} = \mathbf{U}^* (|\mathbf{\Lambda}|^2 + (\mu_2/\mu_1) \mathbf{I})^{-1} \mathbf{U}, \quad (25)$$

which has $O(n \log n)$ cost, since $(|\mathbf{\Lambda}|^2 + (\mu_2/\mu_1) \mathbf{I})$ is a diagonal matrix and the products by \mathbf{U} and \mathbf{U}^* (the DFT and its inverse) can be computed using the FFT.

In conclusion, under periodic BC and for a large class of frames, each iteration of Algorithm 2 for solving (21) has $O(n \log n)$ cost. Finally, convergence of this instance of Algorithm 2 is claimed in the following proposition.

Proposition 3. *The instance of Algorithm 2, with the definitions in (11), (12), (22), and (23), with line 4 computed as in (24)–(25), and the proximity operators in line 6 as given in (15) and (16), converges to a solution of (21).*

Proof: Since g_1 and g_2 are proper, closed, convex functions, so is the objective function in (9). Because g_2 is coercive and \mathbf{P} is the analysis operator of a tight frame, $\ker(\mathbf{P}) = \{\mathbf{0}\}$ (where $\ker(\mathbf{P})$ is the null space of matrix \mathbf{P}), the objective function in (9) is coercive, thus its set of minimizers is not empty [15]. Matrix $\mathbf{H}^{(2)} = \mathbf{P}$ has full column rank, thus the same is true of \mathbf{G} ; combining that with g_1 and g_2 being proper, closed, and convex, and given the existence of a minimizer, allows invoking Proposition 1 to conclude the proof. ■

C. Total Variation

The classical formulation of TV-based image deconvolution consists in a convex optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{A} \mathbf{x}\|_2^2 + \lambda \sum_{i=1}^n \|\mathbf{D}_i \mathbf{x}\|_2, \quad (26)$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^n$, $\lambda > 0$, and $\mathbf{A} \in \mathbb{R}^{n \times n}$ have the same meanings as above, and each $\mathbf{D}_i \in \mathbb{R}^{2 \times n}$ is the matrix that computes the horizontal and vertical differences at pixel i (also with periodic BC) [39], [41]. The sum $\sum_{i=1}^n \|\mathbf{D}_i \mathbf{x}\|_2 = \text{TV}(\mathbf{x})$ defines the so-called *total-variation*⁵ (TV) of image \mathbf{x} .

Problem (26) has the form (2), with $J = n + 1$, and

$$g_1 : \mathbb{R}^n \rightarrow \mathbb{R}, \quad g_1(\mathbf{v}) = \frac{1}{2} \|\mathbf{y} - \mathbf{v}\|_2^2, \quad (27)$$

$$g_i : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad g_i(\mathbf{v}) = \lambda \|\mathbf{v}\|_2, \quad \text{for } i = 2, \dots, n + 1, \quad (28)$$

$$\mathbf{H}^{(1)} \in \mathbb{R}^{n \times n}, \quad \mathbf{H}^{(1)} = \mathbf{A}, \quad (29)$$

$$\mathbf{H}^{(i)} \in \mathbb{R}^{2 \times n}, \quad \mathbf{H}^{(i)} = \mathbf{D}_{i-1}, \quad \text{for } i = 2, \dots, n + 1, \quad (30)$$

⁵In fact, this is the so-called isotropic discrete approximation of the continuous total-variation [11], [36].

For matrix Υ (see (5)), we adopt $\mu_2 = \dots = \mu_{n+1} > 0$. The main steps of the resulting instance of Algorithm 2 for solving (26) are as follows.

- As shown in [39], [41], the matrix to be inverted in line 4 of Algorithm 2 can be written as

$$\mu_1 \mathbf{A}^* \mathbf{A} + \mu_2 \sum_{j=1}^n \mathbf{D}_j^* \mathbf{D}_j = \mu_1 \mathbf{A}^* \mathbf{A} + \mu_2 ((\mathbf{D}^h)^* \mathbf{D}^h + (\mathbf{D}^v)^* \mathbf{D}^v), \quad (31)$$

where $\mathbf{D}^h, \mathbf{D}^v \in \mathbb{R}^{n \times n}$ are the matrices collecting the first and second rows, respectively, of each of the n matrices \mathbf{D}_i ; that is, \mathbf{D}^h computes all the horizontal differences and \mathbf{D}^v all the vertical differences. With periodic BC, the matrices in (31) can be factorized as

$$\mathbf{A} = \mathbf{U}^* \mathbf{\Lambda} \mathbf{U}, \quad \mathbf{D}^h = \mathbf{U}^* \mathbf{\Delta}^h \mathbf{U}, \quad \mathbf{D}^v = \mathbf{U}^* \mathbf{\Delta}^v \mathbf{U}, \quad (32)$$

where $\mathbf{\Lambda}, \mathbf{\Delta}^h$, and $\mathbf{\Delta}^v$ are diagonal matrices. Thus, the inverse of the matrix in (31) can be written as

$$\mathbf{U}^* (\mu_1 |\mathbf{\Lambda}|^2 + \mu_2 |\mathbf{\Delta}^h|^2 + \mu_2 |\mathbf{\Delta}^v|^2)^{-1} \mathbf{U} \equiv \mathbf{K}, \quad (33)$$

which involves a diagonal inversion, with $O(n)$ cost, and the products by \mathbf{U} and \mathbf{U}^* (the direct and inverse DFT), which have $O(n \log n)$ cost (by FFT).

- The sum yielding the vector to which this inverse is applied (line 4 of Algorithm 2) is

$$\begin{aligned} \sum_{j=1}^{n+1} \mu_j (\mathbf{H}^{(j)})^* \boldsymbol{\zeta}^{(j)} &= \mu_1 \mathbf{A}^* \boldsymbol{\zeta}^{(1)} + \mu_2 \sum_{j=1}^n \mathbf{D}_j^* \boldsymbol{\zeta}^{(j+1)} \\ &= \mu_1 \mathbf{A}^* \boldsymbol{\zeta}^{(1)} + \mu_2 (\mathbf{D}^h)^* \boldsymbol{\delta}^h + \mu_2 (\mathbf{D}^v)^* \boldsymbol{\delta}^v, \end{aligned}$$

where $\boldsymbol{\delta}^h, \boldsymbol{\delta}^v \in \mathbb{R}^n$ contain the first and second component, respectively, of all the $\boldsymbol{\zeta}^{(j)}$ vectors, for $j = 1, \dots, n$. As seen above, the product by \mathbf{A}^* has $O(n \log n)$ cost via the FFT, while the products by $(\mathbf{D}^h)^*$ and $(\mathbf{D}^v)^*$ (which correspond to local differences) have $O(n)$ cost.

- The proximity operator of g_1 is as given in (15).
- The proximity operator of $g_i(\mathbf{v}) = \lambda \|\mathbf{v}\|_2$, for $i = 2, \dots, n+1$, is the so-called vector-soft function [39], [41],

$$\text{prox}_{\frac{g_i}{\mu_i}}(\mathbf{v}) = \text{v-soft}\left(\mathbf{v}, \frac{\lambda}{\mu_i}\right) = \frac{\mathbf{v}}{\|\mathbf{v}\|_2} \max\left\{\|\mathbf{v}\|_2 - \frac{\lambda}{\mu_i}, 0\right\},$$

with the convention that $0/\|0\|_2 = 0$.

Plugging all these equalities into Algorithm 2 yields Algorithm 3, which is similar to the one presented in [39].

Finally, convergence of Algorithm 3 is addressed in the following proposition.

Proposition 4. *Consider problem (26) and assume that $\mathbf{1} \notin \ker(\mathbf{A})$. Then Algorithm 3 converges to a solution of (26).*

Proof: Since g_1, \dots, g_{n+1} (27)–(28) are proper, closed, convex functions, so is the objective function in (26). That $\mathbf{1} \notin \ker(\mathbf{A})$ implies coercivity of the objective function in (26), thus existence of minimizer(s), was shown in [12]. Matrix $\mathbf{G} = [\mathbf{A}^*, \mathbf{D}_1^*, \dots, \mathbf{D}_n^*]^*$ can be written as a permutation of the rows of $[\mathbf{A}^*, \mathbf{D}^*]^*$, where $\mathbf{D} = [(\mathbf{D}^h)^*, \dots, (\mathbf{D}^v)^*]^*$; since $\mathbf{D}\mathbf{x} = \mathbf{0} \Leftrightarrow \mathbf{x} = \alpha \mathbf{1}$ ($\alpha \in \mathbb{R}$), if $\mathbf{1} \notin \ker(\mathbf{A})$, then \mathbf{G} has

Algorithm 3: ADMM-TV

```

1 Set  $k = 0$ , choose  $\mu_1, \mu_2 > 0$ ,  $\mathbf{u}_0^{(j)}, \mathbf{d}_0^{(j)}$ ,  $j = 1, \dots, n+1$ 
2 repeat
3    $\boldsymbol{\zeta} \leftarrow \mathbf{u}_k + \mathbf{d}_k$ 
4    $\boldsymbol{\delta}^h = [(\boldsymbol{\zeta}^{(2)})_1, \dots, (\boldsymbol{\zeta}^{(n+1)})_1]^*$ 
5    $\boldsymbol{\delta}^v = [(\boldsymbol{\zeta}^{(2)})_2, \dots, (\boldsymbol{\zeta}^{(n+1)})_2]^*$ 
6    $\mathbf{z}_{k+1} \leftarrow \mathbf{K}(\mu_1 \mathbf{A}^* \boldsymbol{\zeta}^{(1)} + \mu_2 (\mathbf{D}^h)^* \boldsymbol{\delta}^h + \mu_2 (\mathbf{D}^v)^* \boldsymbol{\delta}^v)$ 
7    $\mathbf{u}_{k+1}^{(1)} \leftarrow \frac{1}{1+\mu_1}(\mathbf{y} + \mu_1(\mathbf{A} \mathbf{z}_{k+1} - \mathbf{d}_k^{(1)}))$ 
8    $\mathbf{d}_{k+1}^{(1)} \leftarrow \mathbf{d}_k^{(1)} - (\mathbf{A} \mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(1)})$ 
9   for  $j = 2$  to  $n+1$  do
10     $\mathbf{u}_{k+1}^{(j)} \leftarrow \text{vect-soft}(\mathbf{D}_{j-1} \mathbf{z}_{k+1} - \mathbf{d}_k^{(j)}, \frac{\lambda}{\mu_2})$ 
11     $\mathbf{d}_{k+1}^{(j)} \leftarrow \mathbf{d}_k^{(j)} - (\mathbf{D}_{j-1} \mathbf{z}_{k+1} - \mathbf{u}_{k+1}^{(j)})$ 
12  end
13   $k \leftarrow k + 1$ 
14 until stopping criterion is satisfied
```

full column rank. Finally, since Algorithm 3 is an instance of Algorithm 2 (with all updates computed exactly), Proposition 1 guarantees its convergence to a solution of (26). ■

Notice that the condition $\mathbf{1} \notin \ker(\mathbf{A})$ is very mild, and clearly satisfied by the convolution with any low-pass-type filter, such as those modeling uniform, motion, Gaussian, circular, and other types of blurs. In fact, for these blurring filters, $\mathbf{A}\mathbf{1} = \beta \mathbf{1}$, where β is the DC gain (typically $\beta = 1$).

IV. DECONVOLUTION WITH UNKNOWN BOUNDARIES

A. The Observation Model

To handle images with unknown boundaries, we model the boundary pixels as unobserved, which is achieved by modifying (10) into

$$\mathbf{y} = \mathbf{M} \mathbf{A} \mathbf{x} + \mathbf{n}, \quad (34)$$

where $\mathbf{M} \in \{0, 1\}^{m \times n}$ (with $m < n$) is a masking matrix, *i.e.*, a matrix whose rows are a subset of the rows of an identity matrix. The role of \mathbf{M} is to observe only the subset of the image domain in which the elements of $\mathbf{A}\mathbf{x}$ do not depend on the boundary pixels; consequently, the BC assumed for the convolution represented by \mathbf{A} is irrelevant, and we may adopt periodic BCs, for computational convenience.

Assuming that \mathbf{A} models the convolution with a blurring filter with a limited support of size $(1+2l) \times (1+2l)$, and that \mathbf{x} and $\mathbf{A}\mathbf{x}$ represent square images of dimensions $\sqrt{n} \times \sqrt{n}$, then matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, with $m = (\sqrt{n} - 2l)^2$, represents the removal of a band of width l of the outermost pixels of the full convolved image $\mathbf{A}\mathbf{x}$.

Problem (34) can be seen as hybrid of deconvolution and inpainting [13], where the missing pixels constitute the unknown boundary. If $\mathbf{M} = \mathbf{I}$, model (34) reduces to a standard periodic deconvolution problem. Conversely, if $\mathbf{A} = \mathbf{I}$, (34) becomes a pure inpainting problem. Moreover, the formulation (34) can

be used to model problems where not only the boundary, but also other pixels, are missing, as in standard image inpainting.

The following subsections describe how to handle observation model (34), in the context of the ADMM-based deconvolution algorithms reviewed in the previous section.

B. Frame-Based Synthesis Formulation

1) *Mask Decoupling (MD)*: Under observation model (34), the frame-based synthesis formulation (9) changes to

$$\hat{\mathbf{z}} = \arg \min_{\mathbf{z} \in \mathbb{R}^d} \frac{1}{2} \|\mathbf{y} - \mathbf{MAWz}\|_2^2 + \lambda \|\mathbf{z}\|_1. \quad (35)$$

At this point, one could be tempted to map (35) into (2) using the same correspondences as in (11), (12), and (14), and simply changing (13) into

$$\mathbf{H}^{(1)} \in \mathbb{R}^{m \times d}, \quad \mathbf{H}^{(1)} = \mathbf{MAW}. \quad (36)$$

The problem with this choice is that the matrix to be inverted in line 4 of Algorithm 2 would become

$$(\mathbf{W}^* \mathbf{A}^* \mathbf{M}^* \mathbf{MAW} + (\mu_2/\mu_1) \mathbf{I}), \quad (37)$$

instead of the one in (17). The “trick” used in Subsection III-A to express this inversion in the DFT domain can no longer be used due to presence of \mathbf{M} , invalidating the corresponding FFT-based implementation of line 4 of Algorithm 2. It is clear that the source of the difficulty is the product \mathbf{MA} , which is the composition of a mask (a spatial point-wise operation) with a circulant matrix (a point-wise operation in the DFT domain); to sidestep this difficulty, we need to decouple these two operations, which is achieved by defining (instead of (11))

$$g_1 : \mathbb{R}^n \rightarrow \mathbb{R}, \quad g_1(\mathbf{v}) = \frac{1}{2} \|\mathbf{y} - \mathbf{Mv}\|_2^2, \quad (38)$$

while keeping g_2 , $\mathbf{H}^{(1)}$, and $\mathbf{H}^{(2)}$ as in (12)–(14). With this choice, line 4 of Algorithm 2 is still given by (17) (with its efficient FFT-based implementation); the only change is the proximity operator of the new g_1 ,

$$\begin{aligned} \text{prox}_{g_1/\mu_1}(\mathbf{v}) &= \arg \min_{\mathbf{u}} \|\mathbf{Mu} - \mathbf{y}\|_2^2 + \mu_1 \|\mathbf{u} - \mathbf{v}\|_2^2 \\ &= (\mathbf{M}^* \mathbf{M} + \mu_1 \mathbf{I})^{-1} (\mathbf{M}^* \mathbf{y} + \mu_1 \mathbf{v}); \end{aligned} \quad (39)$$

notice that, due to the special structure of \mathbf{M} , matrix $\mathbf{M}^* \mathbf{M}$ is diagonal, thus the inversion in (40) has $O(n)$ cost, the same being true about the product $\mathbf{M}^* \mathbf{y}$, which corresponds to extending the observed image \mathbf{y} to the size of \mathbf{x} , by creating a boundary of zeros around it. Of course, both $(\mathbf{M}^* \mathbf{M} + \mu_1 \mathbf{I})^{-1}$ and $\mathbf{M}^* \mathbf{y}$ can be pre-computed and then used throughout the algorithm, as long as μ_1 is kept constant. We refer to this approach as *mask decoupling* (MD).

In conclusion, the proposed MD-based ADMM algorithm for image deconvolution with unknown boundaries, under frame-based synthesis regularization, is Algorithm 2 with line 4 implemented as in (17) and the proximity operators in line 6 given by (40) and (16). We refer to this algorithm as FS-MD (*frame synthesis mask decoupling*). As with the periodic BC, the leading cost is $O(n \log n)$ per iteration. Finally, convergence of the FS-MD algorithm is guaranteed

by the following proposition (the proof of which is similar to that of Proposition 2).

Proposition 5. *Algorithm FS-MD (i.e., the instance of Algorithm 2 with the definitions in (38) and (12)–(14), with line 4 computed as in (17), and the proximity operators in line 6 as given in (40) and (16)) converges to a solution of (35).*

2) *Using the Reeves-Šorel Technique*: An alternative to the approach just presented of decoupling the convolution from the masking operator is to use the method of [35], [40] to tackle the inversion (37). Following [35], notice that

$$\mathbf{AW} = \mathbf{S} \begin{bmatrix} \mathbf{MAW} \\ \mathbf{B} \end{bmatrix}, \quad (41)$$

where \mathbf{B} contains the rows of \mathbf{AW} that are missing from \mathbf{MAW} (recall that the rows of \mathbf{M} are a subset of those of an identity matrix) and \mathbf{S} is a permutation matrix that puts these missing rows in their original positions in \mathbf{AW} . Noticing that

$$\begin{aligned} \mathbf{W}^* \mathbf{A}^* \mathbf{AW} &= [\mathbf{W}^* \mathbf{A}^* \mathbf{M}^*, \mathbf{B}^*] \mathbf{S}^* \mathbf{S} \begin{bmatrix} \mathbf{MAW} \\ \mathbf{B} \end{bmatrix} \\ &= \mathbf{W}^* \mathbf{A}^* \mathbf{M}^* \mathbf{MAW} + \mathbf{B}^* \mathbf{B} \end{aligned} \quad (42)$$

(\mathbf{S} is a permutation matrix, thus $\mathbf{S}^* \mathbf{S} = \mathbf{I}$), the inverse of (37) can be written (with $\gamma = \mu_2/\mu_1$) as

$$\begin{aligned} (\mathbf{W}^* \mathbf{A}^* \mathbf{M}^* \mathbf{MAW} + \gamma \mathbf{I})^{-1} &= (\mathbf{W}^* \mathbf{A}^* \mathbf{AW} - \mathbf{B}^* \mathbf{B} + \gamma \mathbf{I})^{-1} \\ &= \mathbf{C} - \mathbf{CB}^* (\mathbf{BCB}^* - \mathbf{I})^{-1} \mathbf{BC}, \end{aligned} \quad (43)$$

where the second equality results from using the Sherman-Morrison-Woodbury matrix inversion identity, after defining $\mathbf{C} \equiv (\gamma \mathbf{I} + \mathbf{W}^* \mathbf{A}^* \mathbf{AW})^{-1}$. Since \mathbf{A} is circulant, \mathbf{C} can be efficiently computed via FFT, as explained in (17)–(20). The inversion $(\mathbf{BCB}^* - \mathbf{I})^{-1}$ in (43) cannot be computed via FFT; however, its dimension corresponds to the number of unknown boundary pixels (number of rows in \mathbf{B}), usually much smaller than image itself. As in [35], [40], we use the CG algorithm to solve the corresponding system; we confirmed experimentally that (as in [40]) taking only one CG iteration (initialized with the estimate of the previous outer iteration) yields the fastest convergence, without degrading the final result. Thus, in our experiments, we use a single CG iteration per outer iteration.

Approximately solving line 4 of Algorithm 2 via one (or even more) iterations of the CG algorithm, rather than an FFT-based exact solution, makes convergence more difficult to analyze, so we will not present a formal proof. In a related problem [23], it was shown experimentally that if the iterative solvers used to implement the minimizations defining the ADMM steps are warm-started (i.e., initialized with the values from the previous outer iteration), then the error sequences η_k and ρ_k , for $k = 0, 1, 2, \dots$ are absolutely summable as required by Theorem 1. Finally, we refer to this algorithm as FS-CG (*frame synthesis conjugate gradient*).

C. Frame-Based Analysis Formulation

1) *Mask Decoupling (MD)*: The frame-based analysis formulation corresponding to observation model (9) is

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{MAx}\|_2^2 + \lambda \|\mathbf{Px}\|_1. \quad (44)$$

Following the MD approach introduced for the synthesis formulation, we map Problem (44) into the form (2), by using g_1 as defined in (38), and keeping $\mathbf{H}^{(1)}$, $\mathbf{H}^{(2)}$, and g_2 as in the periodic BC case: (22), (23), and (12), respectively.

The only difference in the resulting instance of Algorithm 2 is the use of the proximity operator of the new g_1 , as given in (40). In conclusion, the proposed ADMM-based algorithm for image deconvolution with unknown boundaries, under frame-based analysis regularization, is simply Algorithm 2, with line 4 implemented as in (24)–(25), and the proximity operators in line 6 given by (40) and (16). We refer to this algorithm as FA-MD (*frame analysis mask decoupling*). The computational cost of the algorithm is $O(n \log n)$ per iteration, as in the periodic BC case. Convergence of FA-MD is addressed by the following proposition (the proof of which is similar to that of Proposition 3).

Proposition 6. *Algorithm FA-MD (i.e., Algorithm 2 with the definitions in (38), (22), (23), and (12), with line 4 computed as in (24)–(25), and the proximity operators in line 6 as given in (40) and (16)) converges to a solution of (44).*

2) *Using the Reeves-Šorel Technique:* Consider Problem (44) and map into the form (2) using (11)–(12), (23), and

$$\mathbf{H}^{(1)} \in \mathbb{R}^{n \times n}, \quad \mathbf{H}^{(1)} = \mathbf{M}\mathbf{A}. \quad (45)$$

The matrix inverse computed in line 4 of Algorithm 2 is now (with $\gamma = \mu_2/\mu_1$)

$$\left(\mathbf{A}^* \mathbf{M}^* \mathbf{M} \mathbf{A} + \gamma \mathbf{P}^* \mathbf{P}\right)^{-1} = \left(\mathbf{A}^* \mathbf{M}^* \mathbf{M} \mathbf{A} + \gamma \mathbf{I}\right)^{-1}, \quad (46)$$

which can no longer be computed as in (25), since matrix $\mathbf{M}\mathbf{A}$ is not circulant. Using the same steps as in (41)–(43), with \mathbf{A} replacing $\mathbf{A}\mathbf{W}$ and $\mathbf{C} \equiv (\gamma \mathbf{I} + \mathbf{A}^* \mathbf{A})^{-1}$, leads to

$$(\mathbf{A}^* \mathbf{M}^* \mathbf{M} \mathbf{A} + \gamma \mathbf{I})^{-1} = \mathbf{C} - \mathbf{C} \mathbf{B}^* (\mathbf{B} \mathbf{C} \mathbf{B}^* - \mathbf{I})^{-1} \mathbf{B} \mathbf{C}, \quad (47)$$

Since \mathbf{A} is circulant, $\mathbf{C} = (\gamma \mathbf{I} + \mathbf{A}^* \mathbf{A})^{-1}$ can be efficiently computed via FFT, as in (25). As in the synthesis case, the inverse $(\mathbf{B} \mathbf{C} \mathbf{B}^* - \mathbf{I})^{-1}$ in (47) is computed approximately by taking one (warm-started) CG iteration (for the reason explained in Subsection IV-B2). We refer to the resulting algorithm as FA-CG (*frame analysis conjugate gradient*).

D. TV-Based Deconvolution with Unknown Boundaries

1) *Mask Decoupling (MD):* Given the observation model in (34), TV-based deconvolution with unknown boundaries is formulated as

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{y} - \mathbf{M} \mathbf{A} \mathbf{x}\|_2^2 + \lambda \sum_{i=1}^n \|\mathbf{D}_i \mathbf{x}\|_2, \quad (48)$$

with every element having the exact same meanings as above.

Following the MD approach, we map (48) into the form (2) using g_1 as defined in (38) and keeping all the other correspondences (28)–(30) unchanged. The only resulting change to Algorithm 3 is in line 7, which becomes

$$\mathbf{u}_{k+1}^{(1)} \leftarrow (\mathbf{M}^* \mathbf{M} + \mu_1 \mathbf{I})^{-1} \left(\mathbf{M}^* \mathbf{y} + \mu_1 (\mathbf{A} \mathbf{z}_{k+1} - \mathbf{d}_k^{(1)}) \right), \quad (49)$$

as a consequence of the application of the proximity operator of g_1 , as given in (40). In summary, the ADMM-based algorithm for TV-based deconvolution with unknown boundaries has the exact same structure as Algorithm 3, with line 7 replaced by 49. We refer to this algorithm as TV-MD (*TV mask decoupling*). Finally, convergence of TV-MD is addressed in the following proposition (the proof of which is similar to that of Proposition 4).

Proposition 7. *If $\mathbf{1} \notin \ker(\mathbf{M}\mathbf{A})$, then TV-MD (i.e., the version of Algorithm 3 with line 7 replaced by (49)) converges to a solution of (48).*

As for periodic BCs, the condition $\mathbf{1} \notin \ker(\mathbf{M}\mathbf{A})$ is very mild, and is satisfied by the convolution with any low-pass filter (for which $\mathbf{A}\mathbf{1} = \beta\mathbf{1}$, where β is the DC gain, typically $\beta \simeq 1$) and $\mathbf{M}\mathbf{1} \neq 0$, if $m > 1$ (at least one pixel is observed).

2) *Using the Reeves-Šorel Technique:* To use the Reeves-Šorel technique to handle (48), the mapping to the form (2) is done as in (27), (28), and (30), with (29) replaced by (45). The consequence is a simple modification of Algorithm 3, where \mathbf{A} is replaced by $\mathbf{M}\mathbf{A}$, in line 8, while in line 6, \mathbf{A}^* is replaced by $\mathbf{A}^* \mathbf{M}^*$ and \mathbf{K} is redefined (instead of (33)) as the inverse of the matrix in (31), i.e. (with $\gamma = \mu_2/\mu_1$),

$$\mathbf{K} = \frac{1}{\mu_1} \left(\mathbf{A}^* \mathbf{M}^* \mathbf{M} \mathbf{A} + \gamma (\mathbf{D}^h)^* \mathbf{D}^h + \gamma (\mathbf{D}^v)^* \mathbf{D}^v \right)^{-1}. \quad (50)$$

A derivation similar to (41)–(43) allows writing

$$\mathbf{K} = \frac{1}{\mu_1} \left(\mathbf{C} - \mathbf{C} \mathbf{B}^* (\mathbf{B} \mathbf{C} \mathbf{B}^* - \mathbf{I})^{-1} \mathbf{B} \mathbf{C} \right), \quad (51)$$

where

$$\mathbf{C} = (\mathbf{A}^* \mathbf{A} + \gamma (\mathbf{D}^h)^* \mathbf{D}^h + \gamma (\mathbf{D}^v)^* \mathbf{D}^v)^{-1}. \quad (52)$$

The circulant nature of \mathbf{A} , \mathbf{D}^h , and \mathbf{D}^v allows computing \mathbf{C} efficiently in the DFT domain (similarly to (33)). Finally, as above, the inverse $(\mathbf{B} \mathbf{C} \mathbf{B}^* - \mathbf{I})^{-1}$ in (51) is approximated by taking, as above, one CG iterations; the resulting algorithm is referred to as TV-CG (*TV conjugate gradient*).

V. EXPERIMENTS

In the experiments herein reported, we use the benchmark images *Lena* and *cameraman* (of size 256×256), with 4 different blurs (out-of-focus, linear motion, uniform, and Gaussian), all of size 19×19 (i.e., $(2l+1) \times (2l+1)$, with $l = 9$, as explained in Section IV-A), at four different BSNRs (*blurred signal to noise ratio*): 30dB, 40dB, 50dB, and 60dB. The reason why we concentrate on large blurs is that the effect of the boundary conditions is very evident in this case.

To set up a scenario of unknown boundaries, the observed images (of size 238×238 , since $238 = 256 - 18$) are obtained by applying the filters to the 238×238 central region of the original images, using the band of pixels (of width 9) around this region as the boundary pixels. In the proposed formulation, these boundary pixels are modeled as unknown. Notice that this procedure is equivalent to convolving the full (256×256) images using an arbitrary BC (periodic, for computational convenience) and then keeping only the valid

region (*i.e.*, the one that does not depend on the BC). Since $n = 256^2 = 65536$ and $m = 238^2 = 56644$, the number of unknown boundary pixels is 8892.

Preliminary tests showed that both TV and frame-based analysis regularization slightly (but consistently) outperform the frame-based synthesis formulation, in terms of ISNR (*improvement in signal to noise ratio*), thus we will not report experiments with the latter. Concerning TV regularization, we consider four algorithms: (i) TV-MD (proposed in Subsection IV-D1); (ii) TV-CG (proposed in Subsection IV-D2); (iii) Algorithm 3 with the (wrong) assumption of periodic BC (referred to as TV-BC); and (iv) Algorithm 3 also with the (wrong) assumption of periodic BC, after pre-processing the observed image with the “edgetaper” (ET) function (referred to as TV-ET). Similarly, in the frame-based⁶ analysis formulation, we consider four algorithms: (i) FA-MD (proposed in Subsection IV-C1); (ii) FA-CG (proposed in Subsection IV-C2); (iii) the algorithm defined in Subsection III-B, which wrongly assumes periodic BC (referred to as FA-BC); and (iv) FA-BC, after pre-processing the observed image with the “edgetaper” (ET) function (referred to as FA-ET). Notice that the methods that assume periodic BC recover images of the same size as the observed image ($\sqrt{m} \times \sqrt{m}$), while those based on model (34) recover full images ($\sqrt{n} \times \sqrt{n}$), extended with the estimated boundary pixels.

The regularization parameter λ was manually adjusted to yield the best ISNR for each experimental condition. For the ADMM parameters (μ_1 and μ_2), which affect the convergence speed, we used heuristic rules, which lead to a good performance of the algorithm: $\mu_2 = 10\lambda$ and $\mu_1 = \min\{1, 5000\mu_2\}$.

Fig. 2 shows the results obtained with the *Lena* image, with uniform blur at 60dB BSNR, using frame-based analysis regularization. Notice how the wrong assumption of periodic BC causes a complete failure of the deblurring algorithm, while treating the boundary as unknown allows the algorithm to obtain a remarkably good estimate of the unobserved boundary pixels, as well as a deconvolved image without any boundary artifacts. Fig. 3 illustrates a similar experiment, now with the *cameraman* image and linear motion blur at 40dB BSNR; the conclusions are similar to those drawn from Fig. 2, with exception that, in this case, the edgetaper-based approach is able to yield a reasonable image estimate, although far from those produced by TV-CG and TV-MD.

Tables I and II show the ISNR values achieved by the four algorithms mentioned above, for TV and frame-based analysis regularization, respectively. Naturally, the methods that treat the boundary pixels as unknown (the CG and MD versions) yield very similar ISNR values, since they simply corresponds to two variants of an algorithm minimizing the same objective function. It is also clear that by wrongly assuming a periodic BC (algorithms TV-BC and FA-BC) leads to very poor results, as already illustrated in Figs. 2 and 3, and that the edgetaper method is able to alleviate this problem somewhat. Finally, we observe that frame-based analysis regularization yields marginally better results than TV regularization.

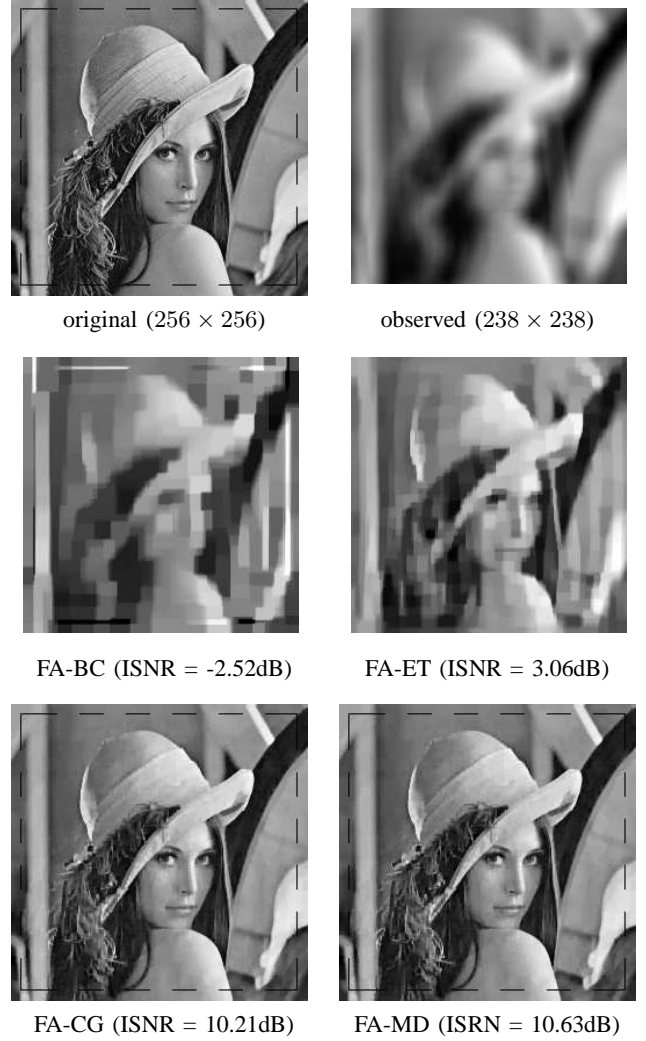


Fig. 2. Results obtained on the *Lena* image, degraded by a uniform 19×19 blur, at 60dB BSNR, by the four algorithms considered (see text for the acronyms). Notice that FA-BC and FA-ET produce 238×238 images, while FA-CG and FA-MD yield 256×256 images; the dashed square shows the limit of the boundary region.

Table III shows the average times⁷ taken by the four algorithms tested at each of the BSNR values considered. To obtain these results, the MD variant of each algorithm is first run using its own convergence criterion (relative variation of the objective function falling below 10^{-4}), and then the CG variant is run until it reaches the same value of the objective function. These results show that the MD versions are faster than their CG-based counterparts by factors that range from (roughly) from 1.5 to 2. This speed-up is confirmed by the average time per iteration of each of the four algorithms, shown in Table IV; each iteration of the MD variant is roughly 1.6 to 1.8 faster than the CG counterpart. Notice that, in addition to converging faster, the MD approach is also considerably simpler to implement, since it does not involve the inner CG iterations. Since, as shown above, both methods yield similar ISNR values, this allows concluding that, in

⁶In the frame-based approach, we use a redundant Haar frame, with four levels, although we could use any other frame with fast transforms.

⁷The methods were implemented in MATLAB and the experiments run on a Intel Core i5 processor at 2.39GHz.

TABLE I

ISNR VALUES (IN dB) OBTAINED BY THE FOUR ALGORITHMS TESTED (SEE THE ACRONYMS IN THE TEXT) FOR TV-BASED IMAGE DEBLURRING.

Blur	BSNR	Cameraman				Lena			
		TV-BC	TV-ET	TV-CG	TV-MD	TV-BC	TV-ET	TV-CG	TV-MD
Uniform	30dB	1.05	4.42	5.47	5.44	0.44	3.07	5.52	5.26
Out-of-focus	30dB	1.41	5.18	5.68	5.66	0.70	4.32	6.02	5.83
Linear motion	30dB	2.09	6.83	8.26	8.24	1.17	5.88	7.85	7.54
Gaussian	30dB	1.63	3.25	3.21	3.21	0.82	3.31	3.08	3.03
Average for	30dB	1.54	4.92	5.65	5.64	0.78	4.14	5.61	5.42
Uniform	40dB	1.04	4.89	7.07	7.02	0.44	3.13	7.18	7.05
Out-of-focus	40dB	1.40	6.60	8.34	8.34	0.70	4.62	8.25	7.94
Linear motion	40dB	2.07	8.30	12.57	12.41	1.16	6.31	11.24	11.08
Gaussian	40dB	1.61	4.01	4.03	4.03	0.80	3.84	3.85	3.82
Average for	40dB	1.53	5.95	8.02	7.95	0.77	4.47	7.63	7.47
Uniform	50dB	1.04	5.82	9.77	9.75	0.44	3.14	9.09	9.06
Out-of-focus	50dB	1.40	7.35	11.85	11.78	0.70	4.67	10.92	10.80
Linear motion	50dB	2.06	8.64	16.87	16.67	1.17	6.37	12.33	12.59
Gaussian	50dB	1.61	4.44	4.77	4.78	0.79	4.00	4.48	4.47
Average for	50dB	1.53	6.56	10.81	10.77	0.77	4.54	9.20	9.23
Uniform	60dB	1.04	6.24	11.59	11.95	0.44	3.14	9.84	10.41
Out-of-focus	60dB	1.40	7.51	14.68	14.89	0.70	4.67	12.82	13.27
Linear motion	60dB	2.07	8.69	20.03	19.88	1.16	6.38	13.71	13.56
Gaussian	60dB	1.53	4.67	4.91	4.97	0.80	4.02	4.60	4.70
Average for	60dB	1.53	6.78	12.80	12.92	0.77	4.55	10.24	10.48
Global average		1.53	6.05	9.32	9.32	0.77	4.43	8.17	8.15

TABLE II

ISNR VALUES (IN dB) OBTAINED BY THE FOUR ALGORITHMS TESTED (SEE THE ACRONYMS IN THE TEXT) WITH THE FRAME-BASED ANALYSIS FORMULATION OF IMAGE DEBLURRING.

Blur	BSNR	Cameraman				Lena			
		FA-BC	FA-ET	FA-CG	FA-MD	FA-BC	FA-ET	FA-CG	FA-MD
Uniform	30dB	-0.32	4.48	5.27	5.27	-2.54	2.96	5.07	5.07
Out-of-focus	30dB	0.78	5.17	5.53	5.53	-1.50	4.33	5.51	5.50
Linear motion	30dB	0.91	7.27	8.11	8.12	-1.85	6.18	7.55	7.50
Gaussian	30dB	1.11	2.98	2.93	2.92	-0.32	2.90	2.62	2.63
Average for	30dB	0.62	4.98	5.46	5.46	-1.55	4.09	5.19	5.18
Uniform	40dB	-0.33	5.09	7.19	7.17	-2.54	3.05	6.91	6.83
Out-of-focus	40dB	0.78	7.33	8.51	8.51	-1.50	4.88	7.95	7.95
Linear motion	40dB	0.91	9.38	12.58	12.59	-1.84	7.12	11.29	11.22
Gaussian	40dB	1.10	3.79	3.79	3.80	-0.33	3.56	3.48	3.48
Average for	40dB	0.62	6.40	8.02	8.02	-1.55	4.65	7.41	7.37
Uniform	50dB	-0.34	6.06	10.00	9.99	-2.53	3.06	8.95	9.02
Out-of-focus	50dB	0.79	8.62	12.10	12.10	-1.50	5.02	11.03	10.99
Linear motion	50dB	0.91	9.86	16.89	16.82	-1.84	7.29	14.70	14.63
Gaussian	50dB	1.10	4.49	4.67	4.67	-0.33	3.87	4.27	4.21
Average for	50dB	0.62	7.26	10.91	10.90	-1.55	4.81	9.74	9.71
Uniform	60dB	-0.33	6.59	12.32	12.52	-2.52	3.06	10.21	10.63
Out-of-focus	60dB	0.78	8.88	15.79	15.77	-1.50	5.04	14.15	14.21
Linear motion	60dB	0.92	9.92	20.37	20.34	-1.84	7.31	16.31	16.41
Gaussian	60dB	1.10	4.76	4.99	5.01	-0.33	3.94	4.61	4.59
Average for	60dB	0.62	7.54	13.37	13.41	0.77	4.84	11.32	11.46
Global average		0.62	6.55	9.44	9.45	-1.55	4.60	8.41	8.43

TABLE III
COMPUTATION TIMES OF THE FOUR ALGORITHMS TESTED; THE TIMES (IN SECONDS) REPORTED IN EACH ROW CORRESPOND TO AVERAGES OVER THE FOUR BLURS AT THE INDICATED BSNR VALUES.

BSNR	Cameraman				Lena			
	TV-MD	TV-CG	FA-MD	FA-CG	TV-MD	TV-CG	FA-MD	FA-CG
30dB	1.32	2.62	1.87	3.05	2.65	3.69	3.08	4.03
40dB	1.52	2.96	2.39	3.65	2.85	4.13	3.60	5.04
50dB	2.28	4.20	4.70	7.38	3.32	5.18	5.97	8.76
60dB	8.63	16.54	16.25	26.87	11.02	21.10	16.28	25.77
Average	3.44	6.58	6.30	10.24	4.96	8.53	7.23	10.90

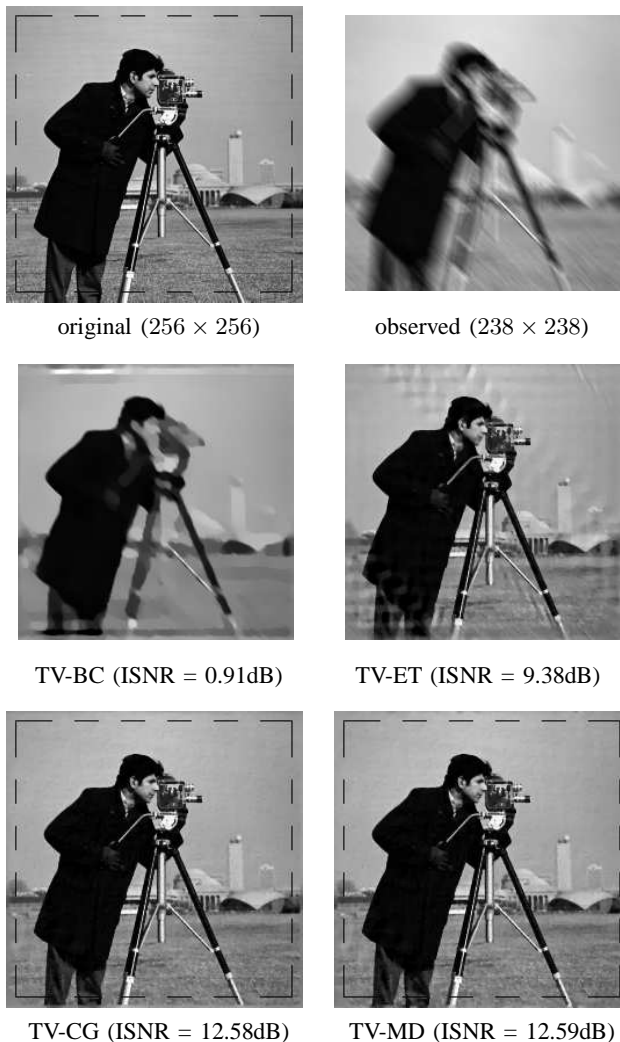


Fig. 3. Results obtained on the *cameraman* image, degraded by a 19×19 linear motion blur, at 40dB BSNR, by the four algorithms considered (see text for the acronyms). Notice that FA-BC and FA-ET produce 238×238 images, while FA-CG and FA-MD yield 256×256 images; the dashed square shows the limit of the boundary region.

the context of ADMM-based deblurring algorithms, the mask decoupling approach is preferable to the use of the Reeves technique [35] with a CG-based inner step.

Finally, we illustrate the successful application of the proposed TV-MD and TV-CG approaches in simultaneous non-periodic deblurring and inpainting (the FA-MD and FA-CG algorithms yield very similar results). In this case, the methods

TABLE IV
AVERAGE TIME PER ITERATION (IN SECONDS) OF EACH OF THE FOUR ALGORITHMS CONSIDERED.

Algorithm	TV-MD	TV-CG	FA-MD	FA-CG
Time per iteration	0.033	0.058	0.059	0.096

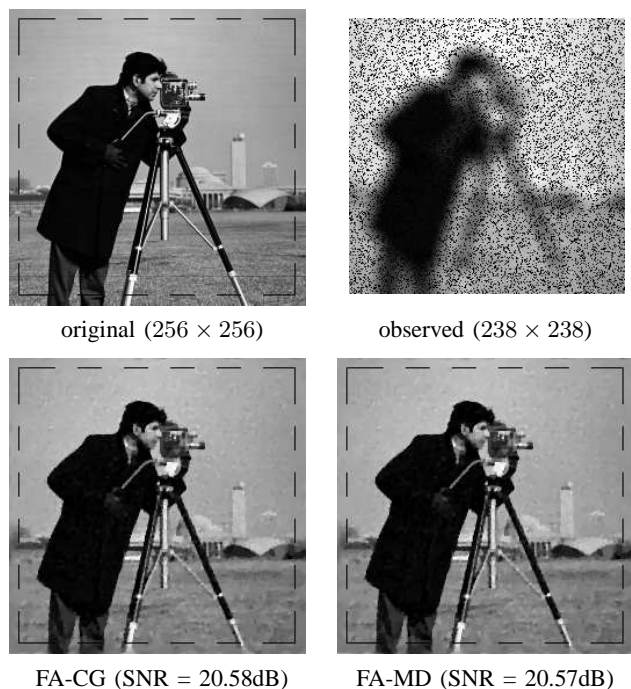


Fig. 4. Illustration on the use of the proposed method for simultaneous deblurring and inpainting. The observed image suffered a 19×19 uniform blurring, followed by the loss of 20% of (randomly located) pixels.

that assume periodic BC (with or without the edgetaper) simply cannot be used. Figure 4 shows the results obtained with the “cameraman” image, non-periodically blurred with a 19×19 uniform blur (using the same boundary conditions as in the previous experiments), at 40dB BSNR, and with 20% missing pixels. Of course, these results are not a full experimental assessment of the effectiveness of the proposed algorithms for image inpainting, which will be addressed in future work.

VI. CONCLUSIONS AND FUTURE WORK

We have presented a new strategy to extend recent fast image deconvolution algorithms, based on the *alternating*

direction method of multipliers (ADMM), to problems with unknown boundary conditions; previous versions of this class of methods were limited to deconvolution problems with periodic boundary conditions. In fact, the proposed algorithms are able to address a more general class of problems, where the degradation model includes not only a convolution with some blur filter, but also loss of pixels (the so-called image inpainting problem). We have considered total-variation regularization as well as frame-based analysis and synthesis formulations, and gave convergence guarantees for the algorithms proposed. Experiments using large blur filters (where the effect of the unknown boundaries is more evident) and several noise levels showed the adequacy of the proposed approach.

Ongoing and future work includes the application and extension of the approach herein developed to: video deblurring; image and video super-resolution; spatially varying regularization. Another direction of current research concerns the application of the proposed approach in iterative blind deconvolution [5].

REFERENCES

- [1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An augmented Lagrangian approach to linear inverse problems with compound regularization," in *IEEE Intern. Conf. on Image Processing – ICIP*, 2010.
- [2] —, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. on Image Processing*, vol. 19, pp. 2345–2356, 2010.
- [3] —, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, pp. 681–695, 2011.
- [4] —, "Non-cyclic deconvolution using an augmented Lagrangian method," in *IEEE International Conference on Computer as a Tool (EUROCON)*, 2011, pp. 1–4.
- [5] M. Almeida and M. A. T. Figueiredo, "Blind deblurring of images with unknown boundaries using the alternating direction method of multipliers," 2013, submitted.
- [6] H. Andrews and B. Hunt, *Digital Image Restoration*. Prentice-Hall, 1977.
- [7] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, pp. 183–202, 2009.
- [8] J. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. on Image Processing*, vol. 16, pp. 2992–3004, 2007.
- [9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [10] J.-F. Cai, S. Osher, and Z. Shen, "Split Bregman methods and frame based image restoration," *Multiscale Model. Simul.*, vol. 8, no. 2, pp. 337–369, 2009.
- [11] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 89–97, 2004.
- [12] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numerische Mathematik*, vol. 76, pp. 167–188, 1997.
- [13] T. F. Chan, A. M. Yip, and F. E. Park, "Simultaneous total variation image inpainting and blind deconvolution," *International Journal of Imaging Systems and Technology*, vol. 15, pp. 92–102, 2005.
- [14] P. L. Combettes and J.-C. Pesquet, "A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, pp. 564–574, 2007.
- [15] P. L. Combettes and V. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM Journal on Multiscale Modeling & Simulation*, vol. 4, pp. 1168–1200, 2005.
- [16] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.
- [17] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Department of Computational and Applied Mathematics, Rice University, Tech. Rep. 12–14, 2012.
- [18] M. Donatelli, C. Estatico, A. Martinelli, and S. Serra-Capizzano, "Improved image deblurring with anti-reflective boundary conditions and re-blurring," *Inverse Problems*, vol. 22, pp. 2035–2053, 2006.
- [19] J. Eckstein and D. P. Bertsekas, "On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.
- [20] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, pp. 947–968, 2007.
- [21] E. Esser, "Applications of Lagrangian based alternating direction methods and connections to split Bregman," UCLA, Tech. Rep. 09-31, 2009.
- [22] M. Figueiredo and R. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. on Image Processing*, vol. 12, pp. 906–916, 2003.
- [23] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. on Image Processing*, vol. 19, pp. 3133–314, 2010.
- [24] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. on Image Processing*, pp. 906–916, 2003.
- [25] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via infinite element approximations," *Computers and Mathematics with Applications*, vol. 2, p. 1740, 1976.
- [26] T. Goldstein and S. Osher, "The split Bregman method for l1-regularized problems," *SIAM Journal on Imaging Science*, vol. 2, pp. 323–343, 2009.
- [27] R. Liu and J. Jia, "Reducing boundary artifacts in image deconvolution," in *IEEE Intern. Conf. on Image Processing – ICIP*, 2008, pp. 505–508.
- [28] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 2009.
- [29] A. Matakos, S. Ramani, and J. Fessler, "Image restoration using non-circulant shift-invariant system models," in *IEEE Intern. Conf. on Image Processing – ICIP*, 2012, pp. 3061–3064.
- [30] —, "Accelerated edge-preserving image restoration without boundary artifacts," *IEEE Trans. on Image Processing*, 2013, to appear.
- [31] M. Ng, *Iterative methods for Toeplitz systems*. Oxford University Press, 2004.
- [32] M. Ng, R. Chan, and W.-C. Tang, "A fast algorithm for deblurring models with Neumann boundary conditions," *SIAM Journal on Scientific Computing*, vol. 21, pp. 851–866, 1999.
- [33] N. Pustelnik, J.-C. Pesquet, and C. Chau, "Relaxing tight frame condition in parallel proximal methods for signal restoration," *IEEE Transactions on Signal Processing*, vol. 60, pp. 968–973, 2012.
- [34] S. Ramani and J. Fessler, "A splitting-based iterative algorithm for accelerated statistical X-ray CT reconstruction," *IEEE Trans. on Medical Imaging*, vol. 31, pp. 677–688, 2012.
- [35] S. J. Reeves, "Fast image restoration without boundary artifacts," *IEEE Trans. on Image Processing*, vol. 14, no. 10, pp. 1448–1453, 2005.
- [36] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica-D*, vol. 60, pp. 259–268, 1992.
- [37] I. Selesnick and M. Figueiredo, "Signal restoration with overcomplete wavelet transforms: Comparison of analysis and synthesis priors," in *Proceedings of SPIE*, vol. 7446, 2009.
- [38] G. Steidl and T. Teuber, "Removing multiplicative noise by Douglas-Rachford splitting methods," *Journal of Mathematical Imaging and Vision*, vol. 36, no. 2, pp. 168–184, 2010.
- [39] M. Tao and J. Yiang, "Alternating direction algorithms for total variation deconvolution in image reconstruction," Department of Mathematics, Nanjing University, Tech. Rep. TR0918, 2009, http://www.optimization-online.org/DB_HTML/2009/11/2463.html.
- [40] M. Šorel, "Removing boundary artifacts for real-time iterated shrinkage deconvolution," *IEEE Trans. on Image Processing*, vol. 21, no. 4, pp. 2329–2334, 2012.
- [41] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal on Imaging Science*, vol. 1, pp. 248–272, 2008.
- [42] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. on Signal Processing*, vol. 57, pp. 2479–2493, 2009.
- [43] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for ℓ_1 -minimization with applications to compressed sensing," *SIAM Journal on Imaging Sciences*, vol. 1, pp. 143–168, 2008.
- [44] L. Zappella, A. Del Bue, X. Llado, and J. Salvi, "Simultaneous motion segmentation and structure from motion," in *IEEE Workshop on Applications of Computer Vision*, 2011, pp. 679–684.

