Accelerated Edge-Preserving Image Restoration Without Boundary Artifacts

Antonios Matakos, *Student Member, IEEE*, Sathish Ramani, *Member, IEEE*, and Jeffrey A. Fessler, *Fellow, IEEE*

Abstract—To reduce blur in noisy images, regularized image restoration methods have been proposed that use nonquadratic regularizers (like l_1 regularization or total-variation) that suppress noise while preserving edges in the image. Most of these methods assume a circulant blur (periodic convolution with a blurring kernel) that can lead to wraparound artifacts along the boundaries of the image due to the implied periodicity of the circulant model. Using a noncirculant model could prevent these artifacts at the cost of increased computational complexity. In this paper, we propose to use a circulant blur model combined with a masking operator that prevents wraparound artifacts. The resulting model is noncirculant, so we propose an efficient algorithm using variable splitting and augmented Lagrangian (AL) strategies. Our variable splitting scheme, when combined with the AL framework and alternating minimization, leads to simple linear systems that can be solved noniteratively using fast Fourier transforms (FFTs), eliminating the need for more expensive conjugate gradient-type solvers. The proposed method can also efficiently tackle a variety of convex regularizers, including edge-preserving (e.g., total-variation) and sparsity promoting (e.g., l_1-norm) regularizers. Simulation results show fast convergence of the proposed method, along with improved image quality at the boundaries where the circulant model is inaccurate.

Index Terms—Augmented Lagrangian (AL), edge-preserving regularization, image restoration, noncirculant system, variable splitting.

I. INTRODUCTION

IMAGE restoration is a well studied problem and there are many methods for deblurring and denoising. Usually image restoration is treated as an optimization problem where the restored image is obtained by minimizing a cost function consisting of a data fidelity term and a regularization term. The data fidelity term ensures good fit of the blur model to the measurements, and the regularizer ensures stability of the solution and incorporates smoothness to suppress noise. A quadratic data fidelity term is often used, based on the additive zero-mean Gaussian noise model. For the regularizer, using a quadratic term can lead to over-smoothing. Recently there is increasing interest in non-quadratic regularizers, especially

edge-preserving ones like total variation (TV) [1]–[3] and sparsity promoting ones like l_1 regularization [4], [5].

Most existing restoration methods make simplifying assumptions concerning the system model and the most common one is the use of a circulant blurring model [3]–[10] because it facilitates FFT-based optimization (e.g., non-iterative matrix inversions [6], [10]). Despite its popularity, the purely circulant model is inaccurate, since it implies an unrealistic periodic extension at the image boundaries that can lead to severe reconstruction artifacts [11].

In this work we focus on a more realistic non-circulant blurring model that is shift-invariant within the region of interest and propose an efficient algorithm for image restoration with non-quadratic regularization. Our model is equivalent to that proposed in [11], and similarly to [12] we focus on edge-preserving regularizers instead of the quadratic regularizer used in [11]. However, unlike [11], [12] which treat boundaries explicitly via a low-dimensional auxiliary variable, our approach is based on an elegant formulation that requires no pre-processing of the data or explicit treatment of the unknown, extrapolated boundaries. In addition, the methods in [11], [12] require CG-type solvers to optimize the associated auxiliary variable that is avoided in our proposed algorithm based on the AL framework with variable splitting.

Our formulation combines a circulant blur model with a masking operator to prevent wraparound artifacts. Then we use a specific variable splitting strategy that decouples the circulant blur and the mask. When combined with the AL framework and alternating minimization, our splitting leads to an iterative algorithm with simple update steps that can be implemented non-iteratively in closed-form. We present numerical results that illustrate the improved quality of reconstructed images using a non-circulant model and also the improved convergence speed of our proposed algorithm compared to other state-of-the-art methods that can be used to tackle the non-circulant reconstruction problem.

II. PROBLEM FORMULATION

Regularized image restoration can be approached in two main ways. One is the analysis formulation [13], where the objective is to obtain an estimate of the true image $\hat{x}$, and the other is the synthesis formulation [4], where the objective is to estimate a set of transform coefficients $\hat{w}$ and obtain the reconstructed image through a transform as $\hat{x} = W\hat{w}$. In this work, we focus on the analysis formulation. Our approach can

The authors are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: amatakos@umich.edu; sramani@umich.edu; fessler@umich.edu).

be extended to the synthesis one using techniques similar to that proposed in [6].

For image restoration, we consider the following analysis formulation, where the image estimate $\hat{\mathbf{x}}$ is obtained by minimizing a cost function

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}}\Big\{\Psi(\mathbf{x}) \triangleq J(\mathbf{x}) + \lambda\Phi(\mathbf{R}\mathbf{x})\Big\} \qquad (1)$$

$J(\cdot)$ is the data fidelity term, $\Phi(\cdot)$ is a regularizer function, $\mathbf{R}$ is a sparsifying transform (e.g., wavelet frames or finite differences) and λ is the regularization parameter.

A. Data Fidelity

To design a restoration algorithm one must make some modeling assumptions. Often deblurring algorithms are developed assuming a circulant blur model represented by $\check{\mathbf{A}}$ and the data fidelity term is modeled as [3]–[10]:

$$J(\check{\mathbf{x}}) \triangleq \big\|\mathbf{y} - \check{\mathbf{A}}\check{\mathbf{x}}\big\|_2^2 \qquad (2)$$

where $\mathbf{y}$ is the observed $(M \times 1)$ vectorized blurred and noisy image, $\check{\mathbf{x}}$ is the vectorized $(M \times 1)$ image to be reconstructed, and $\check{\mathbf{A}}$ is an $(M \times M)$ circulant blurring matrix. Even though this model is very popular in the image restoration literature [3]–[10], it is inaccurate since the assumption of circulant blur rarely, if ever, applies in practice.

Reconstructing images under the assumption of a purely circulant model can lead to severe artifacts due to discontinuities at the boundaries caused by the periodic extension of the image [11]. Simple approaches like zero-padding or replicated boundary extension do not resolve this issue since they do not eliminate the discontinuity at the boundaries. As suggested in [11] data pre-processing techniques like replicated boundary extension combined with edge-tapering can reduce, but not completely eliminate the artifacts. This will be illustrated in the experiments section (Section IV). A typical data preprocessing approach yields an augmented $(N \times 1$, with $N > M)$[1] data vector

$$\tilde{\mathbf{y}} \triangleq \text{edgetaper}\{\text{replicate}(\mathbf{y})\} \qquad (3)$$

where replicate corresponds to replicated boundary extension, e.g., for 1D $y[k] = y[n]$ for $k > n$ (implemented using Matlab's padarray function with the 'replicate' option), and edgetaper corresponds to edge-tapering, i.e., blurring the boundaries of the image using the known PSF to eliminate discontinuities caused by periodic extension [11, Section I] (implemented using Matlab's edgetaper function). A corresponding data fidelity term is given by:

$$J(\mathbf{x}) \triangleq \big\|\tilde{\mathbf{y}} - \mathbf{A}\mathbf{x}\big\|_2^2 \qquad (4)$$

where $\mathbf{x}$ is the $(N \times 1)$ image to be reconstructed, and $\mathbf{A}$ is an $(N \times N)$ circulant blurring matrix.

Instead of the circulant model (which uses periodic boundary conditions), an alternative is to use reflexive boundary conditions along with DCT, as suggested in [14]. In this case the blurring matrix has the Toeplitz-plus-Hankel form and can

be diagonalized using the DCT. A corresponding data fidelity term is given by:

$$J(\check{\mathbf{x}}) \triangleq \|\mathbf{y} - \mathbf{A}_R\check{\mathbf{x}}\|_2^2 \qquad (5)$$

where $\check{\mathbf{x}}$ is the $(M \times 1)$ image to be reconstructed, and $\mathbf{A}_R$ is an $(M \times M)$ Toeplitz-plus-Hankel blurring matrix. This DCT method requires the PSF to be symmetric [11], [14], which may not hold for some applications e.g., motion blur. In cases of non-symmetric PSFs this method can only be used to find preconditioners that facilitate iterative solvers like CG [14].

To eliminate boundary artifacts, the methods in [11], [12] use a data fidelity term

$$J(\mathbf{x}) \triangleq \big\|\mathbf{y} - \tilde{\mathbf{A}}\mathbf{x}\big\|_2^2 \qquad (6)$$

with an $(M \times N)$ non-circulant blurring matrix $\tilde{\mathbf{A}}$. Then, the non-circulant matrix $\tilde{\mathbf{A}}$ is augmented with additional rows (using a low dimensional auxiliary matrix $\mathbf{a}$) to create a circulant $(N \times N)$ system matrix $\mathbf{A}$. Instead of tackling the non-circulant Hessian $\tilde{\mathbf{A}}'\tilde{\mathbf{A}}$ directly, it is rewritten as:

$$\tilde{\mathbf{A}}'\tilde{\mathbf{A}} = \mathbf{A}'\mathbf{A} - \mathbf{a}'\mathbf{a} \qquad (7)$$

consisting of a block-circulant (with circulant blocks) Hessian $\mathbf{A}'\mathbf{A}$ and a low rank component $\mathbf{a}'\mathbf{a}$. Inverting the non-circulant Hessian $\tilde{\mathbf{A}}'\tilde{\mathbf{A}}$ (along with a suitable block-circulant regularization matrix) is then carried out using the Sherman–Morrison Matrix Inversion Lemma (MIL) [11, Eq. (5)–(7)]. The MIL separates the two terms in the RHS of (7) and involves inverting the circulant part (i.e., $\mathbf{A}'\mathbf{A}$ with the regularization term) and a low-dimensional matrix involving the auxiliary matrix $\mathbf{a}$. The circulant component is inverted using FFTs and inversion of the low dimensional component can be performed with an iterative algorithm (e.g., CG).

In this work, similarly to [11], [12], we consider a more realistic non-circulant model, that is shift-invariant within the region of interest, but avoids the assumption of periodic end conditions. However, instead of using the manipulation in (7), we introduce a masking operator that eliminates the wraparound artifacts at the boundaries caused by periodic convolution. With our proposed model the data fidelity term becomes

$$J(\mathbf{x}) \triangleq \|\mathbf{y} - \mathbf{T}\mathbf{A}\mathbf{x}\|_2^2 \qquad (8)$$

where $\mathbf{T}$ is an $(M \times N)$ masking matrix that truncates the circular wraparound at the boundaries, $\mathbf{A}$ is an $(N \times N)$ circulant matrix, and $\mathbf{x}$ is an $(N \times 1)$ vector as in (4) and (6)[2]. The matrix $\mathbf{T}$ is a truncated identity matrix resulting from the removal of rows corresponding to boundary pixels, and thus, $\mathbf{T}'\mathbf{T}$ is an $(N \times N)$ diagonal matrix with 0s and 1s. Even though the model in (8) is shift-invariant within the image, the masking operator makes the overall system model $\mathbf{T}\mathbf{A}$ shift-variant. To efficiently handle this shift-variant model, we

---

[1] For an observed image $\mathbf{y}$ of size $M = N_y \times N_y$ and a PSF of size $N_h \times N_h$ the size of the processed data is $N = (N_y + N_h - 1) \times (N_y + N_h - 1)$.

[2] For the overall system model we have $\mathbf{T}\mathbf{A} = \tilde{\mathbf{A}}$ and it corresponds to a non-circular convolution operator with extended end conditions. For a blurring kernel $\mathbf{h}$ of size $N_h \times N_h$ we have $\mathbf{y} = \mathbf{h} * *\mathbf{x}$ and the size of $\mathbf{x}$ is $N = (N_y + N_h - 1) \times (N_y + N_h - 1)$, which corresponds to the size of the unknown image that contributes to the observed $N_y \times N_y$ image.

propose to use a variable splitting scheme that decouples $\mathbf{T}$ and $\mathbf{A}$ and in turn allows the use of FFT-based computations as described in Section III-B.

B. Regularizer

The algorithms discussed in this paper can tackle a general class of convex regularizers $\Phi(\cdot)$ in (1), but we will focus on edge-preserving and sparsity promoting regularizers like Total Variation (TV) and analysis l_1 regularization, respectively, for brevity.

1) l_1 regularization or discrete anisotropic total-variation:

$$\Phi(\mathbf{Rx}) = \|\mathbf{Rx}\|_1 \tag{9}$$

where $\mathbf{R} = \mathbf{W}$ is a wavelet frame, typically excluding the approximation level, or $\mathbf{R} = \mathbf{C} \triangleq [\mathbf{C}_1' \ \mathbf{C}_2']'$ is a matrix of horizontal and vertical finite differences.

2) Discrete isotropic total-variation

$$\Phi(\mathbf{Cx}) = \sum_{n=1}^{N} \sqrt{\left|[\mathbf{C}_1\mathbf{x}]_n\right|^2 + \left|[\mathbf{C}_2\mathbf{x}]_n\right|^2}. \tag{10}$$

To reduce computational complexity and allow FFT-based non-iterative solving of linear systems, the sparsifying matrix $\mathbf{R}$ (wavelet frame or finite differences) is assumed to have periodic end conditions such that $\mathbf{R}'\mathbf{R}$ is circulant.

III. PROPOSED MODEL-BASED RESTORATION

In this work, our goal is to estimate the image $\mathbf{x}$ by minimizing the following cost function that we call problem **P1**:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \Psi(\mathbf{x}) \triangleq \frac{1}{2}\|\mathbf{y} - \mathbf{TAx}\|_2^2 + \lambda\Phi(\mathbf{Rx}) \right\}. \tag{11}$$

Minimizing (11) is a non-trivial optimization problem. Existing methods include non-linear conjugate gradient (**NCG**) [15], iterative shrinkage/thresholding (**ISTA**) [4], [16], (**M)FISTA** [8], [17], and variable splitting/ADMM (Alternating Direction Method of Multipliers) algorithms [6], [10], [18]–[20]. Some of these methods, e.g., (**M)FISTA**, **SALSA** [6], **Split-Bregman** (**SB**) [20], and **FTVd** [18] are computationally efficient when used with the data model in (2) since they exploit the circulant nature of $\mathbf{A}$ in (2). However, a straightforward application of these methods to the non-circulant model in (8) may increase their computation time since the inner sub-probems of these algorithms may no longer admit explicit closed-form updates. We discuss this in detail in Section III-A and also provide experimental evidence in Section IV-B. To handle the non-circulant model in (8) we propose a specific variable splitting strategy [6], [21] presented in Section III-B. We then use the AL framework and alternating minimization to obtain an iterative algorithm that exploits the structures of $\mathbf{T}$ and $\mathbf{A}$ and thereby, converges faster.

A. Existing Restoration Algorithms

1) Nonlinear Conjugate Gradient (NCG): Using **NCG** to solve **P1** requires computing the gradient of $\Psi(\mathbf{x})$. This is problematic for TV or l_1 norm regularizers that use the non-smooth absolute value function. To alleviate this problem the common approach is to use a rounding parameter to approximate the absolute value function as

$$|x| \approx \sqrt{|x|^2 + \varepsilon} \tag{12}$$

where ε is the rounding parameter. **NCG** also needs a line search method for which we can use the technique proposed in [15, Sec. IV].

The disadvantages of **NCG** are slow convergence, and perhaps, the fact that it does not converge to a solution of **P1**, due to the approximation in (12).

2) (MF)ISTA: **ISTA** was first introduced for restoration problems with synthesis-type priors [4], [16] and it was later generalized to analysis-type priors [8], [17]. (**M)FISTA** is an improvement on **ISTA** that converges faster to a solution of **P1**. The general methodology of both **ISTA** and (**M)FISTA** is to convert the original problem **P1** into a denoising problem that can be solved non-iteratively for l_1-synthesis priors or iteratively for analysis priors. The difference between **ISTA** and (**M)FISTA** is the use of a specific two-step update in (**M)FISTA** [8, Eq. (4.1)–(4.3)] that accelerates convergence [8, Thm. 4.4]. For analysis regularizers, the denoising step cannot be performed in a single-step, in which case one can use a Chambolle-type algorithm as in [7, Eq. (5)–(7)].

Even though **MFISTA** converges faster than the simpler **ISTA**, variable-splitting/ADMM algorithms [6], [10], [21] have been developed that exhibit faster convergence.

3) Variable-Splitting/ADMM Algorithms [18], [20], [6]: The main idea of these algorithms is to break down the original problem **P1** into smaller tasks by introducing appropriate auxiliary constraint variables. The resulting minimization subproblems are decoupled and easier to solve compared to the original minimization problem **P1**. There are several different ways to choose the splitting variables that lead to a variety of such variable-splitting based algorithms [6], [10], [21]. These algorithms have been shown to converge faster than **MFISTA** [6]. The algorithm we developed for the non-circulant image restoration problem is based on the variable-splitting ADMM framework presented in [10], [21], [22].

a) SALSA [6]: One approach for solving **P1** is to split the regularization term by introducing an auxiliary variable $\mathbf{u} = \mathbf{x}$. The constrained problem is formulated as:

$$\min_{\mathbf{x},\mathbf{u}} \left\{ \Psi(\mathbf{x},\mathbf{u}) \triangleq \tfrac{1}{2}\|\mathbf{y} - \mathbf{TAx}\|_2^2 + \lambda\Phi(\mathbf{Ru}) \right\} \text{ s.t. } \mathbf{u} = \mathbf{x} \tag{13}$$

and the associated AL function is

$$\mathcal{L}(\mathbf{x},\mathbf{u},\mu,\boldsymbol{\eta}) = \Psi(\mathbf{x},\mathbf{u}) + \frac{\mu}{2}\|\mathbf{u} - \mathbf{x} - \boldsymbol{\eta}\|_2^2 \tag{14}$$

where $\boldsymbol{\eta}$ is linearly related to the Lagrange multiplier for the constraint in (13) and $\mu > 0$ is an AL penalty parameter [6], [10], [21], [22].

This formulation leads to **SALSA** [6], where (13) is solved with the following alternating minimization scheme:

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \begin{array}{l} \frac{1}{2} \left\| \mathbf{y} - \mathbf{TAx} \right\|_2^2 \\ + \frac{\mu}{2} \left\| \mathbf{u}^{(k)} - \mathbf{x} - \boldsymbol{\eta}^{(k)} \right\|_2^2 \end{array} \right\} \quad (15)$$

$$\mathbf{u}^{(k+1)} = \underset{\mathbf{u}}{\operatorname{argmin}} \left\{ \lambda \Phi(\mathbf{Ru}) + \frac{\mu}{2} \left\| \mathbf{u} - \mathbf{x}^{(k+1)} - \boldsymbol{\eta}^{(k)} \right\|_2^2 \right\} \quad (16)$$

$$\boldsymbol{\eta}^{(k+1)} = \boldsymbol{\eta}^{(k)} - \mathbf{u}^{(k+1)} + \mathbf{x}^{(k+1)}. \quad (17)$$

Since the cost function (15) is quadratic, the minimization with respect to $\mathbf{x}$ can be expressed in closed form as:

$$\mathbf{x}^{(k+1)} = \mathbf{H}_{\mu,1}^{-1} \left[\mathbf{A}'\mathbf{T}'\mathbf{y} + \mu \left(\mathbf{u}^{(k+1)} - \boldsymbol{\eta}_1^{(k)} \right) \right] \quad (18)$$

where

$$\mathbf{H}_{\mu,1} \triangleq \mathbf{A}'\mathbf{T}'\mathbf{TA} + \mu \mathbf{I}. \quad (19)$$

For the purely circulant model in (2), $\mathbf{T} = \mathbf{I}$ and one can invert $\mathbf{H}_{\mu,1}$ using FFTs. However, for the more realistic model in (8), the Hessian $\mathbf{H}_{\mu,1}$ has no exploitable structure and finding a minimizer non-iteratively can be computationally intensive for large N. Instead, one can apply a few PCG iterations using the circulant preconditioner $\mathbf{M}_1 \triangleq \left(\mathbf{A}'\mathbf{A} + \mu \mathbf{I} \right)^{-1}$ along with warm starting to find an approximate solution.

The minimization with respect to $\mathbf{u}$ in (16) can be implemented non-iteratively for certain synthesis-type regularizers. However, for analysis-type regularizers (e.g., involving finite differences or wavelet frames), it has to be performed iteratively. Following the implementation of **SALSA** [6], we used the Chambolle-type algorithm [7, Eq. (6)–(7)] for the update of $\mathbf{u}$. Despite the approximate update steps, **SALSA** can be shown to converge to a solution of **P1** [6], [23].

b) Split-Bregman Algorithm [20]: An alternative approach is to split the regularization term by introducing the auxiliary variable $\mathbf{v} = \mathbf{Rx}$. In this case the constrained problem is formulated as

$$\min_{\mathbf{x},\mathbf{v}} \left\{ \Psi(\mathbf{x}, \mathbf{v}) \triangleq \frac{1}{2} \left\| \mathbf{y} - \mathbf{TAx} \right\|_2^2 + \lambda \Phi(\mathbf{v}) \right\} \text{ s.t. } \mathbf{v} = \mathbf{Rx} \quad (20)$$

and the associated AL function is

$$\mathcal{L}(\mathbf{x}, \mathbf{v}, \mu, \boldsymbol{\eta}) = \Psi(\mathbf{x}, \mathbf{v}) + \frac{\mu}{2} \left\| \mathbf{v} - \mathbf{Rx} - \boldsymbol{\eta} \right\|_2^2 \quad (21)$$

where $\boldsymbol{\eta}$ is related to the Lagrange multiplier for the constraint in (20).

The splitting in (20) is similar to that in the **SB** [20] and **FTVd** [18] algorithms, and the problem is again solved using the AL framework and by performing the following minimizations alternatively:

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ \begin{array}{l} \frac{1}{2} \left\| \mathbf{y} - \mathbf{TAx} \right\|_2^2 \\ + \frac{\mu}{2} \left\| \mathbf{v}^{(k)} - \mathbf{Rx} - \boldsymbol{\eta}^{(k)} \right\|_2^2 \end{array} \right\} \quad (22)$$

$$\mathbf{v}^{(k+1)} = \underset{\mathbf{v}}{\operatorname{argmin}} \left\{ \lambda \Phi(\mathbf{v}) + \frac{\mu}{2} \left\| \mathbf{v} - \mathbf{Rx}^{(k+1)} - \boldsymbol{\eta}^{(k)} \right\|_2^2 \right\} \quad (23)$$

$$\boldsymbol{\eta}^{(k+1)} = \boldsymbol{\eta}^{(k)} - \mathbf{v}^{(k+1)} + \mathbf{Rx}^{(k+1)}. \quad (24)$$

As in **SALSA**, the minimization with respect to $\mathbf{x}$ in (22) has a closed-form solution since the associated cost function is quadratic:

$$\mathbf{x}^{(k+1)} = \mathbf{H}_{\mu,2}^{-1} \left[\mathbf{A}'\mathbf{T}'\mathbf{y} + \mu \mathbf{R}' \left(\mathbf{v}^{(k+1)} - \boldsymbol{\eta}^{(k)} \right) \right] \quad (25)$$

where

$$\mathbf{H}_{\mu,2} \triangleq \mathbf{A}'\mathbf{T}'\mathbf{TA} + \mu \mathbf{R}'\mathbf{R}. \quad (26)$$

Again in this case, for the model in (8), the Hessian matrix $\mathbf{H}_{\mu,2}$ has no exploitable structure and solving (25) can be computationally intensive for large N. As in **SALSA**, one can apply a few PCG iterations with warm starting and a circulant preconditioner $\mathbf{M}_2 \triangleq \left(\mathbf{A}'\mathbf{A} + \mu \mathbf{R}'\mathbf{R} \right)^{-1}$.

The minimization with respect to $\mathbf{v}$ in (23) can be performed non-iteratively for several synthesis- and analysis-type regularizers, including TV and l_1 norm of wavelet coefficients, using a soft thresholding/shrinkage rule:

$$\mathbf{v}^{(k+1)} = \operatorname{shrink} \left\{ \mathbf{Rx}^{(k)} + \boldsymbol{\eta}^{(k)}, \frac{\lambda}{\mu} \right\}. \quad (27)$$

For the bilateral TV and l_1 regularization the shrinkage operator is element-wise, whereas for isotropic TV a vector shrinkage rule can be used, as in [18, Section 2.1].

A reviewer suggested a variation of the **SB** algorithm, formulated by using the method in [11] for the updates of $\mathbf{x}$ in (22). Although it is more complicated than applying CG to (25), this formulation has the benefit of solving a much smaller problem using CG iterations only for the boundaries and updates the rest of the image with the use of FFTs. The method is explained in detail in [11] and also in the alternating minimization scheme of [12]. For this method one could apply CG iterations for the boundaries only at the first few outer iterations of the algorithm to reduce computation time. Such a scheme would not lead to a solution of **P1** in (11), but can yield images with improved quality compared to any pre-processing method. Depending on the noise level and type of blur the restored image could even be practically indistinguishable from the actual solution of **P1**. However, there is no way to know in advance for how many steps should the CG sub-iterations be applied to achieve restoration with sufficiently reduced artifacts. For the purpose of convergence speed comparisons we will refer to this variation as the **Split-Bregman-MIL** (**SB-MIL**) algorithm. This name was used since the method uses the Matrix Inversion Lemma (MIL) to decouple the circulant part from the boundaries and efficiently perform the update for $\mathbf{x}$.

Compared to **SALSA**, the benefit of **SB** and **SB-MIL** is the single-step update for $\mathbf{v}$ that avoids the inner iterations of the Chambolle-type [7, Eq. (6)–(7)] algorithm required in **SALSA**.

B. Proposed Algorithm: ADMM-P2

Even though both **SALSA** and **SB**-based algorithms (e.g., **SB** [20] and **SB-MIL** studied here) decouple the regularization term, their main caveat is the need for (P)CG iterations to obtain the update of $\mathbf{x}$ ((18) and (25)) or a low-dimension vector corresponding to the boundaries in **SB-MIL**. These

inner iterations can increase convergence time. Our method alleviates this problem by introducing a second auxiliary variable $\mathbf{u}_0 = \mathbf{A}\mathbf{x}$. This additional splitting is similar to those used in [21, Sec. IV-B] for MRI, [22, Sec. III] for CT, and [10] for image restoration, although in [10] the goal is to separate the entire data-fidelity term from the data model, whereas in our case the additional splitting $\mathbf{u}_0$ separates the blur and the masking operator inside the data-fidelity term in (11). The resulting constrained problem **P2** is given by:

$$\min_{\mathbf{x},\mathbf{u}_0,\mathbf{u}_1} \left\{ \Psi(\mathbf{u}_0, \mathbf{u}_1) \triangleq \tfrac{1}{2} \|\mathbf{y} - \mathbf{T}\mathbf{u}_0\|_2^2 + \lambda \Phi(\mathbf{u}_1) \right\} \quad (28)$$
$$\text{s.t. } \mathbf{u}_0 = \mathbf{A}\mathbf{x} \text{ and } \mathbf{u}_1 = \mathbf{R}\mathbf{x}.$$

where $\mathbf{A}$ is the $N \times N$ circulant blurring matrix and $\mathbf{R}$ is an $R \times N$ sparsifying transform matrix. We can now rewrite **P2** in a more concise form as follows:

$$\min_{\mathbf{x},\mathbf{w}} \Psi(\mathbf{w}) \text{ s.t. } \mathbf{w} = \mathbf{D}\mathbf{x} \quad (29)$$

where

$$\mathbf{x} \triangleq \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \end{bmatrix}, \qquad \mathbf{D} \triangleq \begin{bmatrix} \mathbf{A} \\ \mathbf{R} \end{bmatrix}. \quad (30)$$

From this formulation it is evident that the constraint matrix $\mathbf{D}$ is full column rank provided that the null spaces of $\mathbf{A}'\mathbf{A}$ and $\mathbf{R}'\mathbf{R}$ intersect only trivially, which is usually true in image restoration problems (low-pass nature of $\mathbf{A}$ and high-pass nature of $\mathbf{R}$). This requirement does not impose serious restrictions on $\mathbf{R}$ for deblurring problems, *i.e.*, it is not necessary for $\mathbf{R}$ to be full column rank, which can be the case with commonly used sparsifying transforms, e.g., finite differences, wavelets (excluding the approximation level). The importance of $\mathbf{D}$ being full column rank relates to the hypothesis of a theorem due to Eckstein and Bertsekas [23, Thm. 8], summarized below, that guarantees convergence of ADMM type algorithms.

To solve the problem **P2** with the method of multipliers we construct the Augmented Lagrangian (AL) function, which can be expressed as:

$$\mathcal{L}(\mathbf{x}, \mathbf{w}, \boldsymbol{\gamma}) = \Psi(\mathbf{w}) + \tfrac{\mu}{2} \|\mathbf{w} - \mathbf{D}\mathbf{x} - \boldsymbol{\eta}\|_{\boldsymbol{\Lambda}}^2, \quad (31)$$

where μ is the AL penalty parameter, the parameter $\boldsymbol{\eta}$ relates to the Lagrange multiplier, and $\boldsymbol{\Lambda}$ is a symmetric positive definite weighting matrix defined as

$$\boldsymbol{\Lambda} \triangleq \begin{bmatrix} \mathbf{I}_N & \mathbf{0} \\ \mathbf{0} & \nu \mathbf{I}_R \end{bmatrix}, \quad (32)$$

where $\nu > 0$.

To solve **P1** in (28) we apply an alternating minimization scheme, which at the kth iteration, leads to the following updates

$$\mathbf{x}^{(k+1)} = \operatorname*{argmin}_{\mathbf{x}} \left\| \mathbf{w}^{(k)} - \mathbf{D}\mathbf{x} - \boldsymbol{\eta}^{(k)} \right\|_{\boldsymbol{\Lambda}}^2 \quad (33)$$

$$\mathbf{w}^{(k+1)} = \operatorname*{argmin}_{\mathbf{w}} \Psi(\mathbf{w}) + \tfrac{\mu}{2} \left\| \mathbf{w} - \mathbf{D}\mathbf{x}^{(k+1)} - \boldsymbol{\eta}^{(k)} \right\|_{\boldsymbol{\Lambda}}^2 \quad (34)$$

$$\boldsymbol{\eta}^{(k+1)} = \boldsymbol{\eta}^{(k)} - \mathbf{w}^{(k+1)} + \mathbf{D}\mathbf{x}^{(k+1)}. \quad (35)$$

Although the theorem [23, Thm. 8] allows for inexact updates, *i.e.*, $\varepsilon_{\mathbf{x}}^k \triangleq \left\| \mathbf{x}^{(k)} - \mathbf{x}_*^{(k)} \right\|_2 > 0$ and $\varepsilon_{\mathbf{w}}^k \triangleq \left\| \mathbf{w}^{(k)} - \mathbf{w}_*^{(k)} \right\|_2 > 0$ where $\left(\mathbf{x}_*^{(k)} \text{ and } \mathbf{w}_*^{(k)} \right)$ are solutions of (33) and (34) respectively, our formulation leads to closed-form updates for (33)–(34), that can be implemented non-iteratively, *i.e.*, $\varepsilon_{\mathbf{x}}^k = \varepsilon_{\mathbf{w}}^k = 0$ (up to numerical round-off errors). We review below the theorem of Eckstein and Bertsekas [23, Thm. 8] that is useful for analyzing the convergence of (33)–(35).

Theorem 1: Consider (28) where $\Psi(\cdot)$ is closed, proper and convex and $\mathbf{D}$ has full column rank. Let $\boldsymbol{\eta}^{(0)} \in \mathcal{R}^{N+R}, \mu > 0, \sum_k \varepsilon_{\mathbf{x}}^k < \infty$, and $\sum_k \varepsilon_{\mathbf{w}}^k < \infty$. If (28) has a solution $(\mathbf{x}^*, \mathbf{w}^*)$ then the sequence of updates $\left\{ \left(\mathbf{x}^{(k)}, \mathbf{w}^{(k)} \right) \right\}_k$ generated by (33)–(35) converges to $(\mathbf{x}^*, \mathbf{w}^*)$. If (28) has no solution then at least one of the sequences $\left\{ \left(\mathbf{x}^{(k)}, \mathbf{w}^{(k)} \right) \right\}_k$ or $\left\{ \boldsymbol{\eta}^{(k)} \right\}_k$ diverges.

Even though the theorem of Eckstein and Bertsekas [23, Thm. 8] uses an AL function with $\boldsymbol{\Lambda} = \mathbf{I}$, we can still apply the theorem to (31) through a simple variable transformation [22, Footnote 3].

Using the structure of $\Psi(\mathbf{w})$ and $\mathbf{D}$, we can reformulate the AL function from (31) as

$$\mathcal{L}(\mathbf{x}, \mathbf{u}, \mu, \nu, \boldsymbol{\eta}) = \Psi(\mathbf{u}_0, \mathbf{u}_1) + \tfrac{\mu}{2} \|\mathbf{u}_0 - \mathbf{A}\mathbf{x} - \boldsymbol{\eta}_0\|_2^2 \quad (36)$$
$$+ \tfrac{\mu\nu}{2} \|\mathbf{u}_1 - \mathbf{R}\mathbf{x} - \boldsymbol{\eta}_1\|_2^2,$$

where $\boldsymbol{\eta}_0$ and $\boldsymbol{\eta}_1$ are related to the Lagrange multipliers for the constraints in (28). Using this form of the AL function, the alternating minimization scheme at the kth step can be expressed as:

$$\mathbf{u}_0^{(k+1)} = \operatorname*{argmin}_{\mathbf{u}_0} \left\{ \begin{array}{l} \tfrac{1}{2} \|\mathbf{y} - \mathbf{T}\mathbf{u}_0\|_2^2 \\ + \tfrac{\mu}{2} \left\| \mathbf{u}_0 - \mathbf{A}\mathbf{x}^{(k)} - \boldsymbol{\eta}_0^{(k)} \right\|_2^2 \end{array} \right\} \quad (37)$$

$$\mathbf{u}_1^{(k+1)} = \operatorname*{argmin}_{\mathbf{u}_1} \left\{ \lambda \Phi(\mathbf{u}_1) + \tfrac{\mu\nu}{2} \left\| \mathbf{u}_1 - \mathbf{R}\mathbf{x}^{(k)} - \boldsymbol{\eta}_1^{(k)} \right\|_2^2 \right\} \quad (38)$$

$$\mathbf{x}^{(k+1)} = \operatorname*{argmin}_{\mathbf{x}} \left\{ \begin{array}{l} \tfrac{\mu}{2} \left\| \mathbf{u}_0^{(k+1)} - \mathbf{A}\mathbf{x} - \boldsymbol{\eta}_0^{(k)} \right\|_2^2 \\ + \tfrac{\mu\nu}{2} \left\| \mathbf{u}_1^{(k+1)} - \mathbf{R}\mathbf{x} - \boldsymbol{\eta}_1^{(k)} \right\|_2^2 \end{array} \right\} \quad (39)$$

$$\boldsymbol{\eta}_0^{(k+1)} = \boldsymbol{\eta}_0^{(k)} - \mathbf{u}_0^{(k+1)} + \mathbf{A}\mathbf{x}^{(k+1)} \quad (40)$$

$$\boldsymbol{\eta}_1^{(k+1)} = \boldsymbol{\eta}_1^{(k)} - \mathbf{u}_1^{(k+1)} + \mathbf{R}\mathbf{x}^{(k+1)}. \quad (41)$$

The minimization with respect to $\mathbf{u}_0$ in (37) is straightforward since the corresponding cost function is quadratic. The closed-form solution is expressed as

$$\mathbf{u}_0^{(k+1)} = \left(\mathbf{T}'\mathbf{T} + \mu \mathbf{I}_N \right)^{-1} \left[\mathbf{T}'\mathbf{y} + \mu \left(\mathbf{A}\mathbf{x}^{(k)} + \boldsymbol{\eta}_0^{(k)} \right) \right] \quad (42)$$

and can be easily computed in a single-step update since it only requires inversion of the diagonal matrix $\left(\mathbf{T}'\mathbf{T} + \mu \mathbf{I}_N \right)$.

The minimization with respect to $\mathbf{u}_1$ in (38) can be performed non-iteratively using a (vector) soft thresholding/shrinkage rule similar to (27), using a threshold value $\frac{\lambda}{\mu\nu}$ instead of $\frac{\lambda}{\mu}$ used in (27).

Finally the minimization with respect to $\mathbf{x}$ in (39) is also straightforward due to the corresponding quadratic cost

---

Algorithm 1 ADMM-P2: AL Algorithm for P2

1: Select $\mathbf{x}^{(0)}$, $\nu > 0$, and $\mu > 0$
2: Precompute $\mathbf{T}'\mathbf{y}$
3: Set $\boldsymbol{\eta}_0^{(0)} = 0$, $\boldsymbol{\eta}_1^{(0)} = 0$ and $k = 0$
4: **repeat**
5:    Obtain $\mathbf{u}_0^{(k+1)}$ using (42)
6:    Obtain $\mathbf{u}_1^{(k+1)}$ using (27)
7:    Obtain $\mathbf{x}^{(k+1)}$ using (43)
8:    $\boldsymbol{\eta}_0^{(k+1)} = \boldsymbol{\eta}_0^{(k)} - \left(\mathbf{u}_0^{(k+1)} - \mathbf{A}\mathbf{x}^{(k+1)} \right)$
9:    $\boldsymbol{\eta}_1^{(k+1)} = \boldsymbol{\eta}_1^{(k)} - \left(\mathbf{u}_1^{(k+1)} - \mathbf{R}\mathbf{x}^{(k+1)} \right)$
10:    $k = k + 1$
11: **until** stop criterion is met

---

function. The closed form of the update is

$$\mathbf{x}^{(k+1)} = \mathbf{H}_\nu^{-1} \left[\mathbf{A}' \left(\mathbf{u}_0^{(k+1)} - \boldsymbol{\eta}_0^{(k)} \right) + \nu \mathbf{R}' \left(\mathbf{u}_1^{(k+1)} - \boldsymbol{\eta}_1^{(k)} \right) \right] \tag{43}$$

where

$$\mathbf{H}_\nu \triangleq \mathbf{A}'\mathbf{A} + \nu \mathbf{R}'\mathbf{R}. \tag{44}$$

We perform multiplication by $\mathbf{H}_\nu^{-1}$ in (43) efficiently using FFTs since the blurring matrix $\mathbf{A}$ is circulant and the sparsifying operator $\mathbf{R}$ (frame or differencing matrix) consists of circulant blocks (periodic end conditions) such that $\mathbf{R}'\mathbf{R}$ is circulant. Thus, the update of $\mathbf{x}$ can be computed non-iteratively.

Combining the above expressions for the updates, we now summarize the proposed AL algorithm for solving **P2** as shown in Algorithm 1.

Unlike **SB**-based algorithms (including **SB-MIL**) and **SALSA** for solving (11), all the steps of **ADMM-P2** are single-step updates due to the extra splitting that decouples the circulant blur operator $\mathbf{A}$ from the masking operator $\mathbf{T}$. The experiments in Section IV-B show that the non-iterative updates of **ADMM-P2** help achieve convergence in less time than other methods.

C. AL Parameter Selection

SALSA and **ADMM-P2** belong to the general class of ADMM algorithms. Thus, the ADMM convergence theorem of Eckstein and Bertsekas [23, Thm. 8] applies in these cases and the algorithms are guaranteed to converge to a solution of the original problem **P1**, even when the inner minimization steps are not performed exactly. The **SB**-based algorithms are also convergent, although the convergence theory is different from that of ADMM [19], [20].

SALSA, **ADMM-P2**, **SB**, and **SB-MIL** are guaranteed to converge regardless of the choice of AL parameters μ and ν [23]. However, the choice of AL parameters affects the convergence speed of these algorithms. We found experimentally that the best AL parameters for fast convergence depend on the regularization parameter λ and the maximum intensity ($x_{\max}$) of the blurred image, similarly to the choice proposed in [9]. The best parameters can also depend on the spread of eigenvalues of $\mathbf{A}'\mathbf{A}$ and $\mathbf{R}'\mathbf{R}$.

For **SALSA** we found that a choice of $\mu = 2^7 \lambda / x_{\max}$ works well for several different choices of regularization parameter λ.

For **SB** and **SB-MIL**, we found experimentally that the choice of the AL parameter μ also depends on $\mu_{\min} \triangleq \arg\min_\mu \kappa \left(\mathbf{A}'\mathbf{A} + \mu \mathbf{R}'\mathbf{R} \right)$, where $\kappa(\cdot)$ is the condition number. As a rule of thumb we choose the AL parameter as $\mu = 2^8 \lambda \mu_{\min} / x_{\max}$.

For **ADMM-P2**, we found experimentally that the best product $\mu\nu$ depends on λ, $x_{\max}$ and $\nu_{\min} \triangleq \arg\min_\nu \kappa \left(\mathbf{A}'\mathbf{A} + \nu \mathbf{R}'\mathbf{R} \right)$. Thus, for fast convergence we choose the AL parameters to achieve $\mu\nu = 2^8 \lambda \nu_{\min} / x_{\max}$. Also, we found that we can fix the parameter μ to a value $\mu_0 = 2^{-4}$ and then choose the parameter ν as: $\nu = 2^8 \lambda \nu_{\min} / \mu_0 x_{\max}$.

The above rules for the parameter selection work well when the blurring system matrix $\mathbf{A}$ is scaled such that the maximum eigenvalue of $\mathbf{A}'\mathbf{A}$ is equal to 1.

IV. EXPERIMENTS

The first part of the experiments compares the quality of the reconstructed images obtained using the proposed non-circulant (8) and the purely circulant (2) reconstruction models. The second part compares the convergence speed of the proposed **ADMM-P2** algorithm to that of **NCG**, **ISTA**, **MFISTA**, **SALSA**, **SB**, and **SB-MIL** for restoration using the proposed non-circulant model.

A. Noncirculant Versus Circulant Restoration

For all our experiments we used the 256×256 cameraman image as the true, noise-free image, and following the experimental setting in [8] we scaled it to the range $[0, 1]$ (Fig. 1a). For the blurring kernel we used a uniform 9×9 blur, a uniform 15×15 blur and a 15 pixel straight motion blur at a $30°$ angle. For this image and blur sizes the valid part of the convolution is the 248×248 (for 9×9 blur) or 242×242 (for 15×15 blur) central region and the data simulated with Matlab's conv2 function using the 'valid' option are of this size.

To assess the quality of our proposed non-circulant reconstruction method we compared it to reconstructions from

1) a purely circulant model (2) without data pre-processing,
2) a purely circulant model (2) where we used data pre-processing with boundary replication combined with edge tapering,
3) and reflexive end conditions and DCT (5).

When using the non-circulant model in (8) or the purely circulant model with data pre-processing in (4), the reconstructed images have original 256×256 size and then they are truncated to match the data size. When reconstructing with the purely circulant model without pre-processing in (2) or with the DCT method in (5), the reconstructed image has the same size as the data, so no truncation is necessary.

For our experiments, we used three types of regularizers, namely isotropic and anisotropic Total Variation, and l_1 norm of the coefficients of the undecimated 2-level Haar wavelet transform, excluding the approximation level.

We performed three sets of experiments with different levels of Blurred Signal-to-Noise Ratio (BSNR $\triangleq$
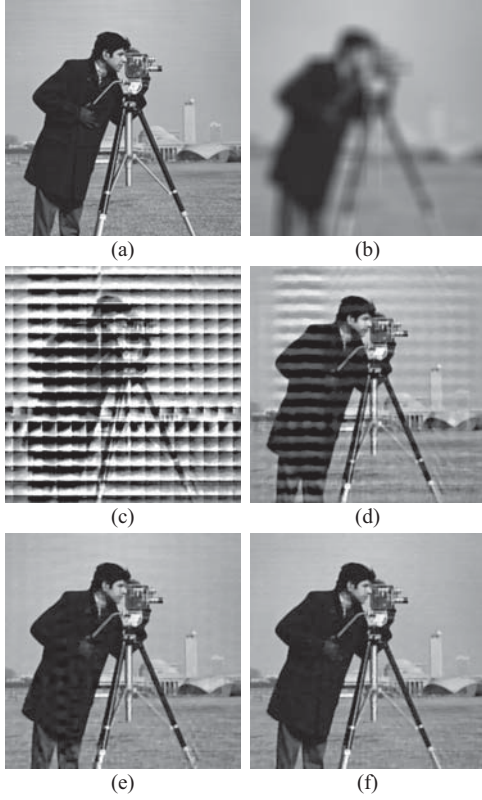
Fig. 1. Experiment 4 using 15×15 uniform blur: restoration results using isotropic TV regularization. (a) Cameraman true image (242×242), scaled to the range $[0, 1]$. (b) Blurred and noisy image from MATLAB "valid" conv2 operation. (c) Restored image from purely circulant model (2). (d) Restored image from purely circulant model with data pre-processing (4). (e) Restored image using reflexive boundary conditions and DCT (5). (f) Restored image from proposed noncirculant model (8).

$10 \log_{10} \left(\mathrm{var}(\mathbf{Ax}) / \sigma^2 \right)$ [3] and we chose the regularization parameter λ to achieve low Normalized Root Mean Square (NRMS) error for a given BSNR level. The experimental setup is as follows

1) 20 dB BSNR $(\sigma^2 = 5.03 \times 10^{-4})$ with $\lambda = 2^{-10}$,
2) 30 dB BSNR $(\sigma^2 = 5.03 \times 10^{-5})$ with $\lambda = 2^{-12}$,
3) 40 dB BSNR $(\sigma^2 = 5.03 \times 10^{-6})$ with $\lambda = 2^{-15}$,
4) 50 dB BSNR $(\sigma^2 = 5.03 \times 10^{-7})$ with $\lambda = 2^{-17}$,

where σ^2 is the Gaussian noise variance. We quantified the reconstruction quality by computing the Improvement in Signal-to-Noise Ratio (ISNR) between the restored image $\hat{\mathbf{x}}$ and the true image $\mathbf{x}_{\mathrm{true}}$. The ISNR was measured in the 248×248 or 242×242 (depending on the blur size) central region that corresponds to the valid part of the convolution, since this was the actual size of the input data vector $\mathbf{y}$. Any extrapolated values (when reconstructing with the non-circulant model) are not used in the ISNR calculations and not shown in any of the figures.

Fig. 1 shows the restored images from experiment 4 (50dB BSNR) with uniform 15×15 blur using isotropic TV regularization. Fig. 1c shows that the restored image using the purely circulant model in (2) exhibits severe ringing artifacts, similar to those in the results of [11], that are not confined to the boundaries of the image. When data pre-processing is used, the ringing artifacts are significantly reduced, but not completely

removed (Fig. 1d). These artifacts can be more severe and lead to more degradation when the strength of the regularizer decreases as seen in the quantitative results in Table I. When reflexive boundary conditions and the DCT are used, the artifacts are again significantly reduced but clearly visible in high BSNR cases. Finally, we see that the reconstructed images from the proposed non-circulant model in (8) are free of any ringing artifacts and the reconstructed image closely resembles the true noise-free image (Fig. 1f).

Fig. 2 shows the restored images from experiment 2 (30dB BSNR) with 15 pixel straight motion blur using isotropic TV regularization. Fig. 2c shows that the restored image using the purely circulant model in (2) exhibits severe ringing artifacts, similar to those in Fig. 1c. When data pre-processing is used (Fig. 2d) the ringing artifacts are reduced and are less obvious compared to Fig. 1d because of the high regularization parameter for low BSNR. When reflexive boundary conditions and the DCT are used (Fig. 2e) the method fails to produce accurate results since the PSF is non-symmetric [11], [14]. Finally, we see that the reconstructed images from the proposed non-circulant model in (8) are free of any ringing artifacts and the reconstructed image closely resembles the true noise-free image (Fig. 2f).

Fig. 3 shows the restored images from experiment 1 (20dB BSNR) with with uniform 9×9 blur using isotropic TV regularization. The restored image using the purely circulant model (Fig. 3c) exhibits severe ringing artifacts as in the previous cases. However, the reconstructed images using data pre-processing (Fig. 3d) and using reflexive boundary conditions and DCT (Fig. 3e) look very similar to the reconstructed images from the non-circulant model (Fig. 3f). This happens because the strong regularizer used to suppress noise oversmooths the image and thus the reconstruction artifacts are smoothed out as well making the images look almost identical.

Table I gives the quantitative results in terms ISNR for all experiments. In all cases, the ISNR of reconstructions based on the non-circulant model are higher than that of the reconstructions based on the purely circulant model without data pre-processing. In the case of reconstruction with the circulant model and data preprocessing or with reflexive boundary conditions and DCT, we can see that for lower SNR, where stronger regularization is required, the smoothing of the regularizer can suppress the ringing artifacts leading to reconstruction ISNR comparable to that of the proposed non-circulant model reconstruction. However, in higher SNR regimes, where less smoothing is required, the artifacts from the circulant model reconstruction become more prominent and the non-circulant reconstruction shows significantly improved reconstruction quality.

B. Convergence Speed Comparison

For the convergence speed experiments we used the setting from Experiment 3 (*i.e.*, 40 dB BSNR, $\sigma^2 = 5.03 \times 10^{-6}$ with $\lambda = 2^{-15}$) with 9×9 uniform blur. We compared the convergence speed of the following algorithms that were all implemented in Matlab.

1) **NCG**-L with L line-search sub-iterations [15];

TABLE I

ISNR IN dB FOR ALL EXPERIMENTS. THE * DENOTES RECONSTRUCTION USING THE PURELY CIRCULANT MODEL WITH DATA PRE-PROCESSING (4).
FOR THE CASE OF MOTION BLUR, THE RESULTS FOR REFLEXIVE END CONDITIONS ARE NOT PRESENTED SINCE THE METHOD IS
NOT APPROPRIATE FOR NONSYMMETRIC PSF

Blur Type	9×9 Unif.				15×15 Unif.				15 Pix. Mot.			
Reg. Type	Non-Circ.	Circ.	Circ.*	Refl.	Non-Circ.	Circ.	Circ.*	Refl.	Non-Circ.	Circ.	Circ.*	Refl.
Experiment 1: BSNR = 20 dB												
Wavelet l_1	**3.7**	−0.9	3.6	3.6	**3.4**	−2.7	3.3	**3.4**	**4.1**	−1.3	**4.1**	N/A
TV$_{\text{ANISO}}$	**3.5**	−1.2	3.4	3.4	**3.7**	−2.9	3.6	**3.7**	**4.3**	−1.5	**4.3**	N/A
TV$_{\text{ISO}}$	**3.6**	−1.4	**3.6**	3.4	**4.0**	−3.1	3.9	3.7	**4.5**	−1.7	4.4	N/A
Experiment 2: BSNR = 30 dB												
Wavelet l_1	**5.7**	-3.3	5.6	**5.7**	**4.9**	−5.3	4.7	4.8	**7.1**	−4.4	6.7	N/A
TV$_{\text{ANISO}}$	**5.4**	-3.5	**5.4**	**5.4**	**5.1**	−5.6	4.8	**5.1**	**7.1**	−4.6	6.7	N/A
TV$_{\text{ISO}}$	**5.8**	-3.7	5.7	5.4	**5.4**	−5.8	5.1	5.1	**7.4**	−4.8	6.8	N/A
Experiment 3: BSNR = 40 dB												
Wavelet l_1	**8.6**	-8.1	7.9	8.2	**7.2**	−12.8	6.4	7.1	**11.6**	−9.2	8.8	N/A
TV$_{\text{ANISO}}$	**8.4**	-8.4	7.5	7.9	**7.2**	−13.5	6.2	6.9	**11.4**	−10.2	8.1	N/A
TV$_{\text{ISO}}$	**8.3**	-8.6	7.3	7.9	**7.4**	−13.8	6.3	6.9	**11.4**	−10.6	7.7	N/A
Experiment 4: BSNR = 50 dB												
Wavelet l_1	**11.8**	-10.5	9.0	10.6	**9.8**	−15.0	6.4	9.3	**15.7**	−13.2	8.1	N/A
TV$_{\text{ANISO}}$	**11.5**	-10.4	8.3	10.1	**9.7**	−15.2	3.1	9.3	**15.0**	−13.5	6.6	N/A
TV$_{\text{ISO}}$	**11.6**	-10.6	8.2	10.1	**9.9**	−15.3	2.4	9.3	**14.7**	−13.8	5.8	N/A

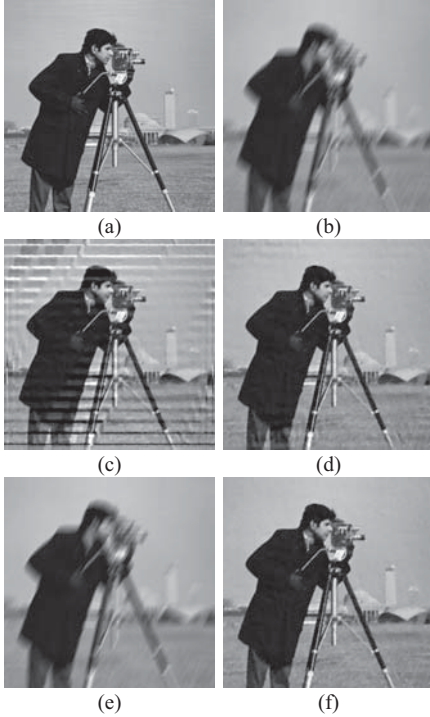Fig. 2. Experiment 3 using 15 pixel motion blur: restoration results using isotropic TV regularization. (a) Cameraman true image (242×242), scaled to the range $[0, 1]$. (b) Blurred and noisy image from MATLAB'S 'valid' conv2 operation. (c) Restored image from purely circulant model (2). (d) Restored image from purely circulant model with data pre-processing (4). (e) Restored image using reflexive boundary conditions and DCT (5). For this case, the results are inaccurate since this method is not applicable for non-symmetric PSF [11], [14]. Also the image had to be scaled to fit the dynamic range of the rest of the reconstructed images. (f) Restored image from proposed non-circulant model (8).

Fig. 3. Experiment 1 using 9×9 uniform blur: restoration results using isotropic TV regularization. (a) Cameraman true image (248×248), scaled to the range $[0, 1]$. (b) Blurred and noisy image from MATLAB'S 'valid' conv2 operation. (c) Restored image from purely circulant model (2). (d) Restored image from purely circulant model with data pre-processing (4). (e) Restored image using reflexive boundary conditions and DCT (5). (f) Restored image from proposed noncirculant model (8).

2) **ISTA** [7];
3) **MFISTA**-M with M sub-iterations of the Chambolle-type algorithm [7, Eq. (6)–(7)];
4) **SALSA**-N-M with N CG sub-iterations and M sub-iterations of [7, Eq. (6)–(7)];
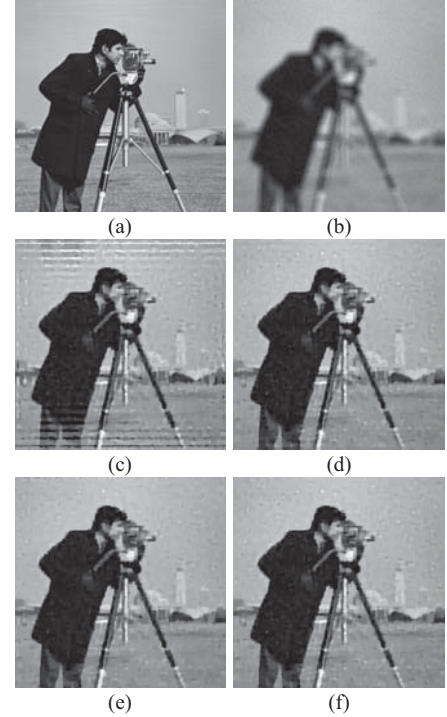5) **SB**-N with N CG sub-iterations;
6) **SB-MIL**-N with N CG sub-iterations;
7) **ADMM-P2** (proposed).

For these experiments we focused on algorithms that solve **P1** in (11). Thus, methods that use approximations in the data-fit term, *i.e.*, circulant reconstruction (with or without data

TABLE II

OUTER ITERATION TIME OF COMPETING ALGORITHMS MEASURED IN MILLISECONDS. L IS THE NUMBER OF LINE-SEARCH ITERATIONS, M IS THE NUMBER OF CHAMBOLLE-TYPE ITERATIONS, AND N IS THE NUMBER OF CG ITERATIONS

	TV	Wavelet l_1 Norm
NCG	$36 + 7L$	$95 + 15L$
ISTA	34	159
MFISTA	$17 + 19M$	$57 + 135M$
SALSA	$20 + 19M + 23N$	$22 + 135M + 24N$
SB	$39 + 31N$	$100 + 30N$
SB-MIL	$40 + 13N$	$102 + 14N$
ADMM-P2	38	90

TABLE III

TIME AND ITERATION NUMBER REQUIRED FOR EVERY ALGORITHM TO REACH -50dB ERROR COMPARED TO THE CONVERGED IMAGE $\mathbf{x}^{(\infty)}$

	TV		**Wavelet l_1**	
	Time	Iter. #	Time	Iter. #
NCG-5	56.1	298	155.6	1897
ISTA	559.1	15 000	885.0	8000
MFISTA-1	18.1	355	46.8	323
MFISTA-4	44.1	407	161.8	367
SALSA-4-1	11.7	88	15.4	71
SALSA-4-4	13.6	71	29.6	60
SALSA-4-10	24.3	74	66.2	61
SB-4	12.5	82	21.9	103
SB-MIL-4	4.9	80	10.7	100
ADMM-P2	2.8	107	5.8	109

pre-processing (2) and (4)) and reflexive end conditions with DCT (5), were not included, since they would obviously not converge to a solution of (11). **NCG** is an exception since the approximation affects the regularization and not the data-fit term.

For **NCG** we chose the rounding parameter to be $\varepsilon = 10^{-6}$, which yielded good convergence speed without compromising too much the resulting solution. We also used $L = 2$, and 5 line-search iterations. For **MFISTA**, we used $M = 1, 4, 10$, and 20 iterations of [7, Eq. (6)–(7)]. For **SALSA** we used $N = 1, 4$, and 10 CG iterations and $M = 1, 4, 10$, and 15 iterations of [7, Eq. (6)–(7)]. Finally, for **SB** and **SB-MIL** we used $N = 1, 4$, and 10 CG iterations. The CG-solvers inside **SALSA** and **SB** were terminated when $\|\mathbf{x}^{(k+1)}\|_2^2 / \alpha^{(k)} \|\mathbf{d}^{(k)}\|_2^2 < \delta$, where $\mathbf{x}^{(k+1)}$ is the new update, $\mathbf{d}^{(k)}$ is the search direction vector, $\alpha^{(k)}$ is the step size after the kth iteration, and δ is a threshold chosen as $\delta = 10^{-6}$. For all AL-based algorithms we chose the parameters μ and ν using the rules described in Section III-C, with $\lambda = 2^{-15}$, $x_{\max} = 1$, and $\mu_0 = 2^{-4}$.

All the experiments were conducted on a PC with a dual quad-core 2.6GHz Intel Xeon processor. Table II shows the per-iteration time of each algorithm measured in milliseconds.

Since the goal of the restoration problem is to find a solution to the original problem **P1**, we quantified the speed of conver-

gence as the normalized l_2 distance between the estimate at iteration k ($\mathbf{x}^{(k)}$), and the limit $\mathbf{x}^{(\infty)}$ (that represents a solution of **P1**) given by

$$\xi^{(k)} = 10 \log_{10} \left(\frac{\|\mathbf{x}^{(k)} - \mathbf{x}^{(\infty)}\|_2^2}{\|\mathbf{x}^{(\infty)}\|_2^2} \right).$$

The limit $\mathbf{x}^{(\infty)}$ was obtained by running 10^5 iterations of **MFISTA**-20, since **MFISTA** converges to a solution of **P1** and also it would not give an unfair advantage to the proposed ADMM algorithm. Since the computational load per-iteration of each algorithm varies, we evaluated $\xi^{(k)}$ as a function of both iteration number and algorithm run-time. For all algorithms we used $\mathbf{x}^{(0)} = \mathbf{A}'\mathbf{T}'\mathbf{y}$ as the initial guess. Table III shows the time and iteration number required from each algorithm to reach a -50dB error $\xi^{(k)}$. The threshold of -50dB was chosen as a benchmark since below this point there are practically no visual differences between the reconstructed image $\mathbf{x}^{(k)}$ and the converged image $\mathbf{x}^{(\infty)}$.

Fig. 4 shows the convergence rate $\xi^{(k)}$ in terms of iteration and run-time for experiments involving TV and wavelet l_1 norm regularizers. The AL based algorithms (**SALSA**, **SB**, **SB-MIL** and **ADMM-P2**) converge significantly faster than the rest for all cases.

In terms of convergence rate per iteration, **SALSA** has a slight advantage over **SB**, **SB-MIL** and **ADMM-P2** when we use enough inner Chambolle-type iterations. However, since the inner iterations can be computationally expensive, especially for the analysis l_1 regularization with a wavelet frame, its run-time is significantly higher. In terms of run-time, **ADMM-P2** is the fastest algorithm with a speed-up of about 2 times compared to the **Split-Bregman-MIL** algorithm, which is the closest competitor.

V. DISCUSSION

As seen in Fig. 1, image restoration with a circulant model can lead to severe image distortion, when applied to data with realistic boundaries. The ringing artifacts due to the discontinuity at the boundaries are not localized and even existing methods for data-preprocessing [11] either cannot fully suppress these artifacts and/or may be applicably only in specific cases like symmetric PSFs.

Our proposed method showed significant improvement of the restored images compared to the standard method (that uses an unrealistic, purely circulant blur model), when applied to data with more realistic boundaries. Our results reinforce the importance of using reconstruction models that do not make any specific assumptions about boundary extension in the data, and also show that the effects of model mismatch at the boundaries (although often ignored) can be severe in terms of image quality degradation. In the case of reflexive boundary conditions the artifacts can be significantly reduced, but the applicability of this method is only limited to symmetric PSFs [11], [14], whereas our more general model based approach can handle efficiently any type of PSF. In addition, the formulation of our model, even though similar to the one used in [11], [12], leads to a more elegant approach of the non-circulant reconstruction problem that does not require any
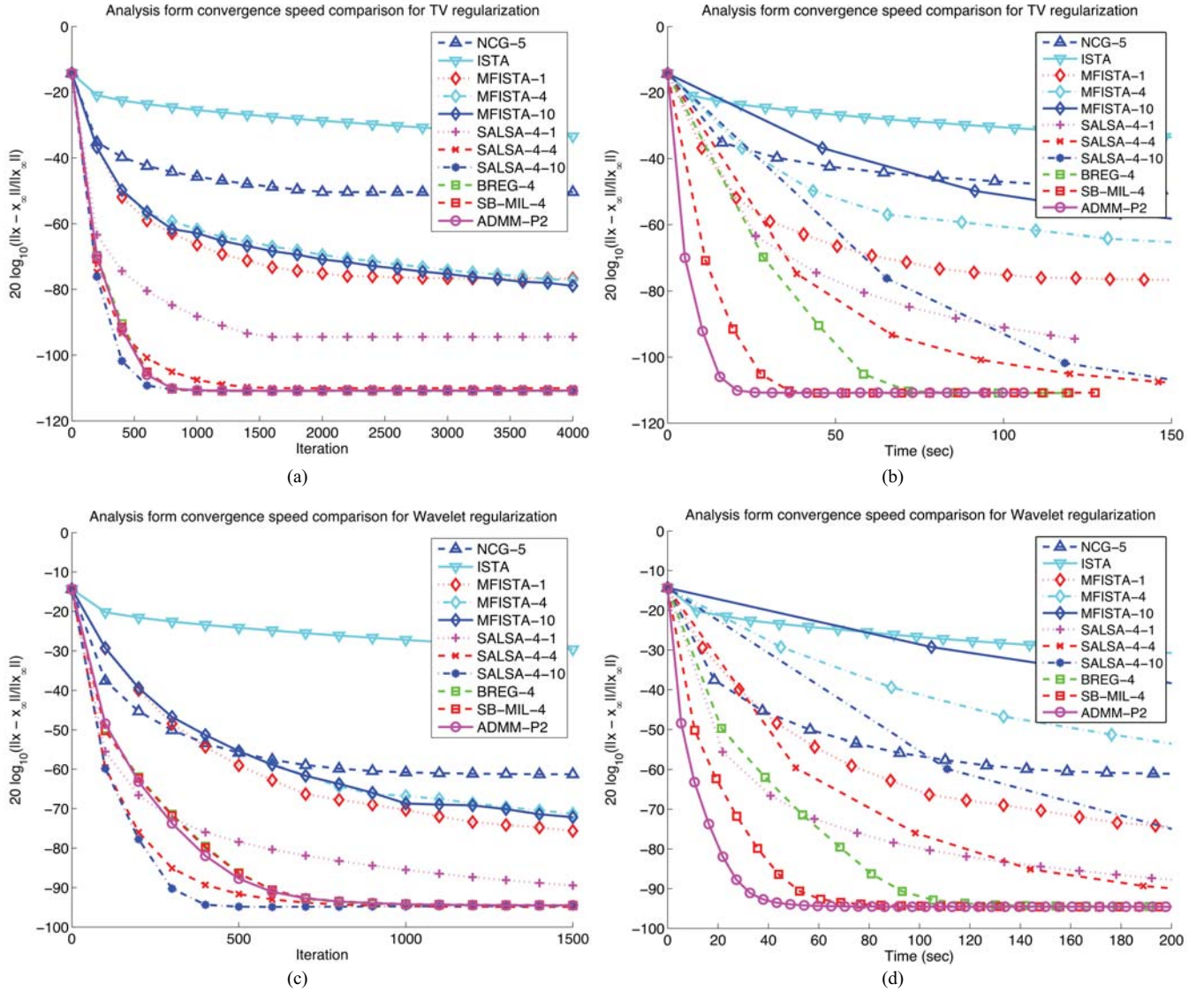
Fig. 4.   Experiment 3: algorithm convergence speed results with convergence measured in terms of drop in NRMSE in dB, $\xi^{(k)}$, between the estimated image $\hat{x}$ and the converged image $x^{(\infty)}$. Left column (a,c) is convergence per iteration and right column (b,d) is convergence in run-time. First row (a,b) is from TV regularizer and second row (c,d) is from wavelet l_1-norm regularizer.

data pre-processing and estimation of the extrapolated image boundaries.

Our proposed **ADMM-P2** algorithm, using an additional splitting variable, can efficiently handle the additional complexity introduced by the masking operator without the need for costly CG iterations for the inner sub-problems. The fact that our algorithm uses only non-iterative updates enhances its efficiency even when compared to more sophisticated approaches like **SB-MIL**. In addition the formulation is more straightforward compared to **SB-MIL** and allows for easier implementation. The caveat of tuning one additional AL parameter can be easily alleviated by using an empirical method like the one presented in Section III-C. Our **ADMM-P2** algorithm, being specifically designed for noncirculant deblurring problems, exhibits improved performance compared to existing state-of-the-art methods [6], [17], [20]. Finally, our **ADMM-P2** algorithm can be easily extended

to three-dimensional problems, where it could be potentially useful in 3D microscopy deconvolution [24], and also accommodate different likelihood models (e.g., Poisson noise) [9].

After finalizing our work, we became aware of a related recent arXiv submission [25].

REFERENCES

[1] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithm," *Phys. D: Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–68, Nov. 1992.

[2] T. F. Chan, S. Osher, and J. Shen, "The digital TV filter and nonlinear denoising," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 231–41, Feb. 2001.

[3] J. P. Oliveira, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Adaptive total variation image deblurring: A majorization-minimization approach," *Signal Process.*, vol. 89, no. 9, pp. 1683–93, 2009.

[4] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–16, Aug. 2003.

[5] C. Vonesch and M. Unser, "A fast multilevel algorithm for wavelet-regularized image restoration," *IEEE Trans. Image Process.*, vol. 18, no. 3, pp. 509–23, Mar. 2009.

[6] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–56, Sep. 2010.

[7] I. W. Selesnick and M. A. T. Figueiredo, "Signal restoration with overcomplete wavelet transforms: Comparison of analysis and synthesis priors," in *Proc. SPIE Opt. Eng.$^+$ Appl., Int. Soc. Opt. Photon.*, 2009, p. 74460D.

[8] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[9] M. A. T. Figueiredo and J. M. Bioucas-Dias, "Restoration of Poissonian images using alternating direction optimization," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3133–3145, Dec. 2010.

[10] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–695, Mar. 2011.

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

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

[13] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Prob.*, vol. 23, no. 3, pp. 947–968, Jun. 2007.

[14] M. Ng, R. Chan, and W. Tang, "A fast algorithm for deblurring models with Neumann boundary conditions," *SIAM J. Sci. Comput.*, vol. 21, no. 3, pp. 851–66, 1999.

[15] J. A. Fessler and S. D. Booth, "Conjugate-gradient preconditioning methods for shift-variant PET image reconstruction," *IEEE Trans. Image Process.*, vol. 8, no. 5, pp. 688–699, May 1999.

[16] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[17] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[18] 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. 248–2472, 2008.

[19] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing," *SIAM J. Imaging Sci.*, vol. 1, no. 1, pp. 143–168, 2008.

[20] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imaging Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[21] S. Ramani and J. A. Fessler, "Parallel MR image reconstruction using augmented Lagrangian methods," *IEEE Trans. Med. Imaging*, vol. 30, no. 3, pp. 694–706, Mar. 2011.

[22] S. Ramani and J. A. Fessler, "A splitting-based iterative algorithm for accelerated statistical X-ray CT reconstruction," *IEEE Trans. Med. Imaging*, vol. 31, no. 3, pp. 677–688, Mar. 2012.

[23] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Programm.*, vol. 55, nos. 1–3, pp. 293–318, Apr. 1992.

[24] P. Sarder and A. Nehorai, "Deconvolution methods for 3-D fluorescence microscopy images," *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 32–45, May 2006.

[25] M. S. C. Almeida and M. A. T. Figueiredo, "Deconvolving images with unknown boundaries using the alternating direction method of multipliers," to be published.

Antonios Matakos (S'04) received the Diploma degree in engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, and the M.S.E. degree from the University of Michigan, Ann Arbor, MI, USA, in 2006 and 2009, respectively, both in electrical engineering, where he is currently pursuing the Ph.D. degree in electrical engineering.

His current research interests include statistical image processing and model-based, iterative reconstruction methods for MRI.

Sathish Ramani (S'08–M'09) received the M.Sc. degree in physics from the Sri Sathya Sai Institute of Higher Learning, Puttaparthy, India, the M.Sc. degree in electrical engineering from the Indian Institute of Science, Bangalore, India, and the Ph.D. degree in electrical engineering from the Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2000, 2005, and 2009, respectively.

He was a Swiss National Science Foundation Post-Doctoral Fellow with the, Systems Group, Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI, USA, from 2009 to 2010, where he is currently a Research Fellow. His current research interests include splines, interpolation, variational methods in image processing, and optimization algorithms for biomedical imaging applications.

Dr. Ramani was a recipient of the Best Student Paper Award from the IEEE International Conference on Acoustics, Speech, and Signal Processing in 2006.

Jeffrey A. Fessler (F'06) received the B.S.E.E. degree from Purdue University, West Lafayette, IN, USA, in 1985, and the M.S.E.E. degree, the M.S. degree in statistics, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 1986, 1989, and 1990, respectively.

He is currently a Professor with the Departments of Electrical Engineering and Computer Science, Radiology, and Biomedical Engineering, University of Michigan, Ann Arbor, MI, USA, where he was an Alexander Hollaender Post-Doctoral Fellow with the Department of Energy, Division of Nuclear Medicine, from 1991 to 1992, and was an Assistant Professor of Nuclear Medicine and the Bioengineering Program from 1993 to 1995. From 1985 to 1988, he was a National Science Foundation Graduate Fellow at Stanford University. He has supervised doctoral research on PET, SPECT, X-ray CT, MRI, and optical imaging problems. His current research interests include statistical aspects of imaging problems.

Dr. Fessler was a recipient of the Francois Erbsmann Award for his IPMI93 presentation. He was an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS and the IEEE TRANSACTIONS ON MEDICAL IMAGING. He is currently an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING. He was the Co-Chair of the 1997 SPIE Conference on Image Reconstruction and Restoration, the Technical Program Co-Chair of the 2002 IEEE International Symposium on Biomedical Imaging (ISBI), and the General Chair of ISBI 2007. He was the Chair of the Steering Committee of the IEEE TRANSACTIONS ON MEDICAL IMAGING, and the Chair of the ISBI Steering Committee. He was an Associate Chair of the Department from 2006 to 2008.