

# A Framework for Fast Image Deconvolution with Incomplete Observations

Miguel Simões, Luis B. Almeida, José Bioucas-Dias, *Member, IEEE*, Jocelyn Chanussot, *Fellow, IEEE*,

**Abstract**—In image deconvolution problems, the diagonalization of the underlying operators by means of the FFT usually yields very large speedups. When there are incomplete observations (e.g., in the case of unknown boundaries), standard deconvolution techniques normally involve non-diagonalizable operators, resulting in rather slow methods, or, otherwise, use inexact convolution models, resulting in the occurrence of artifacts in the enhanced images. In this paper, we propose a new deconvolution framework for images with incomplete observations that allows us to work with diagonalized convolution operators, and therefore is very fast. We iteratively alternate the estimation of the unknown pixels and of the deconvolved image, using, e.g., an FFT-based deconvolution method. This framework is an efficient, high-quality alternative to existing methods of dealing with the image boundaries, such as edge tapering. It can be used with any fast deconvolution method. We give an example in which a state-of-the-art method that assumes periodic boundary conditions is extended, through the use of this framework, to unknown boundary conditions. Furthermore, we propose a specific implementation of this framework, based on the alternating direction method of multipliers (ADMM). We provide a proof of convergence for the resulting algorithm, which can be seen as a “partial” ADMM, in which not all variables are dualized. We report experimental comparisons with other primal-dual methods, where the proposed one performed at the level of the state of the art. Four different kinds of applications were tested in the experiments: deconvolution, deconvolution with inpainting, superresolution, and demosaicing, all with unknown boundaries.

**Index Terms**—Deconvolution, incomplete observations, convex non-smooth optimization, alternating direction method of multipliers (ADMM), primal-dual optimization, inpainting, superresolution, demosaicing.

## I. INTRODUCTION

**D**EBLURRING is one of the classical problems of the image processing field. It consists in the recovery of sharp images from blurred ones, where the blur can be any kind of degradation that results in a decrease of image sharpness. The blur can be caused, for example, by camera motion, or by the propagation of light through the atmosphere. For a review of deblurring methods, see [1] and the more

Miguel Simões is with Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Portugal, and GIPSA-Lab, Université de Grenoble, France.

Luis B. Almeida and José Bioucas-Dias are with Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Portugal.

Jocelyn Chanussot is with GIPSA-Lab, Grenoble Institute of Technology, France and with the Faculty of Electrical and Computer Engineering, University of Iceland.

This work was partially supported by the Fundação para a Ciência e Tecnologia, Portuguese Ministry of Science and Higher Education, project PEst-OE/EEI/0008/2013 and grant SFRH/BD/87693/2012.

recent [2]. In this work, we consider the situation in which some pixels of the blurred image are not observed. Examples are the pixels corresponding to the boundaries of the image, saturated or missing pixels, and the extra pixels that would have been observed if the sensors that acquired the images had a higher spatial resolution than the actual sensors that were used, as in the case of superresolution [3] and demosaicing [4]. We assume that the blur corresponds to a convolution with a known *point spread function* (PSF). Since the blurring PSF is assumed to be known, this corresponds to the so-called non-blind deconvolution problem, but is also of use in blind deconvolution algorithms, since most of these involve non-blind deconvolution as one of the processing steps.

In this paper, bold lowercase letters denote column vectors, and bold uppercase letters denote matrices.  $\mathbf{I}_p$  denotes the identity matrix of size  $p \times p$ .  $\mathbf{1}_p$  denotes a vector of ones of size  $p$ , and  $\mathbf{0}$  denotes a zero vector or matrix of appropriate size. A superscript on a vector, as in  $\mathbf{x}^i$ , denotes the index of a sequence of vectors. We will use the notation  $\{\mathbf{x}^i\}$  as a shorthand for representing the sequence  $\{\mathbf{x}^i\}_{i=1}^{+\infty}$ . The subgradient operator will be denoted by  $\partial$ . The inverse of matrix  $\mathbf{A}$  will be denoted by  $\mathbf{A}^{-1}$  and the  $\mathbf{A}$ -norm will be denoted by  $\|\cdot\|_{\mathbf{A}}$ , i.e.,  $\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^T \mathbf{A} \mathbf{x}}$ , where  $\mathbf{A}$  is positive-definite and  $\mathbf{x}$  is a vector.

### A. Problem statement

Consider the estimation of a sharp image from a blurred one. Assume that the support of the convolution kernel has size  $(2b + 1) \times (2b + 1)$  pixels and is centered at the origin.<sup>1</sup> Let the size of the blurred image be  $m \times n$ . To express that image as a function of the sharp one, we need to consider a region of the sharp image of size  $m' \times n'$ , with  $m' = m + 2b$  and  $n' = n + 2b$ ; the central  $m \times n$  zone of this sharp image is in the same spatial location as the blurred image. Fig. 1 illustrates this situation.

Let  $\mathcal{X}$  be a given nonempty convex subset of  $\mathbb{R}^{m'n'}$ . Assuming a linear observation model with additive noise, we can express the blurring operation as

$$\mathbf{y} = \mathbf{T}\mathbf{x} + \mathbf{n}, \quad (1)$$

in which the images are represented by column vectors with the pixels arranged in lexicographic order,  $\mathbf{y} \in \mathbb{R}^{mn}$  is the observed image,  $\mathbf{x} \in \mathcal{X}$  is the sharp image,  $\mathbf{T}$  is a  $mn \times m'n'$  block-Toeplitz-Toeplitz-block (BTTB) matrix such that

<sup>1</sup>For simplicity, we consider only kernels supported in square regions centered at the origin. The extension to other situations would be straightforward.

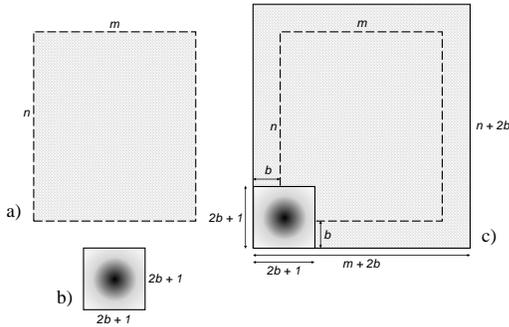


Fig. 1. An illustration of the dimensions of the images involved in typical deblurring problems. a) Blurred image. b) Blurring kernel. c) Sharp image.

$\mathbf{T}\mathbf{x}$  represents the convolution of the sharp image with the blurring PSF, and  $\mathbf{n} \in \mathbb{R}^{mn}$  is the observation noise.

In this paper, we address the estimation of the sharp image  $\mathbf{x}$  from the blurred one, in situations in which some of the pixels of the blurred image are unobserved. This situation occurs, for example, in inpainting and superresolution problems. Demosaicing is a form of superresolution, and also involves unobserved pixels. Often, in simple deconvolution problems, one is only interested in estimating the central  $m \times n$  region of  $\mathbf{x}$ , which we shall designate by *cropped sharp image*, denoted by  $\bar{\mathbf{x}}$ . If one is interested in estimating the whole image  $\mathbf{x}$ , this can be seen as an inpainting problem, since we are estimating the pixels of  $\mathbf{x}$  in a boundary zone of width  $b$  around the central  $m \times n$  region, and this zone is not present in  $\mathbf{y}$ . Even if we do not wish to estimate this zone, the need to properly handle the boundary zone still exists in most real-life deconvolution methods, as we discuss next.

A difficulty with the use of model (1) is that deconvolution methods based on it normally involve products by large BTTB matrices and/or the inversion of such matrices, and both operations are computationally heavy. In order to obtain fast deconvolution methods, many authors replace (1) with models that involve simpler computations. One of the most frequently used models is

$$\mathbf{y} = \bar{\mathbf{T}}\bar{\mathbf{x}}' + \mathbf{n}, \quad (2)$$

in which  $\bar{\mathbf{T}}$  is a block-circulant-circulant-block (BCCB) matrix of size  $mn \times mn$ , and  $\bar{\mathbf{x}}' \in \mathbb{R}^{mn}$  is an approximation of the true cropped sharp image  $\bar{\mathbf{x}}$ .  $\bar{\mathbf{T}}\bar{\mathbf{x}}'$  represents the circular convolution of  $\bar{\mathbf{x}}'$  with the blurring PSF. The speed advantage of (2) comes from the fact that BCCB matrices are diagonalized by the two-dimensional discrete Fourier transform (DFT), and therefore products and inverses involving such matrices can be efficiently computed using the FFT. However, the fact that  $\bar{\mathbf{x}}'$  is only an approximation of the true image  $\bar{\mathbf{x}}$  means that (2) is not an exact model of the convolution process. As a consequence of this, the sharp images obtained by these methods normally exhibit artifacts, typically in the form of ringing. The use of model (2) is often referred to as the use of *periodic boundary conditions*, because it is equivalent to the use of (1) with the true sharp image  $\mathbf{x}$  replaced with an image obtained by periodically repeating  $\bar{\mathbf{x}}$  in the horizontal and vertical directions, and then retaining only the central  $m' \times n'$  region of the resulting periodic image. Other possibilities exist for obtaining fast deconvolution methods, besides the use of periodic boundary

conditions. For example, one can use *reflexive* or *anti-reflexive* boundary conditions, which, under appropriate assumptions, lead to matrices that are diagonalizable, respectively, by the 2D discrete cosine transform and the 2D discrete sine transform, and therefore also yield significant speed advantages [5], [6], [7]. However, the use of any of these boundary conditions (and, in fact, the use of any artificially imposed boundary conditions) corresponds to the use of an inexact convolution model, and therefore gives rise to artifacts.

The occurrence of artifacts can be completely eliminated by the use of an exact model of the convolution process. A relatively recent method, which we shall designate by AM [8], [9], uses a model of the form

$$\mathbf{y} = \mathbf{M}\tilde{\mathbf{T}}\mathbf{x} + \mathbf{n}, \quad (3)$$

in which  $\tilde{\mathbf{T}}$  is an  $m'n' \times m'n'$  BCCB matrix that corresponds to a circular convolution with the blurring PSF, and  $\mathbf{M}$  is an  $mn \times m'n'$  masking matrix that selects, from  $\tilde{\mathbf{T}}\mathbf{x}$ , only the pixels that correspond to the observed image, discarding a boundary zone of width  $b$  in the periphery of the image  $\mathbf{T}\mathbf{x}$ . The circular convolution  $\tilde{\mathbf{T}}\mathbf{x}$  in (3) only differs from the linear (i.e., non-circular) convolution  $\mathbf{T}\mathbf{x}$  of (1) by the presence of that boundary zone, and therefore Eq. (2) is an exact model of the convolution process. Computationally, this method has the advantage of using a diagonalizable matrix,  $\tilde{\mathbf{T}}$ , but needs to deal with the fact that  $\mathbf{M}\tilde{\mathbf{T}}$  is not easily diagonalizable. This difficulty is circumvented, in AM, through an adaptation of the Alternating Directions Method of Multipliers (ADMM) [10], [11]. By means of the splitting of a variable, AM decouples the matrix  $\mathbf{M}$ , which is diagonalizable in the spatial domain, from  $\tilde{\mathbf{T}}$ , which is diagonalizable in the frequency domain, thereby allowing a significant speedup to be achieved.

The deconvolution framework proposed in the present work uses a different, but also exact, convolution model. The convolution process is modeled as

$$\tilde{\mathbf{y}} = \tilde{\mathbf{T}}\mathbf{x} + \mathbf{n}, \quad (4)$$

in which  $\tilde{\mathbf{T}}$  is the same BCCB matrix as in (3), and  $\tilde{\mathbf{y}} \in \mathbb{R}^{m'n'}$  represents the observed image  $\mathbf{y}$  surrounded by the boundary region of width  $b$  mentioned in the previous paragraph. In the proposed framework, this boundary region is estimated, instead of being masked out, as happened in AM. We will present two implementations of this framework: an implementation using an off-the shelf deconvolution method that assumes circular boundary conditions, and an efficient implementation using a “partial” ADMM in which not all variables are dualized, and for which we present a convergence theorem.

Deconvolution methods that do not impose boundary conditions, such as AM and the method proposed in this paper, are often referred to as methods that use *unknown boundaries*. We will use that nomenclature in this paper.

## B. Related work

As mentioned above, many of the published deconvolution methods, such as [5], [6], [7], [12], use artificially imposed boundary conditions, leading to the occurrence of artifacts. Some methods try to reduce the intensity of those artifacts

by preprocessing the observed image's borders, e.g., through edge tapering or smoothing. A brief review of preprocessing methods is given in [13]. Some other methods, such as [14], [15], consider the blurred image as being extended with a boundary zone as in (4), but, instead of correctly estimating that zone, use a synthetic extension that is relatively easy to compute. Since the contents of that extension are artificially imposed, these methods give rise to artifacts.

Methods that use exact models have the potential to completely eliminate the occurrence of artifacts. One such method is the one proposed in [13], which implicitly uses model (4), although that model is not explicitly mentioned in the paper. That method is limited to the use of quadratic regularizers, which allow a fast implementation but yield relatively low-quality deconvolution results. The method has been extended in [16], allowing the use of more general regularizers. In the latter form, the method achieves a relatively high speed by limiting the estimation of the boundary zone (an estimation that is time-consuming, in that method's formulation) to a few initial iterations of the optimization procedure. This version of the method improves on the results of [13] due to the use of more appropriate regularizers, but the imperfect estimation of the boundary zone gives rise to artifacts in the deblurred images. Similar approaches for astronomical images can be found in [17], [18]. Another method, proposed in [19], uses model (1), and is rather slow due to the use of non-diagonalizable BTTB matrices.

As previously said, the AM method [8], [9] uses model (3) and is based on ADMM. Compared to other algorithms, the use of ADMM that is made in that method has some convenient properties: its convergence is guaranteed under rather general conditions, and each iteration only involves operations with a low computational cost (e.g., the inversion of diagonalizable matrices). ADMM is a primal-dual method, which means that it solves a primal optimization problem as well as its dual convex formulation, in the sense of the Fenchel-Rockafellar duality theory (see [20], [21], [22] for recent reviews of proximal and primal-dual methods). In Section III, we experimentally compare AM with the ADMM-based method proposed in the present paper, and find the performances of both to be similar.

The works [23] and [24] propose primal-dual methods that do not involve the inversion of large BTTB matrices. In Section III, we experimentally test one of these methods [23], and find it to be rather slow. Another primal-dual method that considers non-periodic boundaries was proposed in [25]. It needs the structure of the boundaries to be known *a priori*, which normally is not the case in real-life situations. The use of an artificially chosen structure results, once again, in the occurrence of artifacts.

In [26], a method that has some resemblance to our proposed deconvolution framework was introduced, in the context of the solution of systems of linear equations with Toeplitz system matrices.

### C. Contributions and outline

This work has two main contributions:

- We propose a new framework for solving deconvolution problems with unobserved pixels. This framework is an efficient, high-quality alternative to the use of heuristic methods, such as edge tapering, to reduce the artifacts produced by deconvolution methods that assume periodic boundary conditions. We give an example of how this framework can be used to extend a state-of-the-art deconvolution method to the use of unknown boundaries.
- The proposed framework can also be used to develop new deconvolution methods. We propose a specific ADMM-based implementation of the framework, for which we give a proof of convergence. We experimentally compare it with some state-of-the-art methods.

The structure of this paper is as follows. Section II describes the proposed framework and the ADMM-based implementation, and gives the convergence proof for the latter. Section III presents experimental results, and has two parts: the first one illustrates the use of the proposed framework to convert an off-the-shelf, state-of-the-art deblurring method (IDDBM3D [12]) to the use of unknown boundaries; the second part presents a comparison between the proposed ADMM-based method and some other published ones, in problems of deconvolution with and without inpainting, of superresolution, and of demosaicing, all with unknown boundaries. Section IV concludes.

## II. THE PROPOSED FRAMEWORK

### A. Basic structure

As previously mentioned, the framework that we propose is based on model (4). From here on, we'll express the extended blurred image  $\tilde{\mathbf{y}}$  in a form that is more convenient for the treatment that follows, and that encompasses not only the case of unknown boundaries, but also all the other cases of unobserved pixels. We'll denote the number of observed pixels of the blurred image by  $k$ , and the number of unobserved pixels (including the above-mentioned boundary zone) by  $d$ . Let  $\mathbf{z} \in \mathcal{Z}$  and  $\mathbf{y} \in \mathbb{R}^k$  be column vectors containing, respectively, the elements of  $\tilde{\mathbf{y}}$  that correspond to unobserved pixels and those that correspond to observed pixels;  $\mathcal{Z}$  is some given nonempty convex subset of  $\mathbb{R}^d$ . In a simple deconvolution problem with unknown boundaries,  $\mathbf{z}$  will contain the boundary zone, and  $\mathbf{y}$  will contain the observed blurred image. In a combined deconvolution and inpainting problem,  $\mathbf{z}$  will contain both the boundary zone and the additional unobserved pixels, and  $\mathbf{y}$  will contain the pixels of the blurred image that were actually observed. We will reorder the elements of the extended image  $\tilde{\mathbf{y}}$  as  $\begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \mathbf{P}\tilde{\mathbf{y}}$ , where  $\mathbf{P}$  is an appropriate permutation matrix, so that the observed pixels are in the first positions and the unobserved pixels are in the last positions of the vector  $\begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix}$ .

Conceptually, the proposed framework is rather simple. It consists of using the blurring model (4), and alternately estimating  $\mathbf{x}$  and  $\mathbf{z}$ , as shown in Fig. 2. In this framework, step 2, which estimates  $\mathbf{x}$ , can be performed, essentially, with any existing deblurring method that assumes circular

### Framework 1

1. Initialize  $\mathbf{x}^1$  and  $\mathbf{z}^1$ . Let  $i = 1$ .

**repeat**

2. Compute  $\mathbf{x}^{i+1}$  given  $\mathbf{x}^i, \mathbf{z}^i$ .

3. Compute  $\mathbf{z}^{i+1}$  given  $\mathbf{x}^{i+1}$ .

4. Increment  $i$ .

**until** stopping criterion is satisfied.

Fig. 2. The proposed deblurring framework.

boundary conditions.<sup>2</sup> Step 3 is performed by just computing the reconstructed blurred image, given by  $\tilde{\mathbf{T}}\mathbf{x}$ , and selecting from it the pixels that correspond to  $\mathbf{z}$ .

An important difference of the proposed framework relative to most published deblurring methods, including AM, is that, in this framework, the unobserved pixels of the blurred image (represented by  $\mathbf{z}$ ) are explicitly estimated. This means that we have one more variable to estimate ( $\mathbf{z}$ ).

As given in Fig. 2, the proposed framework is rather general. It can be used to design new deblurring methods, an example of which is the efficient ADMM-based method that we propose in Section II-B. It is also an efficient, high-quality alternative to methods such as edge tapering, to convert existing deblurring methods that impose specific boundary conditions into methods that work with unknown boundaries. We illustrate this in Section III-A, by using the proposed framework to convert an off-the-shelf, state-of-the-art deblurring method that assumes circular boundary conditions (IDD-BM3D) into a method that uses unknown boundaries.

### B. ADMM-based method

ADMM has been widely used, in recent years, to solve high-dimensional problems in signal and image processing, due to its ability to yield high-quality solutions in a computationally efficient way, in many practical situations. In its original formulation, ADMM can be used to solve problems of the form

$$\underset{\mathbf{u}}{\text{minimize}} \quad f(\mathbf{u}) + \psi(\mathbf{K}\mathbf{u}),$$

where both  $f$  and  $\psi$  are closed, proper convex, possibly non-smooth functions (see Appendix A for details). In comparison with some other primal-dual methods, in the context of image deconvolution, ADMM uses an extra set of variables and involves a matrix inversion. These characteristics, however, apparently are the ones that allow it to remain quite competitive relative to more recent primal-dual methods, as illustrated, for example, by the results that we report in Sections III-B to III-D.

In the ADMM-based implementation of Framework 1 that we propose, we'll use the blurring model of Eq. (4). The noise  $\mathbf{n}$  will be assumed to be i.i.d. Gaussian. We'll use a maximum-a-posteriori (MAP) formulation, and consequently, the data-

<sup>2</sup>The framework can also be used with methods that use other boundary conditions. The only difference will be in the structure of the matrix  $\tilde{\mathbf{T}}$  used in model (4). Instead of being a BCCB matrix, it will have the proper structure for the boundary conditions under consideration.

fitting term of our objective function will be given by

$$f(\mathbf{x}, \mathbf{z}) = \frac{1}{2} \left\| \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} - \mathbf{H}\mathbf{x} \right\|^2, \quad (5)$$

with  $\mathbf{H} = \tilde{\mathbf{P}}\tilde{\mathbf{T}}$  and  $\mathbf{x} \in \mathcal{X}$ , where  $\mathcal{X}$  is some given convex subset of  $\mathbb{R}^{k+d}$ .

The problem to be solved will be expressed as

$$\underset{\mathbf{x}, \mathbf{z}}{\text{minimize}} \quad f(\mathbf{x}, \mathbf{z}) + \phi(\mathbf{D}\mathbf{x}), \quad (6)$$

where  $\mathbf{D} \in \mathbb{R}^{l \times (k+d)}$  is a matrix that extracts a linear representation of the estimated image, such as edges,  $l$  is the number of components of that representation, and  $\phi(\mathbf{D}\mathbf{x})$  is a regularizer that promotes some desirable characteristic of images, such as sharp edges.

We'll start by considering the use of ADMM in its standard form to solve problem (6). The resulting method will not be very efficient, because it will involve a step that is computationally heavy, but it will be useful to motivate the method that we propose, and to analyze some of its properties. We'll then describe our proposed method, which avoids the above-mentioned computational inefficiency through the use of a *partial* ADMM.<sup>3</sup>

In what follows, we will make use of the variables  $\mathbf{v}, \mathbf{d} \in \mathbb{R}^{l+d}$ , decomposed as  $\mathbf{v} = \begin{bmatrix} \mathbf{v}_x \\ \mathbf{v}_z \end{bmatrix}$  and  $\mathbf{d} = \begin{bmatrix} \mathbf{d}_x \\ \mathbf{d}_z \end{bmatrix}$ , with  $\mathbf{v}_x, \mathbf{d}_x \in \mathbb{R}^l$  and  $\mathbf{v}_z, \mathbf{d}_z \in \mathbb{R}^d$ . To apply ADMM to problem (6), we first define

$$\mathbf{u} = \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_d \end{bmatrix}, \quad \bar{f}(\mathbf{u}) = f(\mathbf{x}, \mathbf{z}),$$

and also define

$$\psi(\mathbf{v}) = \phi(\mathbf{v}_x),$$

so that  $\psi(\mathbf{K}\mathbf{u}) = \phi(\mathbf{D}\mathbf{x})$ .

We rewrite (6) as

$$\underset{\mathbf{u}, \mathbf{v}}{\text{minimize}} \quad \bar{f}(\mathbf{u}) + \psi(\mathbf{v}) \\ \text{subject to} \quad \mathbf{v} = \mathbf{K}\mathbf{u}. \quad (7)$$

Applying ADMM to this problem, we obtain an iteration of the sequence of steps

$$\begin{bmatrix} \mathbf{x}^{i+1} \\ \mathbf{z}^{i+1} \end{bmatrix} \in \arg \min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}, \mathbf{z}) + \frac{\mu}{2} \left\| \mathbf{v}^i - \mathbf{K} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} - \mathbf{d}^i \right\|^2, \quad (8)$$

$$\mathbf{v}^{i+1} \in \arg \min_{\mathbf{v}} \psi(\mathbf{v}) + \frac{\mu}{2} \left\| \mathbf{v} - \mathbf{K} \begin{bmatrix} \mathbf{x}^{i+1} \\ \mathbf{z}^{i+1} \end{bmatrix} - \mathbf{d}^i \right\|^2, \quad (9)$$

$$\mathbf{d}^{i+1} = \mathbf{d}^i - \left( \mathbf{v}^{i+1} - \mathbf{K} \begin{bmatrix} \mathbf{x}^{i+1} \\ \mathbf{z}^{i+1} \end{bmatrix} \right). \quad (10)$$

From now on, we will refer to the iteration of steps (8)–(10) as the *standard ADMM*.

As mentioned above, the standard ADMM won't normally be computationally efficient. This is due to the fact that, in step (8),  $\mathbf{x}$  and  $\mathbf{z}$  need to be estimated simultaneously, and this will normally involve the inversion of a large matrix that is not easily diagonalizable. For the (rather small) images used

<sup>3</sup>We qualify it as "partial" because not all variables are dualized.

in the experimental tests of Section III, which have  $256 \times 256$  pixels, this would involve a matrix with  $(256 \times 256)^2 \approx 4 \times 10^9$  elements. Since matrix inversion runs in  $O[n^3]$  time, directly inverting matrices as large as these would not be feasible, in useful time, with current computers. Furthermore, it would be impracticable to manipulate such large matrices, given the memory sizes of most present-day computers. For the much larger images that are commonly used in practice, the difficulties would be even larger.

To motivate the solution that we propose, we note that if, in problem (6), we consider minimizing relative to  $\mathbf{x}$  and to  $\mathbf{z}$  separately, only the minimization relative to  $\mathbf{x}$  will be difficult to perform. The minimization relative to  $\mathbf{z}$  will be easy to implement in a computationally efficient way, because it is the minimization of a quadratic function, and the matrix  $\mathbf{H} = \mathbf{P}\tilde{\mathbf{T}}$  is diagonalizable in the frequency domain (with an appropriate permutation, corresponding to the product by  $\mathbf{P}$ ). In view of this, we will separate the minimization relative to  $\mathbf{x}$  from the minimization relative to  $\mathbf{z}$ , applying them in an alternating manner, and we will apply the ADMM machinery only to the variable  $\mathbf{x}$ , instead of applying it to  $\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}$ , as happened in the standard ADMM. Of course, the convergence guarantees of the standard ADMM won't apply to the proposed method. We will, therefore, present a convergence proof for it.

Since we are applying the ADMM machinery only to  $\mathbf{x}$ , step (8) of the standard ADMM will be replaced by a minimization of

$$f(\mathbf{x}, \mathbf{z}) + \frac{\mu}{2} \left\| \mathbf{v}_x^i - \mathbf{D}\mathbf{x} - \mathbf{d}_x^i \right\|^2, \quad (11)$$

which we will solve approximately by means of an alternating minimization on  $\mathbf{x}$  and  $\mathbf{z}$  through one or more block-Gauss-Seidel (BGS) passes. Furthermore, steps (9) and (10) will have to be modified so as to refer only to  $\mathbf{x}$ , and not to  $\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}$ . If we use just one BGS pass to minimize (11), the complete method will correspond to the iteration of

$$\mathbf{x}^{i+1} \in \arg \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{z}^i) + \frac{\mu}{2} \left\| \mathbf{v}_x^i - \mathbf{D}\mathbf{x} - \mathbf{d}_x^i \right\|^2, \quad (12)$$

$$\mathbf{z}^{i+1} \in \arg \min_{\mathbf{z}} f(\mathbf{x}^{i+1}, \mathbf{z}), \quad (13)$$

$$\mathbf{v}_x^{i+1} \in \arg \min_{\mathbf{v}_x} \phi(\mathbf{v}_x) + \frac{\mu}{2} \left\| \mathbf{v}_x - \mathbf{D}\mathbf{x}^{i+1} - \mathbf{d}_x^i \right\|^2, \quad (14)$$

$$\mathbf{d}_x^{i+1} = \mathbf{d}_x^i - (\mathbf{v}_x^{i+1} - \mathbf{D}\mathbf{x}^{i+1}). \quad (15)$$

If we use more BGS passes, instead of just one, there will be an inner loop consisting of steps (12) and (13).

As can easily be seen, this method falls within the scope of Framework 1, the main steps being (12) and (13); steps (14) and (15) are added by the use of the ADMM technique. We will call the iteration of (12)–(15) (with one or more BGS passes) the *partial ADMM*; this designation stems from the fact that we only apply the ADMM technique to  $\mathbf{x}$ , and not to  $\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}$ .

We will now address the issue of the convergence of the partial ADMM. We will start by proving (in Theorem 1) the convergence of a somewhat more general method, and we will then show (in Corollary 1) that the partial ADMM is a special case of that method, and is therefore encompassed by Theorem 1.

Until now, we have assumed the data-fitting term  $f$  to be given by (5). For the proof of convergence, we will allow  $f$  to have the more general form

$$f(\mathbf{x}, \mathbf{z}) = \frac{1}{2} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}^T \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix} + \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}^T \begin{bmatrix} \mathbf{e} \\ \mathbf{f} \end{bmatrix} + g, \quad (16)$$

where  $\mathbf{A} \in \mathbb{R}^{(k+d) \times (k+d)}$ ,  $\mathbf{B} \in \mathbb{R}^{(k+d) \times d}$ ,  $\mathbf{C} \in \mathbb{R}^{d \times d}$ ,  $\mathbf{e} \in \mathbb{R}^{k+d}$ ,  $\mathbf{f} \in \mathbb{R}^d$ , and  $g \in \mathbb{R}$ , and where we assume that  $\mathbf{C}$  is positive-definite (PD) and that  $\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$  is positive-semidefinite (PSD). These assumptions guarantee that  $f$  is convex, and are not very restrictive. The set of functions that they encompass is only slightly less general than the set of all convex quadratic functions. To obtain the latter set, the assumption on  $\mathbf{C}$  would have to be relaxed to being PSD, but additional assumptions would need to be made (see, e.g., Appendix A.5.5 of [27]).

The convergence result is given by the following theorem:

**Theorem 1.** *Assume that, in problem (6) with  $f$  defined by (16),  $\mathbf{C}$  is PD,  $\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T$  is PSD,  $\mathbf{D}$  is full column rank, and  $\phi$  is closed proper convex and coercive. Define  $\mathbf{K} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_d \end{bmatrix}$ . Then, the set of solutions of problem (6) is non-empty, the sequence  $\left\{ \begin{bmatrix} \mathbf{x}^i \\ \mathbf{z}^i \end{bmatrix} \right\}$  generated by the partial ADMM converges to a solution of that problem, and the sequence  $\left\{ \mathbf{v}^i \right\}$  converges to  $\mathbf{K} \begin{bmatrix} \mathbf{x}^* \\ \mathbf{z}^* \end{bmatrix}$ , where  $\begin{bmatrix} \mathbf{x}^* \\ \mathbf{z}^* \end{bmatrix}$  is the limit of  $\left\{ \begin{bmatrix} \mathbf{x}^i \\ \mathbf{z}^i \end{bmatrix} \right\}$ .*

*Proof.* The proof is given in Appendix B.  $\square$

**Corollary 1.** *For problem (6) with  $f$  given by (5), Theorem 1 applies.*

*Proof.* If we make  $\mathbf{A} = \mathbf{H}^T\mathbf{H}$ ,  $\mathbf{B}^T = -\mathbf{M}_z\mathbf{H}$ ,  $\mathbf{C} = \mathbf{I}_d$ ,  $\mathbf{e}^T = -\mathbf{y}^T\mathbf{M}_y\mathbf{H}$ ,  $\mathbf{f} = \mathbf{0}$ , and  $g = \frac{1}{2}\mathbf{y}^T\mathbf{y}$ , where  $\mathbf{M}_y = [\mathbf{I}_k \ \mathbf{0}]$  is of size  $k \times (k+d)$  and  $\mathbf{M}_z = [\mathbf{0} \ \mathbf{I}_d]$  is of size  $d \times (k+d)$ , the conditions of Theorem 1 are satisfied.  $\square$

Regarding the practical implementation of the partial ADMM, the solutions of the minimization problems that constitute steps (12) and (13), with  $f$  given by (5), are given, respectively, by

$$\begin{aligned} \mathbf{x}^{i+1} &= [\mathbf{H}^T\mathbf{H} + \mu\mathbf{D}^T\mathbf{D}]^{-1} \\ &\quad [\mathbf{H}^T\mathbf{M}_z^T\mathbf{z}^i + \mathbf{H}^T\mathbf{M}_y^T\mathbf{y} + \mu\mathbf{D}^T(\mathbf{v}_x^i - \mathbf{d}_x^i)] \\ &= [\mathbf{H}^T\mathbf{H} + \mu\mathbf{D}^T\mathbf{D}]^{-1} \\ &\quad \left[ \mathbf{H}^T \begin{bmatrix} \mathbf{y} \\ \mathbf{z}^i \end{bmatrix} + \mu\mathbf{D}^T(\mathbf{v}_x^i - \mathbf{d}_x^i) \right] \end{aligned} \quad (17)$$

and

$$\mathbf{z}^{i+1} = \mathbf{M}_z\mathbf{H}\mathbf{x}^{i+1}. \quad (18)$$

Matrices  $\mathbf{M}_y$  and  $\mathbf{M}_z$  are defined in the proof of Corollary 1. They are masking matrices and, in particular,  $\mathbf{M}_y$  is equivalent to the masking matrix  $\mathbf{M}$  used in the AM method (3), i.e.,  $\mathbf{M} = \mathbf{M}_y\mathbf{P}$ .

In practice, we have found it useful to use over-relaxation with a coefficient of 2 in the update of  $\mathbf{z}$ , and therefore we'll replace (18) with

$$\mathbf{z}^{i+1} = 2\mathbf{M}_z\mathbf{H}\mathbf{x}^{i+1} - \mathbf{z}^i. \quad (19)$$

In (17),  $\mathbf{H}^T \mathbf{H} = \tilde{\mathbf{T}}^T \mathbf{P}^T \mathbf{P} \tilde{\mathbf{T}} = \tilde{\mathbf{T}}^T \tilde{\mathbf{T}}$  is a BCCB matrix. If  $\mathbf{D}^T \mathbf{D}$  is also BCCB, the matrix inverse in (17) can be efficiently computed by means of the FFT. On the other hand, since  $\mathbf{H} = \mathbf{P} \tilde{\mathbf{T}}$ , products by  $\tilde{\mathbf{H}}$  or  $\mathbf{H}^T$  can be computed as products by the BCCB matrix  $\tilde{\mathbf{T}}$  followed or preceded by the appropriate permutation, and therefore can also be efficiently computed by means of the FFT. Consequently, the iterations of the proposed method are computationally efficient, having complexity  $O[(k+d)\log(k+d)]$ .

The MATLAB code for the proposed ADMM-based method is available at <https://github.com/alfaiate/DeconvolutionIncompleteObs>.

### III. EXPERIMENTAL STUDY

This section has two parts. The first one, corresponding to Subsection III-A, illustrates the use of the framework of Fig. 2 with a state-of-the-art deblurring method that assumes periodic boundary conditions, to adapt it to the unknown boundaries situation. We plug into the mentioned framework, without modification, a fast, high-quality FFT-based deconvolution method: IDD-BM3D [12]. As a result of the incorporation into the framework, we obtain a deconvolution method that uses unknown boundaries, and that still retains the speed of FFT-based matrix operations. We compare the results of the resulting method with those of the commonly adopted solution of using edge tapering to deal with the unknown boundaries. The second part of this section, consisting of Subsections III-B to III-D, presents experimental results on the use of the proposed ADMM-based method, and comparisons with AM and with another state-of-the-art method.

In the deblurring tests described ahead, we created each blurred image by performing a circular convolution of the corresponding sharp image with the desired blurring kernel, and then cropping the result, keeping only the part of the circular convolution that coincided with the linear convolution. We then normalized the images so that all pixel values were in the interval  $[0, 1]$ . Finally, we added i.i.d. Gaussian noise with a blurred signal-to-noise ratio (BSNR) of 50 dB, unless otherwise indicated.

#### A. Use of Framework 1 with IDD-BM3D

As previously mentioned, step 2 of Framework 1 can be implemented using, essentially, any deconvolution method. One such method is the state-of-the-art IDD-BM3D [12], which assumes circular boundary conditions to be able to perform fast matrix operations in the frequency domain by means of the FFT. It takes a frame-based approach to the deconvolution problem, and performs both a deconvolution and a denoising step. Additionally, it runs another deconvolution method [28] for initialization. We used the published IDD-BM3D software, without change, to implement step 2 of Framework 1. As mentioned in Section II-A, step 3 was performed by computing  $\tilde{\mathbf{T}}\mathbf{x}$  and selecting from it the pixels that corresponded to  $\mathbf{z}$ .

In the experimental tests, we compared four different situations: (a) direct application of IDD-BM3D without the use of Framework 1, to assess the effect of the method's assumption of circular boundary conditions on an image that didn't obey

those conditions; (b) similar to (a), but preprocessing the observed image by edge tapering, to reduce the effect of the mentioned assumption; (c) the use of IDD-BM3D within Framework 1, as described above; and (d) the use of the *partial ADMM* method (the latter situation was included for completeness, but is not essential to the comparison being made here). In situations (a) and (b), IDD-BM3D was run for 400 iterations; in situation (c), we performed 80 iterations of the main loop of Framework 1, and within each of these, IDD-BM3D was run for 5 iterations, giving, again, a total of 400 iterations of IDD-BM3D; in situation (d), we used the Proposed-AD version of the partial ADMM method (see the next section for details).

We ran the experiment using the *cameraman* image with size  $256 \times 256$  pixels, blurred with a  $9 \times 9$  boxcar filter, and with additive i.i.d. Gaussian noise with a BSNR of 40 dB. In situation (b), the image borders were smoothed with MATLAB's *edgetaper* function with an  $11 \times 11$  boxcar filter. This size of the filter was experimentally chosen so as to yield the best final results in terms of improvement in signal-to-noise ratio (ISNR). In situations (a) and (b), IDD-BM3D estimated sharp images of size  $m \times n$ . In situations (c) and (d), the use of Framework 1 led to estimated images of size  $m' \times n'$ . To ensure a fair comparison, only the central  $m \times n$  regions of the estimated images were used to compute the ISNR, in cases (c) and (d).

The results of the tests, along with the corresponding ISNR values, are shown in Fig. 3. As can be seen, using IDD-BM3D without taking into account that the boundary conditions were not circular (situation (a)) produced very strong artifacts. With edge tapering (situation (b)), the artifacts were much reduced, although some remained visible; there also was some loss of detail near the image borders. With the use of IDD-BM3D within Framework 1 (situation (c)), there were barely any artifacts, and the image remained sharp all the way to the borders. The values of the ISNR agree with these observations. The partial ADMM method (situation (d)) yielded somewhat lower ISNR than IDD-BM3D within Framework 1. This agrees with the fact that IDD-BM3D is among the best existing image deblurring methods, with a quality that is hard to equal.

#### B. Proposed method: deblurring with unknown boundaries

In this and the following subsections, we present results of the ADMM-based deblurring method proposed in Section II-B, and we compare it with two published state-of-the-art methods. The first of these is AM, which was chosen because it is an ADMM-based method specifically developed for the problem of deblurring with unobserved pixels, and therefore bears some resemblance to the proposed partial ADMM. Although the MATLAB code for AM was made available by the authors of [8], we implemented the method ourselves in a form that was more efficient than the published one, by performing the products by the matrices  $\mathbf{D}$  and  $\mathbf{D}^T$  in the space domain instead of the frequency domain (the latter required the computation of FFTs and element-wise multiplications, while our implementation only required subtractions). The second method used for comparison was the primal-dual algorithm



Fig. 3. Use of the IDD-BM3D deblurring method and Framework 1 in a blurred image with unknown boundaries. The experiments were run for the  $256 \times 256$  *cameraman* image blurred with a  $9 \times 9$  boxcar filter. (a) Plain IDD-BM3D (ISNR = -14.62 dB). (b) IDD-BM3D with pre-smoothing of the blurred image's borders (ISNR = 8.27 dB). (c) Framework 1, with step 2 implemented through IDD-BM3D (ISNR = 9.64 dB). (d) ADMM-based implementation of Framework 1 (Proposed-AD version,  $\lambda = 4 \times 10^{-5}$ ) (ISNR = 8.43 dB).

of Condat [23], which we shall denote by CM. We chose this method because it can be expressed in a form that does not require the inversion of matrices related to the blurring operator, and this inversion is a computational bottleneck of most deblurring methods based on exact blurring models, as discussed in Section I-A. While [23] does not explicitly consider the problem of image deblurring with unobserved pixels, in [29] it is shown how to adapt the method to a deblurring and demosaicing problem. We implemented this adaptation of the method in a form appropriate to the observation model of Eq. (3). For completeness, we also show results obtained with an approximation of the standard ADMM. As discussed in Section II-B, the direct application of that method is impracticable, even for small images, in most present-day computers. We give the results obtained by approximately solving Eq. (8) through 1000 conjugate-gradients iterations. This number of iterations was experimentally chosen to approximately yield the best speed. We denote this method by 'ADMM-CG'.

All the methods under comparison minimize equivalent cost functions, and therefore we considered it quite probable that they would yield virtually identical estimated images, even though, in our tests, we were not strictly under the convergence conditions of Theorem 1, for reasons that we will discuss further on. As described ahead, we experimentally confirmed, in a few cases, that all the methods converged to essentially the same images, and therefore we focused the subsequent comparisons on the computational efficiencies of the various methods, and not on the quality of their results. The computation times that we report ahead were obtained using MATLAB on an Intel Core i7 CPU running at 3.20 GHz, with 32 GB of RAM.

We used, in the deblurring tests, two well-known images: *Lena* (in grayscale) and *cameraman*, both of size  $256 \times 256$  pixels. These images were blurred with boxcar filters of sizes between  $3 \times 3$  and  $21 \times 21$ , and with truncated Gaussian filters with supports of the same sizes; for a Gaussian filter with a support of size  $l \times l$ , we used a standard deviation of  $\sqrt{l}$ .

As in [8], [9], [29], we used isotropic TV to regularize our problem, i.e., we made  $\phi(\mathbf{D}\mathbf{x}) = \lambda \sum_{j=1}^{k+d} \sqrt{(\mathbf{D}_h \mathbf{x})_j^2 + (\mathbf{D}_v \mathbf{x})_j^2}$ , where  $(\mathbf{a})_j$  denotes the  $j$ th element of the vector  $\mathbf{a}$ ;  $\mathbf{D}_h$  and  $\mathbf{D}_v$  were such that the products by these matrices computed, respectively, the horizontal and vertical first-order differences of a discrete image, with periodic boundaries;  $\mathbf{D} = [\mathbf{D}_h^T \mathbf{D}_v^T]^T$ ; and  $\lambda > 0$ .

The matrix  $\mathbf{D}^T \mathbf{D}$  was BCCB, as required for both the proposed method and AM to be efficient (CM has no such requirement).

With the isotropic TV regularizer, the solution of problem (14) is obtained by a vector soft-thresholding operation,  $\mathbf{v}_x^{i+1} = \text{vector-soft}(\mathbf{v}_x^i, \frac{\lambda}{\mu})$  [30], [31]. With this regularizer, we can guarantee the uniqueness of the solution of problem (6) if  $\mathcal{N}(\mathbf{H}) \cap \mathcal{N}(\mathbf{D}) = \{\mathbf{0}\}$ , which is true if  $\mathbf{1}_k \notin \mathcal{N}(\mathbf{H})$ , where  $\mathcal{N}(\cdot)$  denotes the null space of a matrix. The latter condition is normally verified, since real-world blur kernels usually have nonzero DC gain.

The guarantee of convergence given by Theorem 1 requires that matrix  $\mathbf{D}$  be full column rank. An assumption of this kind is common in the literature, when studying the convergence of primal-dual methods, even though it can be relaxed in some cases (see, e.g., [11] for the ADMM case). With the isotropic TV regularizer,  $\mathbf{D}$  is not full column rank (its rank is  $k + d - 1$ ), which means that we did not have a formal guarantee of convergence of the proposed method in our experimental tests. This was not the case for AM, where the use of an extra variable splitting makes  $\mathbf{D}$  full rank. It was also not the case for CM, in which  $\mathbf{D}$  does not need to be full rank. In practice, we found that the four methods always converged, and we had some experimental evidence that they all converged to the same solution, as reported ahead. If we wished to have a formal guarantee of convergence of the proposed method, we could have used one of two approaches: (1) to use a new variable to decouple the convolution operator from  $\mathbf{x}$ , which would lead to, e.g.,  $\mathbf{D} = [\mathbf{D}_h^T \mathbf{D}_v^T \mathbf{H}^T]^T$ , which is full column rank if  $\mathcal{N}(\mathbf{D}_h) \cap \mathcal{N}(\mathbf{D}_v) \cap \mathcal{N}(\mathbf{H}) = \{\mathbf{0}\}$  (the regularizer would need to be changed accordingly), or (2) to modify the discrete difference matrices, making them full rank, e.g., by adding to them  $\varepsilon \mathbf{I}_{k+d}$  with a small  $\varepsilon > 0$ .

In the proposed method, the complexity of an iteration was dominated by FFTs: three for steps (12) and (13), and one for step (14); the number of FFTs thus depended on the number of BGS passes (each pass involved three FFTs). In our implementation, the other two methods both involved four FFTs per iteration.<sup>4</sup> Our method required the storage of one variable with dimension  $d$  and five with dimension  $k + d$ . AM required the storage of seven variables with dimension  $k + d$ , and CM required the storage of four with dimension  $k + d$ .

<sup>4</sup>The publicly available implementation of AM requires seven FFTs per iteration.

In order to ensure fair comparisons, the initial estimate of  $\mathbf{x}$  was obtained in the same way for the four methods, by padding the observed image with a boundary zone formed by repeating the border pixels of the observed image. This initialization led to slightly faster results than other alternatives, such as linear interpolation between opposite-border pixels.

All the methods have parameters that need to be adjusted. The optimal tuning of the regularization parameter  $\lambda$  is a complex issue, and a number of techniques have been proposed for that purpose, e.g. using Stein’s unbiased risk estimator [31], generalized cross-validation [32], or a measure of the whiteness of the reconstruction error [33]. Since, as mentioned above, we wanted to focus our comparison on the computational costs, and not on the quality of the deblurring results, we fixed  $\lambda = 5 \times 10^{-6}$ , a value that yielded visually good estimated images.

Another parameter that needed to be set, both in our method, in AM and in ADMM-CG, was the penalization parameter  $\mu$ , which influences the convergence speed. The authors of [8] and of [9] developed their own heuristics to set this parameter. Since heuristics like these require an amount of fine-tuning that can be time-consuming, and since the tuning done for some images may not extend to other images, we instead followed a simple strategy, proposed in [11]. It involves little computation, but requires that we keep track of both  $\mathbf{v}^{i+1}$  and  $\mathbf{v}^i$ . The penalization parameter is adapted in every iteration, according to

$$\mu^{i+1} = \begin{cases} 2\mu^i & \text{if } \|\mathbf{d}^{i+1} - \mathbf{d}^i\| > t\|\mathbf{v}^{i+1} - \mathbf{v}^i\|, \\ 0.5\mu^i & \text{if } \|\mathbf{v}^{i+1} - \mathbf{v}^i\| > t\|\mathbf{d}^{i+1} - \mathbf{d}^i\|, \\ \mu^i & \text{otherwise.} \end{cases} \quad (20)$$

Additionally,  $\mathbf{d}^{i+1}$  needs to be rescaled after updating  $\mu$ : it should be halved in the first case above, and doubled in the second one. In the proposed method, we used  $t = 3$ . We used the same strategy to set the penalty parameters of AM. There are two such parameters in that method: one similar to the one of our method, and another one relating to the variable splitting used to decouple matrix  $\mathbf{M}$  from matrix  $\tilde{\mathbf{T}}$ . We adapted the two parameters independently, and used  $t = 10$  for both because lower values led to instabilities. For ADMM-CG, we used  $t = 3$ . The use of the same strategy to set the penalty parameters of the three methods was intended to make the comparison between both as fair as possible. The convergence proof of Theorem 1 is difficult to adapt to a varying  $\mu$ , but still applies if this parameter is kept fixed after a certain number of iterations.

As already mentioned, another parameter of our method is the number of BGS passes. The proof in Appendix B assumes that a single pass is used, and is easily extendable to any fixed number of passes. If the number of passes changes along the iterations, the proof still applies if that number becomes fixed after a certain number of iterations. For setting this parameter, we found that a simple strategy was useful: for a blur of size  $(2b + 1) \times (2b + 1)$ , we set the number of passes to  $b$ . We found this strategy to yield a good choice of the number of

passes, irrespective of the noise level of the observed image. In what follows, we report the results obtained using this adaptive strategy (designated by ‘Proposed-AD’), and also the results obtained with just one BGS pass (designated by Proposed-1’); the latter are reported because they correspond to an especially simple, and therefore interesting situation.

In CM, in order to guarantee convergence, we chose its proximal parameters  $\tau$  and  $\sigma$  so as to satisfy the condition given in [29],  $\tau(\frac{\beta}{2} + \sigma\|\mathbf{D}^T\mathbf{D}\|_I) < 1$ , where  $\|\cdot\|_I$  denotes the induced norm of a matrix, and  $\beta$  is the Lipschitz constant of the gradient of  $f$ ,  $\beta = \|\mathbf{M}_y\mathbf{H}\|_I^2$ . It can be shown [35] that  $\|\mathbf{D}^T\mathbf{D}\|_I \leq 8$ ; we also have  $\|\mathbf{M}_y\mathbf{H}\|_I \leq \|\mathbf{M}_y\|_I\|\mathbf{H}\|_I$ , and  $\|\mathbf{M}_y\|_I = 1$ ,  $\|\mathbf{H}\|_I = 1$ . Therefore, in order to obey the convergence condition, we chose  $\tau = \frac{0.99}{0.5+8\sigma}$ . We verified experimentally that the speed of convergence was approximately maximal as long as  $\sigma \in [10^{-8}, 10^{-4}]$ , and therefore chose  $\sigma = 10^{-6}$ .

The variable  $\mathbf{z}$ , which exists only in the proposed method and in ADMM-CG, was initialized with an approximate solution of the quadratic regularization problem

$$\min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}, \mathbf{z}) + \frac{\tau}{2}\|\mathbf{x}\|^2,$$

where  $\tau$  is a small positive value. The approximate solution was computed by alternating minimization with respect to  $\mathbf{x}$  and  $\mathbf{z}$ , yielding the iterative procedure

$$\mathbf{x}^i = (\mathbf{H}^T\mathbf{H} + \tau\mathbf{I})^{-1} \mathbf{H}^T \begin{bmatrix} \mathbf{y} \\ \mathbf{z}^{i-1} \end{bmatrix} \quad (21)$$

$$\mathbf{z}^i = \mathbf{M}_z\mathbf{H}\mathbf{x}^i, \quad (22)$$

for  $i = 1, 2, \dots, 100$ ,  $\tau = 10^{-3}$ , with  $\mathbf{z}^0$  obtained by padding  $\mathbf{y}$  with its border elements. The complexity involved in the computation of  $\mathbf{x}^{i+1}, \mathbf{z}^{i+1}$  is dominated by the solution of the linear system implicit in (21). Given that  $(\mathbf{H}^T\mathbf{H} + \tau\mathbf{I})$  is BCCB, that solution can be efficiently found in the frequency domain, with complexity  $O[(k + d)\log(k + d)]$ .

To check whether the four methods converged to the same result, we ran them for a very large number of iterations ( $10^6$ ) in a few of the cases mentioned ahead, and found that, in each case, the results were essentially the same for all methods: the root-mean-square errors (RMSE)<sup>6</sup> among the results from different methods were always below  $10^{-7}$ , which is much below what is visually distinguishable. Given this, we arbitrarily chose, for all tests, the results of one of the methods (AM), after the mentioned  $10^6$  iterations, as representatives of the solutions of the corresponding problems. We used these representatives as reference images for the evaluation of the quality of the results of the four methods. Our choice of AM to compute the reference images did not especially benefit this method relative to the other ones, because, if we had chosen any of the other two methods instead of AM, the RMSE values of the results of the tests, computed relative to them, would have been essentially the same. We did not use the original sharp images as references, for two reasons. First, given the

<sup>6</sup>The RMSE was defined as

$$RMSE = \sqrt{1/(m' \times n') \sum_{i=1}^{m' \times n'} (x_i - x_i^r)^2},$$

where  $x_i$  and  $x_i^r$  denote, respectively, the pixels of the estimated image and of the reference image.

<sup>5</sup>For alternative strategies for setting the penalization parameters, see [11] and [34].

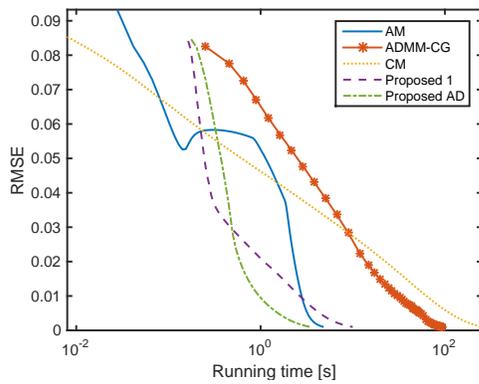


Fig. 4. Deblurring: RMSE of the estimated images as a function of running time, for the various methods. This test used the *cameraman* image with a  $13 \times 13$  boxcar blurring filter.



Fig. 5. Deblurring: observed (a) and estimated (b) images using the Proposed-AD method. The experiments were run for the *cameraman* image with a  $13 \times 13$  boxcar blur.

experiments mentioned in the beginning of this paragraph, we had good reasons to believe that the four methods converged to the same fixed point, in each problem. Second, the solutions of the optimization problems were slightly different from the original sharp images, due to the presence of the noise and of the regularizer, and our main interest was in assessing the speed of convergence of the methods to the solutions of the optimization problems, not in the recovery of the original images. If we had used the original images as references, it might happen that, in their path to convergence, some of the methods would pass closer to those images than other methods, and this could make them stop earlier. This would give them an apparently better speed, while they might not truly have a better convergence speed. In what follows, when we mention RMSE values, these were computed relative to the above-mentioned reference images, using the whole images, including the boundary zone, in the computation.

Fig. 4 illustrates the behavior of the four methods during the optimization. The estimated images from all the methods were visually indistinguishable from one another. Figure 5 shows a result of the Proposed-AD method. In Table I, we show the computing times for *cameraman* with Gaussian blurs of various sizes. The times for *Lena* with the same blurs, and for both images with the boxcar blurs, are presented in Appendix C. In summary, the Proposed-AD method was faster than AM for small and medium-sized boxcar blurs and for small Gaussian blurs, the two methods had similar speeds for large boxcar blurs and medium-sized Gaussian blurs, and AM was faster for large Gaussian blurs.

TABLE I  
RESULTS FOR *cameraman* WITH GAUSSIAN BLURS OF VARIOUS SIZES. THE VALUE OF  $\kappa$  IS THE CONDITION NUMBER OF MATRIX  $\mathbf{H}$ . THE TWO RIGHTMOST COLUMNS SHOW THE NUMBER OF ITERATIONS AND THE TIME TAKEN TO REACH AN RMSE BELOW  $10^{-3}$ .

Method	Blur size	$\kappa \times 10^3$	Iterations	Time (s)
Proposed-1	5	247.0	85	0.866
Proposed-AD			56	0.770
AM			178	1.973
ADMM-CG			45	29.355
CM			14562	63.797
Proposed-1	13	21590.6	531	4.590
Proposed-AD			117	2.687
AM			313	3.399
ADMM-CG			71	106.450
CM			123912	539.466
Proposed-1	21	686384.8	5366	45.169
Proposed-AD			542	17.858
AM			882	9.478
ADMM-CG			319	223.828
CM			305227	1334.286

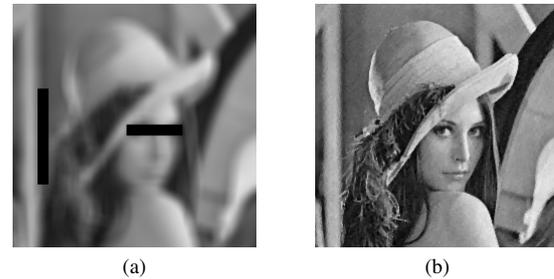


Fig. 6. Inpainting with deblurring: observed (a) and estimated (b) image using the Proposed-AD method. In the observed image, the black rectangles correspond to unobserved regions.

### C. Proposed method: inpainting, superresolution, and demosaicing

The experiments reported in this subsection used the settings that were described in the previous subsection, unless otherwise noted. In all experiments, the boundary zone was considered as unobserved, as in the previous subsection.

a) *Inpainting*: As mentioned in Section II, the inpainting problem can be formulated within the proposed framework by simply including in  $\mathbf{z}$  the unobserved pixels. We performed tests on a problem of simultaneous deblurring and inpainting, using the *Lena* image with a boxcar blur of size  $13 \times 13$ , in which some zones were unobserved, as shown in Fig. 6 (a). The regularization parameter was kept at  $\lambda = 2 \times 10^{-6}$ . The result obtained with the Proposed-AD method can be seen in Fig. 6 (b). The other methods produced visually indistinguishable results. The evolution of the different methods during the optimization process is shown in Fig. 7.

b) *Superresolution*: In superresolution problems,  $\mathbf{z}$  should include the extra pixels that would have been present if the image had the higher resolution. For this test, we used a  $608 \times 337$  crop of a single band of the so-called *Pavia* hyperspectral image. The experiments consisted of increasing the resolution by a factor of 3, and we used a  $3 \times 3$  boxcar blurring filter in the estimation of the high-resolution image. The regularization parameter was set, again, to  $\lambda = 2 \times 10^{-4}$ . Figs. 8 and 9 show the results. The estimated images from all

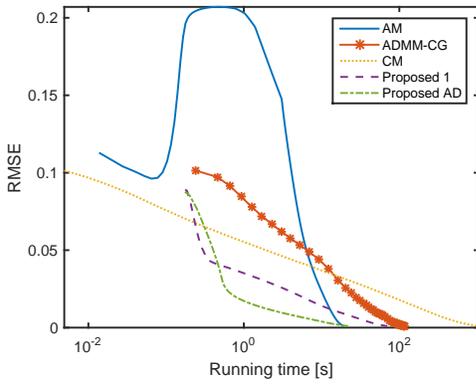


Fig. 7. Inpainting with deblurring: RMSE of the estimated images as a function of running time, for the various methods.

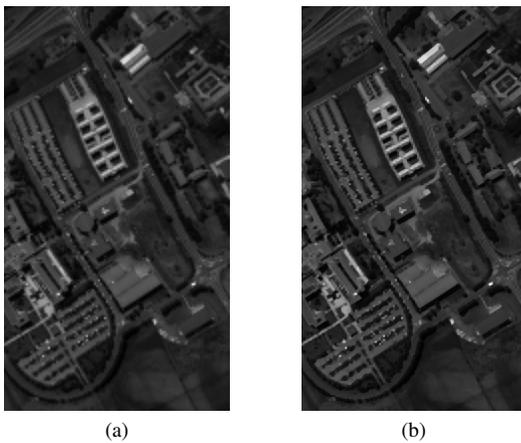


Fig. 8. Superresolution: observed (a) and estimated (b) images using the Proposed-AD method. The upsampling ratio was 3.

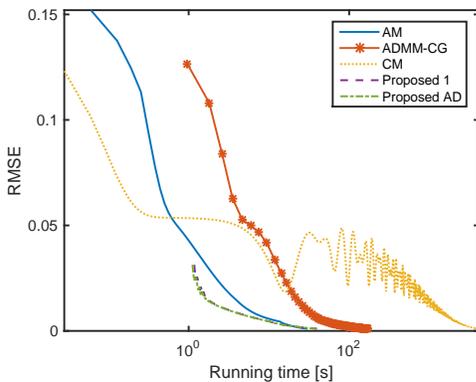


Fig. 9. Superresolution: RMSE of the estimated images as a function of running time, for the various methods.

the methods were visually indistinguishable from one another.

*c) Demosaicing:* The demosaicing operation, which is a step of the processing pipeline used in most digital cameras, seeks to reconstruct a full-resolution color image from its subsampled RGB channels. In most digital cameras, the image sensor observes only one color in each pixel, and therefore the color channels are subsampled relative to the resolution of the sensor. The goal of demosaicing is to fuse

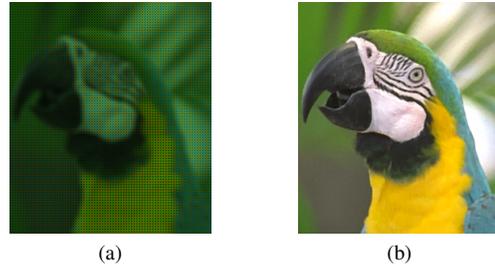


Fig. 10. Demosaicing: observed (a) and estimated (b) image using the Proposed-AD method.

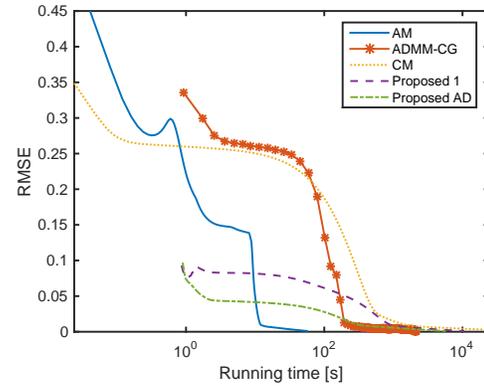


Fig. 11. Demosaicing: RMSE of the estimated images as a function of running time, for the various methods.

the color channels, obtaining full-resolution images for all of them. We performed tests on a problem of simultaneous deblurring and demosaicing. We used a  $289 \times 253$  crop of the well-known *parrot* image as ground truth, and synthesized the observed image by first blurring it with an  $8 \times 8$  boxcar filter, and then applying a Bayer color filter. In the representation of the problem within our framework, vector  $\mathbf{y}$  contained the observed pixels from the three color channels, and vector  $\mathbf{z}$  contained the unobserved pixels from all of them (including the boundary zone). In the vector  $\mathbf{y}$ , the observed pixels were grouped according to the channels they belonged to, by making  $\mathbf{y} = [(\mathbf{y}^1)^T (\mathbf{y}^2)^T (\mathbf{y}^3)^T]^T$ ; the same was done for the vectors  $\mathbf{z}$  and  $\mathbf{x}$ . The data-fitting term had the form  $f(\mathbf{u}) = 1/2 \sum_{j=1}^3 \|\begin{bmatrix} \mathbf{y}^j \\ \mathbf{z}^j \end{bmatrix} - \mathbf{H}\mathbf{x}^j\|^2$ . As in the demosaicing example of [29], we used a vectorial version of the TV regularizer, which promotes the alignment of the discontinuities across channels:  $\phi(\mathbf{D}\mathbf{x}) = \lambda \sum_{i=1}^{k+d} \sqrt{\sum_{j=1}^3 \{(\mathbf{D}_h \mathbf{x}^j)_i^2 + (\mathbf{D}_v \mathbf{x}^j)_i^2\}}$ . The regularization parameter was set to  $\lambda = 10^{-5}$ . The results can be seen in Figs. 10 and 11. Once again, the estimated images from all the methods were visually indistinguishable from one another.

#### D. Appraisal

The example of the use of IDD-BM3D within the proposed framework illustrates the fact that this framework can be used to convert existing deblurring methods that assume artificial boundary conditions to methods that use unknown boundaries. As far as we know, there is no other published way to accomplish this. This is a simple alternative to the use of

edge tapering, yielding results of better quality, as illustrated in Subsection III-A.

The proposed partial ADMM implementation of Framework 1 (more specifically, the Proposed-AD form) showed a performance similar to the performance of the state-of-the-art AM method, both in terms of final results and of computational speed. CM yielded similar deblurred images, but showed a significantly lower computational performance. We conjecture that this was due to the fact that both the proposed method and AM, being based on ADMM, make use of second-order information on the objective function (in terms of the matrix  $\mathbf{H}^T\mathbf{H} + \mu\mathbf{D}^T\mathbf{D}$ ). CM, on the other hand, does not use any information of this kind. This difference in speed agrees with our experience in other image enhancement problems, in which we have repeatedly found CM-like methods to be significantly slower than ADMM-based ones.

#### IV. CONCLUSION

We have proposed a framework for deblurring images with unobserved pixels. This framework can be used to convert most deblurring methods to unknown boundaries, irrespective of the specific boundary conditions that those methods assume, being a simple, high-quality alternative to the use of edge tapering.

An ADMM-based deblurring method that falls within the mentioned framework was proposed, and a proof of convergence was provided. This method can be seen as a partial ADMM with a non-dualized variable. Experimental results on problems of deconvolution, inpainting, superresolution and demosaicing, with unknown boundaries, showed, for the proposed method, a performance at the level of the state of the art.

#### APPENDIX A ADMM

ADMM can be used to solve convex optimization problems of the form

$$\underset{\mathbf{u}}{\text{minimize}} \quad f(\mathbf{u}) + \psi(\mathbf{K}\mathbf{u}), \quad (23)$$

where  $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  and  $\psi: \mathbb{R}^m \rightarrow \mathbb{R} \cup \{+\infty\}$  are closed proper convex functions, and  $\mathbf{K} \in \mathbb{R}^{m \times n}$ . The problem is first rewritten as

$$\begin{aligned} & \underset{\mathbf{u}, \mathbf{v}}{\text{minimize}} \quad f(\mathbf{u}) + \psi(\mathbf{v}) \\ & \text{subject to} \quad \mathbf{v} = \mathbf{K}\mathbf{u}, \end{aligned} \quad (24)$$

where  $\mathbf{v} \in \mathbb{R}^m$ . This problem has the augmented Lagrangian [36]

$$\mathcal{L}(\mathbf{u}, \mathbf{v}, \mathbf{d}) = f(\mathbf{u}) + \psi(\mathbf{v}) + \frac{\mu}{2} \|\mathbf{v} - \mathbf{K}\mathbf{u} - \mathbf{d}\|^2, \quad (25)$$

where  $\mu > 0$  is a penalization parameter, and  $\mathbf{d} \in \mathbb{R}^m$  is the so-called scaled dual variable ( $\mu\mathbf{d}$  is the dual variable).

The standard ADMM solves problem (23) through an iteration on a set of problems that are often much simpler,

$$\mathbf{u}^{i+1} = \underset{\mathbf{u}}{\text{arg min}} \quad \mathcal{L}(\mathbf{u}, \mathbf{v}^i, \mathbf{d}^i), \quad (26)$$

$$\mathbf{v}^{i+1} = \underset{\mathbf{v}}{\text{arg min}} \quad \mathcal{L}(\mathbf{u}^{i+1}, \mathbf{v}, \mathbf{d}^i), \quad (27)$$

$$\mathbf{d}^{i+1} = \mathbf{d}^i - (\mathbf{v}^{i+1} - \mathbf{K}\mathbf{u}^{i+1}). \quad (28)$$

The convergence of ADMM was established in [37] (among others) by proving a more general version of the following theorem:

**Theorem 2.** *Consider a problem of the form (23), in which  $\mathbf{K}$  has full column rank. Let  $\mathbf{d}^0 \in \mathbb{R}^m$  and  $\mu > 0$ . Assume that  $\{\mathbf{u}^i\}$ ,  $\{\mathbf{v}^i\}$ , and  $\{\mathbf{d}^i\}$  conform, for all  $i$ , to*

$$\mathbf{u}^{i+1} = \underset{\mathbf{u}}{\text{arg min}} \quad \mathcal{L}(\mathbf{u}, \mathbf{v}^i, \mathbf{d}^i), \quad (29)$$

$$\mathbf{v}^{i+1} = \underset{\mathbf{v}}{\text{arg min}} \quad \mathcal{L}(\mathbf{u}^{i+1}, \mathbf{v}, \mathbf{d}^i), \quad (30)$$

$$\mathbf{d}^{i+1} = \mathbf{d}^i - (\mathbf{v}^{i+1} - \mathbf{K}\mathbf{u}^{i+1}). \quad (31)$$

*If problem (23) has a Karush-Kuhn-Tucker (KKT) pair,<sup>7</sup> then the sequence  $\{\mathbf{u}^i\}$  converges to a solution of that problem,  $\{\mu\mathbf{d}^i\}$  converges to a solution of the dual problem, and  $\{\mathbf{v}^i\}$  converges to  $\mathbf{K}\mathbf{u}^*$ , where  $\mathbf{u}^*$  is the limit of  $\{\mathbf{u}^i\}$ . If problem (23) has no KKT pair, then at least one of the sequences  $\{\mathbf{d}^i\}$  or  $\{\mathbf{v}^i\}$  is unbounded.*

#### APPENDIX B PROOF OF THEOREM 1

The proof follows a similar machinery to the ones proposed in [38], [39], [40], [41], [42]. We start by deriving some preliminary results. Consider problem (6) with  $f$  given by (16), and define, as in the main text,

$$\mathbf{u} = \begin{bmatrix} \mathbf{x} \\ \mathbf{z} \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_d \end{bmatrix}, \quad \psi(\mathbf{K}\mathbf{u}) = \phi(\mathbf{D}\mathbf{x}).$$

Also define

$$\boldsymbol{\omega} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{d} \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} \mathbf{z} \\ \mathbf{v}_x \\ \mathbf{d}_x \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{C} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mu\mathbf{I}_l & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mu\mathbf{I}_l \end{bmatrix},$$

$$\boldsymbol{\Pi} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_d & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ (\mathbf{D}^T\mathbf{D})^{-1}\mathbf{D}^T & \mathbf{0} & \mathbf{I}_l & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_l & \mathbf{0} \end{bmatrix}^T, \quad \boldsymbol{\Gamma} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_d & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_l & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_l \end{bmatrix}.$$

Note that  $\mathbf{w} = \boldsymbol{\Gamma}\boldsymbol{\omega}$ , and therefore  $\boldsymbol{\omega}$  univocally determines  $\mathbf{w}$ .

Lemmas 1–5 present several results concerning the partial ADMM (12)–(15) with  $f$  defined as in (16).

**Lemma 1.** *Assume that  $\phi$  is closed proper convex and coercive and that  $\mathbf{D}$  is full column rank. Then, the set of solutions of problem (6) is non-empty.*

*Proof.* This result follows immediately from [43, Proposition 11.14].  $\square$

**Lemma 2.** *Under the assumptions of Lemma 1, let  $\mathbf{w}^i = [(\mathbf{z}^i)^T (\mathbf{v}_x^i)^T (\mathbf{d}_x^i)^T]^T$  denote the vector  $\mathbf{w}$  that results from the  $i$ th iteration of the partial ADMM (12)–(15), obtained with a given initialization of  $\mathbf{x}$ ,  $\mathbf{z}$ ,  $\mathbf{v}_x$ , and  $\mathbf{d}_x$ . Denote with an asterisk, as in  $\boldsymbol{\omega}^* = [(\mathbf{u}^*)^T (\mathbf{v}^*)^T (\mathbf{d}^*)^T]^T$ , a KKT pair of problem (6).<sup>8</sup> Let  $\mathbf{w}^* = \boldsymbol{\Gamma}\boldsymbol{\omega}^*$ . Then, the following inequality holds:*

$$\|\mathbf{w}^i - \mathbf{w}^*\|_{\mathbf{G}}^2 - \|\mathbf{w}^{i+1} - \mathbf{w}^*\|_{\mathbf{G}}^2 \geq \|\mathbf{w}^{i+1} - \mathbf{w}^i\|_{\mathbf{G}}^2. \quad (32)$$

<sup>7</sup>A pair  $(\mathbf{u}, \mathbf{p})$  is said to be a Karush-Kuhn-Tucker pair of problem (23) if  $-\mathbf{K}^T\mathbf{p} \in \partial f(\mathbf{u})$  and  $\mathbf{p} \in \partial\psi(\mathbf{K}\mathbf{u})$ .

<sup>8</sup>We consider problem (6) reformulated as (7), and say that a pair  $(\mathbf{u}, \mathbf{p})$  is a Karush-Kuhn-Tucker pair of that problem if  $-\mathbf{K}^T\mathbf{p} \in \partial f(\mathbf{u})$  and  $\mathbf{p} \in \partial\psi(\mathbf{K}\mathbf{u})$ . These are necessary and sufficient conditions for  $\mathbf{u}$  to be a solution of this problem [43, Proposition 26.11].

*Proof.* We will give a proof for the case in which only one block-Gauss-Seidel pass is used, per iteration, to compute  $\mathbf{x}$  and  $\mathbf{z}$  in the partial ADMM. The proof is easily extendable to more than one pass per iteration.

The necessary and sufficient conditions for the existence of  $\mathbf{u}^*$ ,  $\mathbf{v}^*$  and  $\mathbf{d}^*$  are the following: primal feasibility,

$$\mathbf{v}^* = \mathbf{K}\mathbf{u}^* \Leftrightarrow \mathbf{v}_x^* = \mathbf{D}\mathbf{x}^*, \mathbf{v}_z^* = \mathbf{z}^*, \quad (33)$$

and dual feasibility,

$$-\mu\mathbf{D}^T\mathbf{d}_x^* = \nabla_{\mathbf{x}}f(\mathbf{x}^*, \mathbf{z}^*), \quad (34)$$

$$-\mu\mathbf{d}_z^* = \nabla_{\mathbf{z}}f(\mathbf{x}^*, \mathbf{z}^*), \quad (35)$$

$$\begin{aligned} \mu\mathbf{d}_x^* &\in \partial_{\mathbf{v}_x}\psi(\mathbf{v}^*) \\ \Leftrightarrow \mu\mathbf{d}_x^* &\in \partial_{\mathbf{v}_x}\phi(\mathbf{v}^*) \end{aligned} \quad (36)$$

$$\mathbf{d}_z^* = \mathbf{0}. \quad (37)$$

Eqs. (36) and (37) use the fact that  $\psi(\mathbf{v})$  doesn't depend on  $\mathbf{v}_z$ , and the latter equation also uses the fact that  $\mu \neq 0$ .

The solutions of the minimization problems of the partial ADMM obey, respectively,

$$\begin{aligned} \nabla_{\mathbf{x}}f(\mathbf{x}^{i+1}, \mathbf{z}^i) &= \mu\mathbf{D}^T(\mathbf{v}_x^i - \mathbf{D}\mathbf{x}^{i+1} - \mathbf{d}_x^i) \\ &= \mu\mathbf{D}^T(\mathbf{v}_x^i - \mathbf{v}_x^{i+1}) - \mu\mathbf{D}^T\mathbf{d}_x^{i+1}, \end{aligned} \quad (38)$$

$$\nabla_{\mathbf{z}}f(\mathbf{x}^{i+1}, \mathbf{z}^{i+1}) = \mathbf{0}, \quad (39)$$

$$\begin{aligned} \mathbf{0} &\in \partial_{\mathbf{v}_x}\phi(\mathbf{v}_x^{i+1}) + \mu(\mathbf{v}_x^{i+1} - \mathbf{D}\mathbf{x}^{i+1} - \mathbf{d}_x^i) \\ \Leftrightarrow \mu\mathbf{d}_x^{i+1} &\in \partial_{\mathbf{v}_x}\phi(\mathbf{v}_x^{i+1}), \end{aligned} \quad (40)$$

where we have used (15) in (38) and (40). Since  $\phi$  is convex,  $\partial\phi$  is monotone, from Kachurovskii's theorem. Denote the vectors  $\mathbf{v}_x$  and  $\mathbf{d}_x$  from two different iterations of the partial ADMM by  $\mathbf{v}_x^i$  and  $\mathbf{d}_x^i$ , and  $\mathbf{v}_x''$  and  $\mathbf{d}_x''$ , respectively. Due to the monotonicity of  $\partial\phi$  and to (40), they will satisfy the condition

$$(\mathbf{v}_x' - \mathbf{v}_x'')^T(\mathbf{d}_x' - \mathbf{d}_x'') \geq 0. \quad (41)$$

Also due to that monotonicity, now together with (40) and (36), we have

$$(\mathbf{v}_x' - \mathbf{v}_x^*)^T(\mathbf{d}_x' - \mathbf{d}_x^*) \geq 0. \quad (42)$$

Computing the gradients in Eqs. (34) and (38), we obtain, respectively,

$$\mathbf{A}\mathbf{x}^* + \mathbf{B}\mathbf{z}^* + \mathbf{e} + \mu\mathbf{D}^T\mathbf{d}_x^* = \mathbf{0}, \quad (43)$$

$$\mathbf{A}\mathbf{x}^{i+1} + \mathbf{B}\mathbf{z}^i + \mathbf{e} + \mu\mathbf{D}^T\mathbf{d}_x^{i+1} - \mu\mathbf{D}^T(\mathbf{v}_x^i - \mathbf{v}_x^{i+1}) = \mathbf{0}, \quad (44)$$

and for equations (35) and (39) we obtain, respectively,

$$\mathbf{B}^T\mathbf{x}^* + \mathbf{C}\mathbf{z}^* + \mathbf{f} = \mathbf{0} \Leftrightarrow \mathbf{z}^* = -\mathbf{C}^{-1}(\mathbf{B}^T\mathbf{x}^* + \mathbf{f}), \quad (45)$$

$$\mathbf{z}^{i+1} = -\mathbf{C}^{-1}(\mathbf{B}^T\mathbf{x}^{i+1} + \mathbf{f}); \quad (46)$$

in (45) we have used (37).

Subtracting (43) from (44) yields

$$\begin{aligned} \mathbf{A}(\mathbf{x}^{i+1} - \mathbf{x}^*) + \mathbf{B}(\mathbf{z}^i - \mathbf{z}^*) \\ - \mu\mathbf{D}^T(\mathbf{v}_x^i - \mathbf{v}_x^{i+1}) + \mu\mathbf{D}^T(\mathbf{d}_x^{i+1} - \mathbf{d}_x^*) = \mathbf{0}. \end{aligned} \quad (47)$$

Using the equality  $\mathbf{B}(\mathbf{z}^i - \mathbf{z}^*) = \mathbf{B}(\mathbf{z}^i - \mathbf{z}^{i+1}) + \mathbf{B}(\mathbf{z}^{i+1} - \mathbf{z}^*)$  and Eqs. (45) and (46), we can rewrite (47) as

$$\begin{aligned} (\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T)(\mathbf{x}^{i+1} - \mathbf{x}^*) + \mathbf{B}(\mathbf{z}^i - \mathbf{z}^{i+1}) \\ - \mu\mathbf{D}^T(\mathbf{v}_x^i - \mathbf{v}_x^{i+1}) + \mu\mathbf{D}^T(\mathbf{d}_x^{i+1} - \mathbf{d}_x^*) = \mathbf{0}. \end{aligned} \quad (48)$$

The matrix  $(\mathbf{A} - \mathbf{B}\mathbf{C}^{-1}\mathbf{B}^T)$  is PSD, as noted in Section II-B. Due to this fact, multiplying both sides of (48), on the left, by  $(\mathbf{x}^{i+1} - \mathbf{x}^*)^T$ , leads to

$$\begin{aligned} 0 \leq -(\mathbf{x}^{i+1} - \mathbf{x}^*)^T\mathbf{B}(\mathbf{z}^i - \mathbf{z}^{i+1}) \\ + \mu(\mathbf{x}^{i+1} - \mathbf{x}^*)^T\mathbf{D}^T(\mathbf{v}_x^i - \mathbf{v}_x^{i+1}) \\ - \mu(\mathbf{x}^{i+1} - \mathbf{x}^*)^T\mathbf{D}^T(\mathbf{d}_x^{i+1} - \mathbf{d}_x^*). \end{aligned} \quad (49)$$

Using Eqs. (15) and (33), we can transform the last two members of the right-hand side of (49) as follows:

$$\begin{aligned} \mu(\mathbf{x}^{i+1} - \mathbf{x}^*)^T\mathbf{D}^T(\mathbf{v}_x^i - \mathbf{v}_x^{i+1}) \\ - \mu(\mathbf{x}^{i+1} - \mathbf{x}^*)^T\mathbf{D}^T(\mathbf{d}_x^{i+1} - \mathbf{d}_x^*) \\ = \mu(\mathbf{v}_x^i - \mathbf{v}_x^{i+1})^T(\mathbf{D}\mathbf{x}^{i+1} - \mathbf{D}\mathbf{x}^*) \\ - \mu(\mathbf{d}_x^{i+1} - \mathbf{d}_x^*)^T(\mathbf{D}\mathbf{x}^{i+1} - \mathbf{D}\mathbf{x}^*) \\ = \mu(\mathbf{v}_x^i - \mathbf{v}_x^{i+1})^T(\mathbf{v}_x^{i+1} - \mathbf{v}_x^*) \\ - \mu(\mathbf{v}_x^i - \mathbf{v}_x^{i+1})^T(\mathbf{d}_x^i - \mathbf{d}_x^{i+1}) \\ - \mu(\mathbf{d}_x^{i+1} - \mathbf{d}_x^*)^T(\mathbf{v}_x^{i+1} - \mathbf{v}_x^*) \\ + \mu(\mathbf{d}_x^{i+1} - \mathbf{d}_x^*)^T(\mathbf{d}_x^i - \mathbf{d}_x^{i+1}). \end{aligned} \quad (50)$$

Subtracting (45) from (46) and multiplying both sides of the result, on the right, by  $\mathbf{C}(\mathbf{z}^i - \mathbf{z}^{i+1})$ , we obtain

$$(\mathbf{z}^{i+1} - \mathbf{z}^*)^T\mathbf{C}(\mathbf{z}^i - \mathbf{z}^{i+1}) = -(\mathbf{x}^{i+1} - \mathbf{x}^*)^T\mathbf{B}(\mathbf{z}^i - \mathbf{z}^{i+1}). \quad (51)$$

Equations (50), (51), (41) and (42), applied to (49), lead to

$$\begin{aligned} 0 \leq (\mathbf{z}^{i+1} - \mathbf{z}^*)^T\mathbf{C}(\mathbf{z}^i - \mathbf{z}^{i+1}) \\ + \mu(\mathbf{v}_x^{i+1} - \mathbf{v}_x^*)^T(\mathbf{v}_x^i - \mathbf{v}_x^{i+1}) \\ + \mu(\mathbf{d}_x^{i+1} - \mathbf{d}_x^*)^T(\mathbf{d}_x^i - \mathbf{d}_x^{i+1}) \\ = (\mathbf{w}^{i+1} - \mathbf{w}^*)^T\mathbf{G}(\mathbf{w}^i - \mathbf{w}^{i+1}). \end{aligned} \quad (52)$$

$\mathbf{G}$  is PD, since it is a block-diagonal matrix composed of PD blocks. We have

$$\begin{aligned} \|\mathbf{w}^{i+1} - \mathbf{w}^*\|_{\mathbf{G}}^2 &= \|(\mathbf{w}^i - \mathbf{w}^*) - (\mathbf{w}^i - \mathbf{w}^{i+1})\|_{\mathbf{G}}^2 \\ &= \|\mathbf{w}^i - \mathbf{w}^*\|_{\mathbf{G}}^2 + \|\mathbf{w}^i - \mathbf{w}^{i+1}\|_{\mathbf{G}}^2 \\ &\quad - 2(\mathbf{w}^i - \mathbf{w}^*)^T\mathbf{G}(\mathbf{w}^i - \mathbf{w}^{i+1}). \end{aligned} \quad (53)$$

Reordering these terms, we obtain

$$\begin{aligned} \|\mathbf{w}^i - \mathbf{w}^*\|_{\mathbf{G}}^2 - \|\mathbf{w}^{i+1} - \mathbf{w}^*\|_{\mathbf{G}}^2 + \|\mathbf{w}^i - \mathbf{w}^{i+1}\|_{\mathbf{G}}^2 \\ = 2(\mathbf{w}^i - \mathbf{w}^*)^T\mathbf{G}(\mathbf{w}^i - \mathbf{w}^{i+1}). \end{aligned} \quad (54)$$

From (52),

$$\begin{aligned} 0 \leq (\mathbf{w}^{i+1} - \mathbf{w}^i + \mathbf{w}^i - \mathbf{w}^*)^T\mathbf{G}(\mathbf{w}^i - \mathbf{w}^{i+1}) \\ \Leftrightarrow 0 \leq -\|\mathbf{w}^{i+1} - \mathbf{w}^i\|_{\mathbf{G}}^2 \\ + (\mathbf{w}^i - \mathbf{w}^*)^T\mathbf{G}(\mathbf{w}^i - \mathbf{w}^{i+1}) \\ \Leftrightarrow (\mathbf{w}^i - \mathbf{w}^*)^T\mathbf{G}(\mathbf{w}^i - \mathbf{w}^{i+1}) \geq \|\mathbf{w}^i - \mathbf{w}^{i+1}\|_{\mathbf{G}}^2. \end{aligned} \quad (55)$$

Finally, (54) and (55) lead to

$$\|\mathbf{w}^i - \mathbf{w}^*\|_{\mathbf{G}}^2 - \|\mathbf{w}^{i+1} - \mathbf{w}^*\|_{\mathbf{G}}^2 \geq \|\mathbf{w}^i - \mathbf{w}^{i+1}\|_{\mathbf{G}}^2. \quad \square$$

**Lemma 3.** *Under the assumptions of Lemma 1,*

- 1) *The sequence  $\{\|\mathbf{w}^i - \mathbf{w}^*\|_{\mathbf{G}}^2\}$  is convergent.*
- 2) *The sequence  $\{\|\mathbf{w}^i - \mathbf{w}^{i+1}\|_{\mathbf{G}}^2\}$  is non-increasing and converges to 0.*
- 3) *The sequence  $\{\mathbf{w}^i\}$  has a convergent subsequence.*
- 4) *Let  $\boldsymbol{\omega}^i = \mathbf{\Pi}\mathbf{w}^i$ ; the sequence  $\{\boldsymbol{\omega}^i\}$  has a convergent subsequence.*

*Proof.* From inequality (32), we can conclude that  $\|\mathbf{w}^{i+1} - \mathbf{w}^*\|_{\mathbf{G}}^2 \leq \|\mathbf{w}^i - \mathbf{w}^*\|_{\mathbf{G}}^2$ . This means that the sequence  $\{\|\mathbf{w}^i - \mathbf{w}^*\|_{\mathbf{G}}^2\}$  is non-increasing. Since its elements are non-negative, the sequence is bounded. Since the sequence is non-increasing and bounded, it is convergent. Taking the limits of both sides of (32), we conclude that  $\|\mathbf{w}^i - \mathbf{w}^{i+1}\|_{\mathbf{G}}^2 \rightarrow 0$ .

Since the sequence  $\{\|\mathbf{w}^i - \mathbf{w}^*\|_{\mathbf{G}}^2\}$  is bounded,  $\{\mathbf{w}^i\}$  is bounded as well. Therefore, it has a convergent subsequence. The same applies to  $\{\boldsymbol{\omega}^i\}$ .  $\square$

**Lemma 4.** *Under the assumptions of Lemma 1, the limit of any convergent subsequence of  $\{\boldsymbol{\omega}^i\}$  is a KKT pair of problem (6).*

*Proof.* From Lemma 3,  $\|\mathbf{w}^{i+1} - \mathbf{w}^i\|_{\mathbf{G}}^2 \rightarrow 0$ . This implies that  $\|\mathbf{z}^i - \mathbf{z}^{i+1}\|_{\mathbf{C}}^2 \rightarrow 0$ ,  $\|\mathbf{v}_x^i - \mathbf{v}_x^{i+1}\|^2 \rightarrow 0$ , and  $\|\mathbf{d}_x^i - \mathbf{d}_x^{i+1}\|^2 \rightarrow 0$ . From the latter and from (15), we have

$$\mathbf{v}_x^{i+1} - \mathbf{D}\mathbf{x}^{i+1} \rightarrow 0. \quad (56)$$

From Lemma 3,  $\{\mathbf{w}^i\}$  has a convergent subsequence. Consider one such subsequence,  $\{\mathbf{w}^{i_j}\}$ , whose limit we designate by  $\mathbf{w}^\infty$ . Denote by  $\mathbf{z}^\infty$ ,  $\mathbf{v}_x^\infty$ , and  $\mathbf{d}_x^\infty$ , respectively, the limits of the corresponding subsequences  $\{\mathbf{z}^{i_j}\}$ ,  $\{\mathbf{v}_x^{i_j}\}$ , and  $\{\mathbf{d}_x^{i_j}\}$ . Additionally, designate by  $\boldsymbol{\omega}^\infty$  the limit of the subsequence  $\{\boldsymbol{\omega}^{i_j}\} = \{\mathbf{\Pi}\mathbf{w}^{i_j}\}$ . Note that  $\mathbf{z}^{i_j+1} \rightarrow \mathbf{z}^\infty$  because, according to Lemma 2,  $\|\mathbf{z}^i - \mathbf{z}^{i+1}\|_{\mathbf{C}} \rightarrow 0$ ; for similar reasons,  $\mathbf{v}_x^{i_j+1} \rightarrow \mathbf{v}_x^\infty$  and  $\mathbf{d}_x^{i_j+1} \rightarrow \mathbf{d}_x^\infty$ . Since  $\{\mathbf{v}_x^{i_j}\}$  converges and  $\mathbf{D}$  is full column rank, (56) shows that  $\{\mathbf{x}^{i_j}\}$  also converges. Denote its limit by  $\mathbf{x}^\infty$ . Taking the limits of both sides of (56) over the subsequence of indexes  $\{i_j\}$ , we obtain

$$\mathbf{v}_x^\infty = \mathbf{D}\mathbf{x}^\infty. \quad (57)$$

Taking the limits on both sides of Eqs. (38)–(40), again over the subsequence of indexes  $\{i_j\}$ , and applying [44, Theorem 24.4], we have

$$-\mu\mathbf{D}^T\mathbf{d}_x^\infty = \nabla_{\mathbf{x}}f(\mathbf{x}^\infty, \mathbf{z}^\infty), \quad (58)$$

$$\mathbf{0} = \nabla_{\mathbf{z}}f(\mathbf{x}^\infty, \mathbf{z}^\infty), \quad (59)$$

$$\mu\mathbf{d}_x^\infty \in \partial_{\mathbf{v}_x}\psi(\mathbf{v}_x^\infty). \quad (60)$$

Variables  $\mathbf{v}_z^i$  and  $\mathbf{d}_z^i$  are not used in the partial ADMM, and therefore we can give them any values that are convenient. By making  $\boldsymbol{\omega}^i = \mathbf{\Pi}\mathbf{w}^i$ , we are setting, for all  $i$ ,  $\mathbf{v}_z^i = \mathbf{z}^i$  and  $\mathbf{d}_z^i = \mathbf{0}$ . This implies that  $\mathbf{v}_z^\infty = \mathbf{z}^\infty$  and  $\mathbf{d}_z^\infty = \mathbf{0}$ . These two equalities, together with (57)–(60), correspond to the optimality conditions of problem (6), i.e.,  $\boldsymbol{\omega}^\infty$  is a KKT pair of this problem.  $\square$

**Lemma 5.** *Under the assumptions of Lemma 1, the sequence  $\{\boldsymbol{\omega}^i\}$  is convergent.*

*Proof.* We follow the argument made in, e.g., [45], [38]. Consider the limits of two convergent subsequences of  $\{\boldsymbol{\omega}^i\}$ ,

$\boldsymbol{\omega}_1^\infty$  and  $\boldsymbol{\omega}_2^\infty$ . According to Lemma 4,  $\boldsymbol{\omega}_1^\infty$  and  $\boldsymbol{\omega}_2^\infty$  are both KKT pairs of problem (7). Therefore, according to Lemma 3, the following two limits exist:

$$\lim_{i \rightarrow \infty} \|\boldsymbol{\omega}^i - \boldsymbol{\omega}_p^\infty\|_{\mathbf{G}} = a_p, \quad p = 1, 2. \quad (61)$$

We have

$$\begin{aligned} & \|\boldsymbol{\omega}^i - \boldsymbol{\omega}_1^\infty\|_{\mathbf{G}}^2 - \|\boldsymbol{\omega}^i - \boldsymbol{\omega}_2^\infty\|_{\mathbf{G}}^2 \\ &= -2(\boldsymbol{\omega}^i)^T(\boldsymbol{\omega}_1^\infty - \boldsymbol{\omega}_2^\infty) + \|\boldsymbol{\omega}_1^\infty\|_{\mathbf{G}}^2 - \|\boldsymbol{\omega}_2^\infty\|_{\mathbf{G}}^2. \end{aligned}$$

Taking the limits of both sides over a subsequence of  $\{\boldsymbol{\omega}^i\}$  that converges to  $\boldsymbol{\omega}_1^\infty$ , we have

$$\begin{aligned} a_1^2 - a_2^2 &= -2(\boldsymbol{\omega}_1^\infty)^T(\boldsymbol{\omega}_1^\infty - \boldsymbol{\omega}_2^\infty) + \|\boldsymbol{\omega}_1^\infty\|_{\mathbf{G}}^2 - \|\boldsymbol{\omega}_2^\infty\|_{\mathbf{G}}^2 \\ &= -\|\boldsymbol{\omega}_1^\infty - \boldsymbol{\omega}_2^\infty\|_{\mathbf{G}}^2, \end{aligned}$$

and taking the limits over a subsequence that converges to  $\boldsymbol{\omega}_2^\infty$ , we have

$$\begin{aligned} a_1^2 - a_2^2 &= -2(\boldsymbol{\omega}_2^\infty)^T(\boldsymbol{\omega}_1^\infty - \boldsymbol{\omega}_2^\infty) + \|\boldsymbol{\omega}_1^\infty\|_{\mathbf{G}}^2 - \|\boldsymbol{\omega}_2^\infty\|_{\mathbf{G}}^2 \\ &= \|\boldsymbol{\omega}_1^\infty - \boldsymbol{\omega}_2^\infty\|_{\mathbf{G}}^2. \end{aligned}$$

These two equalities imply that  $\|\boldsymbol{\omega}_1^\infty - \boldsymbol{\omega}_2^\infty\|_{\mathbf{G}}^2 = 0$ . This means that  $\boldsymbol{\omega}_1^\infty = \boldsymbol{\omega}_2^\infty$ . Therefore, all convergent subsequences of  $\{\boldsymbol{\omega}^i\}$  converge to the same limit, and as a consequence  $\{\boldsymbol{\omega}^i\}$  is convergent.  $\square$

After these preliminary results, we prove Theorem 1.

*Proof of Theorem 1.* With problem (6) reformulated as (7), we are under the conditions of Lemmas 1–5. Lemma 1 shows that the set of solutions of the problem is non-empty. Lemma 5 allows us to conclude that the sequence  $\{\boldsymbol{\omega}^i\}$  converges. Lemma 4 shows that its limit corresponds to a KKT pair. This implies that  $\left\{\begin{bmatrix} \mathbf{x}^i \\ \mathbf{z}^i \end{bmatrix}\right\}$  converges to a solution of problem (6),  $\begin{bmatrix} \mathbf{x}^* \\ \mathbf{z}^* \end{bmatrix}$ , and that  $\{\mathbf{v}^i\}$  converges to  $\mathbf{K}\begin{bmatrix} \mathbf{x}^* \\ \mathbf{z}^* \end{bmatrix}$ .  $\square$

## APPENDIX C

### ADDITIONAL EXPERIMENTAL RESULTS

This Appendix presents the computing times for some of the experiments described in Section III-B.

TABLE II  
RESULTS FOR *cameraman* WITH BOXCAR BLURS OF VARIOUS SIZES.

Method	Blur size	$\kappa \times 10^3$	Iterations	Time (s)
Proposed-1	5	57.0	74	0.796
Proposed-AD			41	0.600
AM			179	3.072
ADMM-CG			25	20.416
CM			12028	52.405
Proposed-1	13	492.4	457	4.030
Proposed-AD			133	3.072
AM			401	4.376
ADMM-CG			49	92.345
CM			58623	259.226
Proposed-1	21	1356.5	410	3.628
Proposed-AD			95	3.244
AM			492	5.336
ADMM-CG			65	142.542
CM			122902	535.551

TABLE III  
RESULTS FOR *Lena* WITH BOXCAR BLURS OF VARIOUS SIZES.

Method	Blur size	$\kappa \times 10^3$	Iterations	Time (s)
Proposed-1	5	57.0	131	1.256
Proposed-AD			52	0.721
AM			264	3.032
ADMM-CG			25	21.188
CM			11353	49.062
Proposed-1	13	492.4	1038	8.908
Proposed-AD			140	3.192
AM			399	4.390
ADMM-CG			49	86.374
CM			51328	223.575
Proposed-1	21	1356.5	2202	18.713
Proposed-AD			176	5.761
AM			552	6.038
ADMM-CG			65	138.104
CM			118684	517.376

TABLE IV  
RESULTS FOR *Lena* WITH GAUSSIAN BLURS OF VARIOUS SIZES.

Method	Blur size	$\kappa \times 10^3$	Iterations	Time (s)
Proposed-1	5	247.0	196	1.828
Proposed-AD			73	0.958
AM			331	3.737
ADMM-CG			35	26.204
CM			13946	61.351
Proposed-1	13	21590.6	6093	52.058
Proposed-AD			776	16.905
AM			1032	11.344
ADMM-CG			81	120.265
CM			196053	867.026
Proposed-1	21	686384.8	68776	583.065
Proposed-AD			7047	233.340
AM			1454	15.863
ADMM-CG			205	345.285
CM			412249	1833.467

REFERENCES

[1] P. Campisi and K. Egiazarian, *Blind image deconvolution: Theory and applications*. Boca Raton, FL: CRC Press, 2007.

[2] A. Rajagopalan and R. Chellappa, *Motion Deblurring: Algorithms and Systems*. Cambridge, UK: Cambridge University Press, 2014.

[3] P. Milanfar, *Super-Resolution Imaging*. Boca Raton, FL, USA: CRC Press, 2010.

[4] R. Lukac, *Single-Sensor Imaging: Methods and Applications for Digital Cameras*. Boca Raton, FL, USA: CRC Press, 2008.

[5] M. Ng, R. Chan, and W. Tang, "A fast algorithm for deblurring models with neumann boundary conditions," *SIAM Journal on Scientific Computing*, vol. 21, no. 3, pp. 851–866, 1999.

[6] P. Hansen, J. Nagy, and D. O’Leary, *Deblurring Images: Matrices, Spectra and Filtering*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2006.

[7] M. Donatelli and S. Serra-Capizzano, "On the treatment of boundary artifacts in image restoration by reflection and/or anti-reflection," in *Matrix Methods: Theory, Algorithms and Applications*, V. Olshevsky and E. Tyrtyshnikov, Eds. Singapore: World Scientific Publishing Company, 2010.

[8] M. Almeida and M. Figueiredo, "Deconvolving images with unknown boundaries using the alternating direction method of multipliers," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3074–86, Aug. 2013.

[9] A. Matakos, S. Ramani, and J. Fessler, "Accelerated edge-preserving image restoration without boundary artifacts," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 2019–2029, 2013.

[10] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–95, 2011.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method

of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[12] A. Danielyan, V. Katkovnik, and K. Egiazarian, "BM3D frames and variational image deblurring," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1715–1728, April 2012.

[13] S. Reeves, "Fast image restoration without boundary artifacts," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1448–1453, 2005.

[14] R. Liu and J. Jia, "Reducing boundary artifacts in image deconvolution," in *Proc. 2009 IEEE Int. Conf. Image Processing*, San Diego, California, 2008, pp. 505–508.

[15] Y. Fan and J. Nagy, "Synthetic boundary conditions for image deblurring," *Linear Algebra Appl.*, vol. 434, no. 11, pp. 2244–2268, 2011.

[16] M. Sorel, "Removing boundary artifacts for real-time iterated shrinkage deconvolution," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2329–2334, 2012.

[17] M. Bertero and P. Boccacci, "A simple method for the reduction of boundary effects in the Richardson-Lucy approach to image deconvolution," *Astron. Astrophys.*, vol. 437, no. 1, pp. 397–403, 2005.

[18] R. Vio, J. Bardsley, M. Donatelli, and W. Wamsteker, "Dealing with edge effects in least-squares image deconvolution problems," *Astron. Astrophys.*, vol. 442, no. 1, pp. 397–403, 2005.

[19] T. Chan, A. Yip, and F. Park, "Simultaneous total variation image inpainting and blind deconvolution," *Int. J. Imaging Syst. Technol.*, vol. 15, no. 1, pp. 92–102, 2004.

[20] P. Combettes and J.-C. Pesquet, "Proximal Splitting Methods in Signal Processing," *ArXiv e-prints*, Dec. 2009.

[21] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2013.

[22] N. Komodakis and J.-C. Pesquet, "Playing with Duality: An Overview of Recent Primal-Dual Approaches for Solving Large-Scale Optimization Problems," *ArXiv e-prints*, Jun. 2014.

[23] L. Condat, "A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optimiz. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013.

[24] P. Combettes, L. Condat, J.-C. Pesquet, and B. C. Vu, "A forward-backward view of some primal-dual optimization methods in image recovery," in *IEEE Int. Conf. Image Processing*, Paris, France, 2014.

[25] D. O’Connor and L. Vandenberghe, "Primal-dual decomposition by operator splitting and applications to image deblurring," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1724–1754, 2014.

[26] P. Ferreira and M. Domnguez, "Trading-off matrix size and matrix structure: Handling Toeplitz equations by embedding on a larger circulant set," *Digit. Signal Process.*, vol. 20, no. 6, pp. 1711 – 1722, 2010.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[28] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3D transform-domain collaborative filtering," in *Proc. SPIE Electronic Imaging*, no. 6812-07, San Jose, CA, USA, 2008.

[29] L. Condat, "A generic proximal algorithm for convex optimization — application to total variation minimization," *Signal Processing Letters, IEEE*, vol. 21, no. 8, pp. 985–989, Aug 2014.

[30] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imaging Sci.*, vol. 1, no. 3, pp. 248–272, 2008.

[31] D. Donoho and I. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Am. Statist. Assoc.*, vol. 90, no. 432, pp. 1200–1224, 1995.

[32] G. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[33] M. Almeida and M. Figueiredo, "Parameter estimation for blind and non-blind deblurring using residual whiteness measures," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2751–2763, July 2013.

[34] B. He, S. Wang, and H. Yang, "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities," *J. Optimiz. Theory App.*, vol. 106, no. 2, pp. 337–356, 2000.

[35] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, no. 20, pp. 89–97, 2004.

[36] J. Nocedal and S. Wright, *Numerical Optimization*. New York: Springer, 2006.

[37] J. Eckstein and D. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 3, pp. 293–318, Jun. 1992.

[38] G. Chen and M. Teboulle, "A proximal-based decomposition method for convex minimization problems," *Math. Prog.*, vol. 64, no. 1-3, pp. 81–101, 1994.

[39] B. He, L. Liao, D. Han, and H. Yang, "A new inexact alternating directions method for monotone variational inequalities," *Math. Prog.*, vol. 92, no. 1, pp. 103–118, 2002.

[40] B. He and X. Yuan, "On non-ergodic convergence rate of Douglas-Rachford alternating direction method of multipliers," Nanjing University, Tech. Rep. 708044, 2012.

[41] W. Deng and W. Yin, "On the global and linear convergence of the generalized alternating direction method of multipliers," Dept. Computational Appl. Math., Rice Univ., Houston, TX, USA, Tech. Rep., 2012.

[42] J. Yang and Y. Zhang, "Alternating direction algorithms for  $\ell_1$ -problems in compressive sensing," *SIAM Journal on Scientific Computing*, vol. 33, no. 1, pp. 250–278, 2011.

[43] H. Bauschke and P. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York, NY, USA: Springer-Verlag New York, 2011.

[44] R. Rockafellar, *Convex Analysis*. New Jersey, USA: Princeton University Press, 1970.

[45] —, "Monotone operators and the proximal point algorithm," *SIAM J. Control Optim.*, vol. 14, no. 5, pp. 877–898, 1976.



**José Bioucas-Dias** (SM'15) received the EE, MSc, PhD, and "Agregado" degrees from Instituto Superior Técnico (IST), Technical University of Lisbon (TULisbon, now University of Lisbon), Portugal, in 1985, 1991, 1995, and 2007, respectively, all in electrical and computer engineering.

Since 1995, José Bioucas-Dias has been with the Department of Electrical and Computer Engineering, IST, where he was an Assistant Professor from 1995 to 2007 and an Associate Professor since 2007. Since 1993, he is also a Senior Researcher with the Pattern and Image Analysis group of the Instituto de Telecomunicações, which is a private non-profit research institution. His research interests include inverse problems, signal and image processing, pattern recognition, optimization, and remote sensing. His research work has been highly cited and he is included in Thomson Reuters' Highly Cited Researchers 2015 list.

José Bioucas-Dias was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS (1997-2000) and IEEE TRANSACTIONS ON IMAGE PROCESSING (2010-2014) and he is a Senior Area Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and an Associate Editor for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He was the General Co-Chair of the 3rd IEEE GRSS Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote sensing (WHISPERS'2011) and has been a member of program/technical committees of several international conferences.



**Miguel Simões** received the M.Sc. degree in electrical and computer engineering from the Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal, in 2010. He is currently working toward the joint Ph.D. degree in electrical and computer engineering, and signal and image processing at the Instituto de Telecomunicações, Instituto Superior Técnico, University of Lisbon, Lisbon, and at the Grenoble Images Parole Signal Automatique (GIPSA-lab), University of Grenoble, Grenoble, France, respectively.

He has previously worked as an Information Technology Consultant in the field of telecommunications. His main areas of research interest are image processing, optimization, and remote sensing.



**Jocelyn Chanussot** (M04SM04F12) received the M.Sc. degree in electrical engineering from the Grenoble Institute of Technology (Grenoble INP), Grenoble, France, in 1995, and the Ph.D. degree from the Université de Savoie, Annecy, France, in 1998. In 1999, he was with the Geography Imagery Perception Laboratory for the Délégation Générale de l'Armement (DGA - French National Defense Department). Since 1999, he has been with Grenoble INP, where he was an Assistant Professor from 1999 to 2005, an Associate Professor from 2005

to 2007, and is currently a Professor of signal and image processing. He is conducting his research at the Grenoble Images Speech Signals and Automatics Laboratory (GIPSA-Lab). His research interests include image analysis, multicomponent image processing, nonlinear filtering, and data fusion in remote sensing. He has been a visiting scholar at Stanford University (USA), KTH (Sweden) and NUS (Singapore). Since 2013, he is an Adjunct Professor of the University of Iceland. In 2015-2017, he is a visiting professor at the University of California, Los Angeles (UCLA).

Dr. Chanussot is the founding President of IEEE Geoscience and Remote Sensing French chapter (2007-2010) which received the 2010 IEEE GRSS Chapter Excellence Award. He was the co-recipient of the NOR SIG 2006 Best Student Paper Award, the IEEE GRSS 2011 and 2015 Symposium Best Paper Award, the IEEE GRSS 2012 Transactions Prize Paper Award and the IEEE GRSS 2013 Highest Impact Paper Award. He was a member of the IEEE Geoscience and Remote Sensing Society AdCom (2009-2010), in charge of membership development. He was the General Chair of the first IEEE GRSS Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote sensing (WHISPERS). He was the Chair (2009-2011) and Cochair of the GRS Data Fusion Technical Committee (2005-2008). He was a member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society (2006-2008) and the Program Chair of the IEEE International Workshop on Machine Learning for Signal Processing, (2009). He was an Associate Editor for the IEEE Geoscience and Remote Sensing Letters (2005-2007) and for Pattern Recognition (2006-2008). Since 2007, he is an Associate Editor for the IEEE Transactions on Geoscience and Remote Sensing. He was the Editor-in-Chief of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2011-2015). In 2013, he was a Guest Editor for the Proceedings of the IEEE and in 2014 a Guest Editor for the IEEE Signal Processing Magazine. He is a Fellow of the IEEE and a member of the Institut Universitaire de France (2012-2017).



**Luís B. Almeida** received the Doutor degree from the Technical University of Lisbon, Portugal, in 1983. Since 1972, he has been a faculty member of Instituto Superior Técnico (IST), Lisbon University, where he has been a full professor since 1995, lecturing in the areas of signal processing and machine learning. From 1984 to 2004, he was head of the Neural Networks and Signal Processing Group of INESC-ID. From 2000 to 2003, he was chair of INESC-ID. In 2005, he joined the Instituto de Telecomunicações (Telecommunications Institute).

From 2008 to 2010, he was chair of the Electrical and Computer Engineering Department of IST. Over the years, he has worked on speech modelling and coding, time-frequency representations of signals and the fractional Fourier transform, learning algorithms for neural networks, blind source separation, and, currently, image processing. He was the recipient of an IEEE Signal Processing Area ASSP Senior Award and of several national awards.