SIAM J. IMAGING SCIENCES
Vol. 10, No. 2, pp. 484–507

ADMM in Krylov Subspace and Its Application to Total Variation Restoration of Spatially Variant Blur*

Joo Dong Yun† and Seungjoon Yang†

Abstract. In this paper we propose an efficient method for a convex optimization problem which involves a large nonsymmetric and non-Toeplitz matrix. The proposed method is an instantiation of the alternating direction method of multipliers applied in Krylov subspace. Our method offers significant advantages in computational speed for the convex optimization problems involved with general matrices of large size. We apply the proposed method to the restoration of spatially variant blur. The matrix representing spatially variant blur is not block circulant with circulant blocks (BCCB). Efficient implementation based on diagonalization of BCCB matrices by the discrete Fourier transform is not applicable for spatially variant blur. Since the proposed method can efficiently work with general matrices, the restoration of spatially variant blur is a good application of our method. Experimental results for total variation restoration of spatially variant blur show that the proposed method provides meaningful solutions in a short time.

Key words. ADMM, Krylov subspace methods, image restoration, space variant blur, total variation

AMS subject classifications. 65F10, 65F22, 68U10, 90C25

DOI. 10.1137/16M1083505

1. Introduction.

Many interesting problems in image processing involve finding a solution of a system of linear equations given by $\mathbf{g} = \mathbf{A}\mathbf{f}$, where $\mathbf{A} \in \mathbb{R}^{d\times d}$ and $\mathbf{g}, \mathbf{f} \in \mathbb{R}^d$. When the problem is ill-posed, a regularized solution can be found by reducing the solution space with constraints that reflect our prior knowledge about the solution. With convex constraints that reflect our prior knowledge, the problem can be written as a convex optimization. The alternating direction method of multipliers (ADMM) [11, 26, 27] is a convex optimization tool, which recently received considerable attention for its ease of incorporating diverse convex constraints to the problem, ease of implementation, and fast computational speed.

The Krylov subspace method [44, 45] is a projection method which restricts the solution space of the problem $\mathbf{g} = \mathbf{A}\mathbf{f}$ to the subspace spanned by the nth order Krylov sequence $\mathbf{x}$, $\mathbf{A}\mathbf{x}$, $\mathbf{A}^2\mathbf{x}$, $\ldots$, $\mathbf{A}^{n-1}\mathbf{x}$ constructed from a vector $\mathbf{x} \in \mathbb{R}^d$. By reducing the solution space to a subspace of small dimensionality, a solution can be found efficiently. The orthonormal basis of the Krylov subspace can be found by the Lanczos bidiagonalization when $\mathbf{A}$ is symmetric and by the Arnoldi process when $\mathbf{A}$ is nonsymmetric. Examples of the Krylov subspace method are minimum residual (MINRES) [41] and generalized minimum residual (GMRES)

*Received by the editors July 6, 2016; accepted for publication (in revised form) December 16, 2016; published electronically April 11, 2017.
http://www.siam.org/journals/siims/10-2/M108350.html
Funding: This research was partially supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2016R1D1A1B01016041).
†School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, Republic of Korea (joodong1@gmail.com, syang@unist.ac.kr).

[46], which find solutions of systems of linear equations with a symmetric and a nonsymmetric $\mathbf{A}$, respectively, and Arnoldi–Tikhonov [15], which finds a solution of a Tikhonov regularization problem with nonsymmetric $\mathbf{A}$.

Image restoration, or deblurring, is an inverse problem of finding a clear image from an observed image which is blurry and noisy. The image acquisition process, including the introduction of blur and additive noise, can be modeled by a system of linear equations [5, 6]. The restoration problem is to find a solution of the system of linear equations, which is usually ill-posed. A meaningful solution is obtained by adopting regularizations based on prior knowledge of the solution [24]. Various types of convex regularizations such as total variation (TV) and ℓ_1 and ℓ_2 norms are used to penalize the roughness of the solution based on our prior knowledge. The restoration problems can be solved in various types of convex optimization frameworks [1, 2, 7, 8, 9].

Images captured by real world imaging systems show blur due to imperfect optical systems or acquisition conditions. Blur in images is often spatially variant in the sense that the shape of the point spread function (PSF) changes depending on pixel locations in an image. Optical aberrations, inaccurate focusing, camera shake, and movement of objects are all possible sources of blur. Blur introduced by an optical aberration, such as spherical aberration, astigmatism, coma, or field curvature of a lens system, depends on pixel locations in an image sensor [29, 34, 37, 38, 47]. Objects outside the depth-of-field show blur differently from objects in focus [47, 53]. Objects farther away show an amount of blur different from that of objects closer to a camera shaken during exposure [35]. Objects moving faster show more severe blur than the background, which remains still when exposure time is long [3]. Degradation by the spatially variant blur is modeled by a system of linear equations with a nonsymmetric and non-Toeplitz matrix. In the case of spatially invariant blur, the degradation is modeled by a system of linear equations with a block circulant matrix with circulant blocks (BCCB) [1, 2, 4, 18, 33]. The BCCB matrix is a type of Toeplitz matrix which can be diagonalized by the discrete Fourier transform (DFT) [32]. Efficient restoration methods based on the diagonalization by the DFT are available for the spatially invariant blur [2, 17, 31]. In the case of spatially variant blur, efficient implementation based on the diagonalization by the DFT is not available. The solution of the system of linear equation with a nonsymmetric and non-Toeplitz matrix is found by iterative methods, which usually are computationally expensive.

This work presents an efficient convex optimization method based on ADMM [12] and a Krylov subspace method [44] to find a regularized solution of the problem $\mathbf{Af} = \mathbf{g}$, where $\mathbf{A}$ is nonsymmetric and non-Toeplitz. The problem is formulated as a general convex optimization problem with convex constraints that ADMM solves. The solution space of the convex optimization problem is restricted to the nth order Krylov subspace. The basis of the Krylov subspace is found by the Arnoldi process. Subproblems involved in ADMM iterations are considerably simplified by the relationship given by the Arnoldi process. The proposed ADMM with Krylov subspace (KADMM) is applied to the TV restoration of spatially variant blur. Performance is evaluated with spatially variant Gaussian blur and spatially variant rotational blur. Performance of the proposed method is compared to that of Nesterov's algorithm (NESTA) [8] and the primal-dual proximal algorithm (PA) [21], which find the solution of the convex optimization problem in the original solution space. The proposed method provides

similar results to NESTA and PA in the mean square error (MSE), even though the solution space is restricted to the nth order Krylov subspace. The proposed method has significantly lower computational complexity in terms of the number of multiplications by a large matrix and also in terms of the wall-clock time.

Spatially variant blur can be modeled by piecewise spatially invariant blur [39, 36], for which efficient implementation by the DFT is applicable. In [40], the piecewise invariant blur model is applied to the restoration of spatially variant Gaussian blur. The use of the piecewise invariant blur model is to accelerate the computationally dominant operations. The quality of the restored images will depend on how well the model approximates the spatially variant blur, and also on how rank-deficient the matrix $\mathbf{A}$ becomes after the modeling. The aim of the proposed method is to reduce the solution space for efficient computation without approximating the spatially variant blur.

The main contribution of this work is to provide an effective convex optimization framework to solve an inverse problem with various regularizations for a system represented by a nonsymmetric non-Toeplitz matrix. By restricting the solution space to the Krylov subspace, a meaningful inexact solution can be found efficiently by the ADMM, for which various constraints and regularizations are readily available. For image restoration, the proposed framework can be applied to fast restoration of spatially variant blur.

This paper is organized as follows. In section 2, ADMM, Krylov subspace methods, and the restoration of spatially variant blur are introduced. In section 3, the general form of the problem that we consider and the proposed method, KADMM, are presented. In section 4, KADMM is applied to the TV restoration of spatially variant blur. Experimental results and discussions are provided in section 5. Section 6 concludes the paper.

2. Basic ingredients.

2.1. Alternating direction method of multipliers.

The solution of optimization problems in the form of

$$\underset{\mathbf{u}\in\mathbb{R}^d}{\text{minimize}} \sum_{j=1}^{J} c_j(\mathbf{B}_j\mathbf{u}) \tag{1}$$

can be found by the ADMM, where $\mathbf{B}_j$'s are arbitrary matrices and c_j's are convex, closed, and proper functions for $j = 1, \ldots, J$. The variable splitting with $\mathbf{v}_j = \mathbf{B}_j\mathbf{u}$ for $j \in [1, J]$ and the augmented Lagrangian method is the key idea of the ADMM. The ADMM algorithm is given in Algorithm 1, where

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_J \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_J \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_J \end{bmatrix}. \tag{2}$$

A proximal operator of a function c is defined as

$$\text{prox}_c(\mathbf{x}) = \arg\min_{\mathbf{y}} c(\mathbf{y}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2. \tag{3}$$

The proximal operators are well defined if the function c is proper and convex [20]. For various choices of the function h, corresponding proximal operators prox_c are available in closed form.

Algorithm 1 ADMM.

repeat
  $\mathbf{u} \leftarrow \arg\min_{\mathbf{u}} \ ||\mathbf{Bu} - \mathbf{v} - \mathbf{d}||_2^2$
  for $j = 1, \ldots, J$ **do**
    $\mathbf{v}_j \leftarrow \text{prox}_{c_j/\mu}(\mathbf{B}_j\mathbf{u} - \mathbf{d}_j)$
    $\mathbf{d}_j \leftarrow \mathbf{d}_j - (\mathbf{B}_j\mathbf{u} - \mathbf{v}_j)$
  end for
until stopping criterion is satisfied.

2.2. Krylov subspace methods. Krylov subspace methods consider an inverse problem of the form $\mathbf{g} = \mathbf{Af}$, where $\mathbf{A} \in \mathbb{R}^{d\times d}$ and $\mathbf{f}, \mathbf{g} \in \mathbb{R}^d$. A Krylov subspace of order n is generated by a matrix $\mathbf{A}$ and a residual vector $\mathbf{r} := \mathbf{g} - \mathbf{Af}_0$ as follows:

$$\mathcal{K}_n(\mathbf{A}, \mathbf{r}) = \text{span}\{\mathbf{r}, \mathbf{Ar}, \ldots, \mathbf{A}^{n-1}\mathbf{r}\}, \tag{4}$$

where $\mathbf{f}_0 \in \mathbb{R}^d$ is an initial solution. Krylov subspace methods approximate the solution vector $\mathbf{f} \in \mathbb{R}^d$ by assuming

$$\mathbf{f} - \mathbf{f}_0 \in \mathcal{K}_n(\mathbf{A}, \mathbf{r}). \tag{5}$$

Then the solution can be written as

$$\mathbf{f} = \mathbf{f}_0 + \mathbf{Q}_n\boldsymbol{\alpha}, \tag{6}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^n$ is the weight for the linear combination and $\mathbf{Q}_n := [\mathbf{q}_1, \ldots, \mathbf{q}_n] \in \mathbb{R}^{d\times n}$ is a matrix with the basis $\{\mathbf{q}_1, \ldots, \mathbf{q}_n\}$ of the nth order Krylov subspace $\mathcal{K}_n(\mathbf{A}, \mathbf{r})$. For nonsymmetric $\mathbf{A}$, the basis can be computed by the Arnoldi process in Algorithm 2. The Arnoldi process also provides the following relationship:

$$\mathbf{AQ}_n = \mathbf{Q}_{n+1}\mathbf{H}_n, \tag{7}$$

where $\mathbf{H}_n \in \mathbb{R}^{(n+1)\times n}$ is an upper Hessenberg matrix. The problem of finding $\mathbf{f} \in \mathbb{R}^d$ becomes a problem of finding the weight $\boldsymbol{\alpha} \in \mathbb{R}^n$ of the linear combination in (6). For example, finding $\mathbf{f}$ with minimum residual is equivalent to finding $\boldsymbol{\alpha}$ with

$$\|\mathbf{g} - \mathbf{Af}\| = \|\mathbf{r} - \mathbf{AQ}_n\boldsymbol{\alpha}\| = \|\beta\mathbf{e}_0 - \mathbf{H}_n\boldsymbol{\alpha}\|, \tag{8}$$

where $\beta = \|\mathbf{r}\|$ and $\mathbf{e}_0$ is the vector with zero elements except the first one. With $n \ll d$, the size of the problem can be reduced significantly. Krylov subspace methods have been applied to the least squares problem [41, 46] and the Tikhonov regularization problem [15]. Those problems are subsets of convex optimization. As an extension, we propose a Krylov subspace method which can be applied to a more general framework of convex optimization.

Algorithm 2 Arnoldi($\mathbf{A}$, $\mathbf{g}$, n, $\mathbf{f}_0$).

1: $\mathbf{r}_0 \leftarrow \mathbf{g} - \mathbf{A}\mathbf{f}_0$
2: $\mathbf{q}_1 \leftarrow \mathbf{r}_0/\|\mathbf{r}_0\|_2$
3: **for** $i = 1, \ldots, n$ **do**
4: $\quad \mathbf{v} \leftarrow \mathbf{A}\mathbf{q}_i$
5: $\quad$ **for** $j = 1, \ldots, i$ **do**
6: $\quad\quad h_{j,i} \leftarrow \mathbf{q}_j^{\mathrm{T}}\mathbf{v}$
7: $\quad\quad \mathbf{v} \leftarrow \mathbf{v} - h_{j,i}\mathbf{q}_j$
8: $\quad$ **end for**
9: $\quad h_{i+1,i} \leftarrow \|\mathbf{v}\|_2$
10: $\quad \mathbf{q}_{i+1} \leftarrow \mathbf{v}/h_{i+1,i}$
11: **end for**

3. ADMM in Krylov subspace (KADMM).

3.1. Problem formulation.

Consider an optimization problem in the form of

$$\underset{\mathbf{u}\in\mathbb{R}^d}{\text{minimize}}\; h_1(\mathbf{A}\mathbf{u}) + \sum_{j=2}^{J} h_j(\mathbf{G}_j\mathbf{u}), \tag{9}$$

where each function $h_j : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is proper, closed, and convex, and each matrix $\mathbf{G}_j \in \mathbb{R}^{d\times d}$ for all j. The matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$ can be nonsymmetric and non-Toeplitz in general.

One can apply ADMM directly to solve the problem in (9) with $\mathbf{B}_1 = \mathbf{A}$ in Algorithm 1. If the involved matrix $\mathbf{A}$ is BCCB, efficient methods based on the DFT are available to solve the subproblems of the ADMM. However, if the matrix is not BCCB, the minimization problem inside line 2 of Algorithm 1 cannot be accelerated by the efficient DFT implementation. In general, solving the inverse problems of unstructured matrices such as nonsymmetric and non-Toeplitz matrices is more difficult when the dimensionality of the involved matrix is huge. To overcome those difficulties associated with directly applying the ADMM, we propose to restrict the solution space to the nth order Krylov subspace when finding the solution of the convex optimization problem with ADMM.

3.2. Application of Krylov subspace method.

In order to find a solution in an efficient manner, we restrict the solution space of the optimization problem in (9) to the nth order Krylov subspace. An inexact solution is found in $\mathbf{u}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r})$ instead of in $\mathbb{R}^d$, where $\mathbf{u}_0$ is the initial solution for the solution $\mathbf{u}$, and $\mathbf{r}$ is the residual, $\mathbf{r} = \mathbf{g} - \mathbf{A}\mathbf{u}_0$. The dimension of the solution space is significantly reduced with a proper choice of $n \ll d$. Applying the Krylov subspace method to (9), we have

$$\underset{\mathbf{u}\in\mathbf{u}_0+\mathcal{K}_n(\mathbf{A},\mathbf{r})}{\text{minimize}}\; h_1(\mathbf{A}\mathbf{u}) + \sum_{j=2}^{J} h_j(\mathbf{G}_j\mathbf{u}). \tag{10}$$

The basis of the Krylov subspace of degree n can be efficiently computed by the Arnoldi process as in Algorithm 2. Since the solution vector projected in Krylov subspace $\mathbf{u}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r})$ can

be written in the form of (6), the objective in (10) becomes

$$h_1(\mathbf{A}\mathbf{Q}_n\boldsymbol{\alpha} + \mathbf{A}\mathbf{u}_0) + \sum_{j=2}^{J} h_j(\mathbf{G}_j\mathbf{Q}_n\boldsymbol{\alpha} + \mathbf{G}_j\mathbf{u}_0), \tag{11}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\mathbf{Q}_n \in \mathbb{R}^{d\times n}$. The optimization problem in (11) can be rewritten as

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\text{minimize}}\ \hat{h}_1(\mathbf{A}\mathbf{Q}_n\boldsymbol{\alpha}) + \sum_{j=2}^{J} \hat{h}_j(\mathbf{G}_j\mathbf{Q}_n\boldsymbol{\alpha}), \tag{12}$$

where $\hat{h}_1(\mathbf{y}) = h_1(\mathbf{y} + \mathbf{A}\mathbf{u}_0)$ and $\hat{h}_j(\mathbf{y}) = h_j(\mathbf{y} + \mathbf{G}_j\mathbf{u}_0)$ for $j = 2, \ldots, J$. Note that the dimension of the search space n in (12) is significantly smaller than the original search space d in (10). In the following sections, it will be shown that this reduced optimization problem can be solved by ADMM very efficiently within a small dimension n.

3.3. Application of ADMM. The ADMM is applied to find the solution of (12). The proposed method of applying the ADMM in the Krylov subspace is denoted by KADMM in the remainder of the paper. The algorithm is shown in Algorithm 3.

At first, let us investigate the applicability of ADMM for (12). The application of ADMM is justified when the following hold:

1. Each function $\hat{h}_j : \mathbb{R}^d \to \mathbb{R}$ is convex, closed, and proper.
2. A numerical formula for the proximal operator of each $\hat{h}_j$ is derived.
3. The least squares problem in line 2 of Algorithm 1 is computable, or the dimension of the problem is reduced.

The first condition is a general assumption for ADMM. The second condition is needed to calculate line 4 of Algorithm 1 for the reduced problem (12). Finally, the third condition is related to line 2 of Algorithm 1. These items will be justified in the following.

Lemma 1. *Each $\hat{h}_j : \mathbb{R}^d \to \mathbb{R}$ in (12) is convex, closed, and proper.*

Proof. By definition, $\hat{h}_j$ in (12) is a translated function from h_j, which is convex, closed, and proper for $j = 1, \ldots, J$. It is trivial to show that translation preserves those properties [10, 13], so $\hat{h}_j$ is also convex, closed, and proper. ∎

Lemma 2. *The proximal operator of $\hat{h}_j$ in (12) is given by*

$$\operatorname{prox}_{\hat{h}_1/\mu}(\mathbf{y}) = -\mathbf{A}\mathbf{u}_0 + \operatorname{prox}_{h_1/\mu}(\mathbf{y} + \mathbf{A}\mathbf{u}_0) \tag{13}$$

and

$$\operatorname{prox}_{\hat{h}_j/\mu}(\mathbf{y}) = -\mathbf{G}_j\mathbf{u}_0 + \operatorname{prox}_{h_j/\mu}(\mathbf{y} + \mathbf{G}_j\mathbf{u}_0) \tag{14}$$

for $j = 2, \ldots, J$.

Proof. Proximal operators of $\hat{h}_j$ should be calculated in order to apply ADMM to (12). The translation property of proximity operators [19] says that if $\psi(\mathbf{x}) = \varphi(\mathbf{x} - \mathbf{z})$, then

$$\operatorname{prox}_{\psi}(\mathbf{y}) = \mathbf{z} + \operatorname{prox}_{\varphi}(\mathbf{y} - \mathbf{z}). \tag{15}$$

Since $\hat{h}_1(\mathbf{y}) = h_1(\mathbf{y}+\mathbf{A}\mathbf{u}_0)$ and $\hat{h}_j(\mathbf{y}) = h_j(\mathbf{y}+\mathbf{G}_j\mathbf{u}_0)$ for $j = 2, \dots, J$, the proximal operators of $\hat{h}_j$ can be written by a function of the proximal operators of h_j, as in (13) and (14). ∎

Now we can say that if the proximal operators of h_j are computable, the proximal operators of $\hat{h}_j$ are also computable. This will be used in line 4 of Algorithm 1 when we apply ADMM.

The ADMM framework in Algorithm 1 is applicable by setting $\mathbf{B}_1 \leftarrow \mathbf{A}\mathbf{Q}_n$, $\mathbf{u} \leftarrow \boldsymbol{\alpha}$, $c_j \leftarrow \hat{h}_j$, and $\mathbf{B}_j \leftarrow \mathbf{G}_j\mathbf{Q}_n$ for $j = 2, \dots, J$. We apply ADMM to the optimization problem in (12), which leads to the KADMM algorithm in Algorithm 3. The proximal operators (13) and (14) are computed in lines 7 and 10 of Algorithm 3, respectively.

Proposition 3. *Under the Krylov subspace assumption, i.e.,* $\mathbf{u} \in \mathbf{u}_0 + \mathcal{K}_n(\mathbf{A}, \mathbf{r})$*, the least squares problem on line* 2 *of Algorithm* 1 *is reduced to*

$$\underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\text{minimize}} \left\| \begin{bmatrix} \mathbf{H}_n \\ \bar{\mathbf{G}}_2 \\ \vdots \\ \bar{\mathbf{G}}_J \end{bmatrix} \boldsymbol{\alpha} - \begin{bmatrix} \mathbf{Q}_{n+1}^T(\mathbf{v}_1 + \mathbf{d}_1) \\ \mathbf{Q}_n^T(\mathbf{v}_2 + \mathbf{d}_2) \\ \vdots \\ \mathbf{Q}_n^T(\mathbf{v}_J + \mathbf{d}_J) \end{bmatrix} \right\|_2^2, \tag{16}$$

where $\bar{\mathbf{G}}_j := \mathbf{Q}_n^T\mathbf{G}_j\mathbf{Q}_n \in \mathbb{R}^{n\times n}$ *for* $j = 2, \dots, J$.

Proof. Applying Algorithm 1 to (12), the function in the least squares problem on line 2 can be written as

$$\left\| \begin{bmatrix} \mathbf{A}\mathbf{Q}_n \\ \mathbf{G}_2\mathbf{Q}_n \\ \vdots \\ \mathbf{G}_J\mathbf{Q}_n \end{bmatrix} \boldsymbol{\alpha} - \begin{bmatrix} \mathbf{v}_1 + \mathbf{d}_1 \\ \mathbf{v}_2 + \mathbf{d}_2 \\ \vdots \\ \mathbf{v}_J + \mathbf{d}_J \end{bmatrix} \right\|_2^2, \tag{17}$$

where the matrix $\mathbf{Q}_n$ in the first row is the product of the Arnoldi process for $\mathbf{A}$, which has the property of $\mathbf{A}\mathbf{Q}_n = \mathbf{Q}_{n+1}\mathbf{H}_n$ as in (7). Note that $\mathbf{Q}_n \in \mathbb{R}^{d\times n}$ is stacked by orthonormal columns, satisfying $\mathbf{Q}_n^T\mathbf{Q}_n = \mathbf{I}_n$ for any n. For a given vector $\mathbf{x} \in \mathbb{R}^d$, $\|\mathbf{Q}_n\mathbf{x}\|_2^2 = \|\mathbf{x}\|_2^2$. Therefore, the matrix size involved with the problem (17) can be reduced as follows:

$$\left\| \begin{bmatrix} \mathbf{Q}_{n+1}^T(\mathbf{Q}_{n+1}\mathbf{H}_n) \\ \mathbf{Q}_n^T\mathbf{G}_2\mathbf{Q}_n \\ \vdots \\ \mathbf{Q}_n^T\mathbf{G}_J\mathbf{Q}_n \end{bmatrix} \boldsymbol{\alpha} - \begin{bmatrix} \mathbf{Q}_{n+1}^T(\mathbf{v}_1 + \mathbf{d}_1) \\ \mathbf{Q}_n^T(\mathbf{v}_2 + \mathbf{d}_2) \\ \vdots \\ \mathbf{Q}_n^T(\mathbf{v}_J + \mathbf{d}_J) \end{bmatrix} \right\|_2^2. \tag{18}$$

Since $\mathbf{Q}_{n+1}^T\mathbf{Q}_{n+1} = \mathbf{I}_{n+1}$, (18) is equivalent to (16). ∎

By the relationship (7) given by the Krylov subspace method, the least squares problem in line 2 of Algorithm 1 can be reduced to a least squares problem of smaller size. This is reflected in line 6 of Algorithm 3.

The matrix involved with the least squares problem in (16) has size $(nJ + 1) \times n$. With a fixed $n \ll d$, the size of the matrix involved with the least squares problem in (16) is much smaller than the matrix originally given by direct application of the ADMM, whose size is

$dJ \times d$. The size of the least squares problem in (16) does not depend on the dimension of the original solution space d, but only on the parameter n. Since $\bar{\mathbf{G}}_j$ has size $n \times n$, it can be stored in a small amount of memory before the outer loop of ADMM starts and used throughout the entire iterations of ADMM. Additionally, if $\mathbf{G}_j$ is an identity matrix of size $d \times d$, then $\bar{\mathbf{G}}_j$ is an identity matrix of size $n \times n$. In this case, we don't have to calculate and store the matrix during computational iterations. We will show that this reduction occurs in certain formulations of the TV deblurring problem.

Proposition 4. *For a vector $\boldsymbol{\alpha} \in \mathbb{R}^n$, the operation of $\mathbf{Q}_{n+1}\mathbf{H}_n\boldsymbol{\alpha}$ is computationally faster than the operation of $\mathbf{A}\mathbf{Q}_n\boldsymbol{\alpha}$.*

Proof. Considering the sizes of the involved matrices, the operations of $\mathbf{Q}_{n+1}(\mathbf{H}_n\boldsymbol{\alpha})$ and $\mathbf{A}(\mathbf{Q}_n\boldsymbol{\alpha})$ approximately require $(2n^2+2nd) \approx O(d)$ and $(2nd+2d^2) \approx O(d^2)$ flops, respectively [28]. The above claim holds for $n \ll d$. ∎

In KADMM, the operation $\mathbf{A}$ is used only for the Arnoldi process to find the basis of the nth order Krylov subspace, and the iterations of the ADMM do not require the computationally expensive operation $\mathbf{A}$.

The KADMM is summarized in Algorithm 3. The KADMM can find an inexact solution to any problem for which the ADMM can be applied. By the Krylov subspace method, not only is the dimension of search space reduced from d to n, but the size of the least squares problem in the ADMM also is reduced greatly. The KADMM finds the solution only in the nth order Krylov subspace.

Algorithm 3 KADMM($\mathbf{A}$,$\mathbf{g}$,n,$\mathbf{u}_0$).

$\mathbf{Q}_{n+1}, \mathbf{H}_n \leftarrow \text{Arnoldi}(\mathbf{A}, \mathbf{g}, n, \mathbf{u}_0)$
for $j = 2, \ldots, J$ **do**
  $\bar{\mathbf{G}}_j \leftarrow \mathbf{Q}_n^T \mathbf{G}_j \mathbf{Q}_n \in \mathbb{R}^{n \times n}$
end for
repeat
  $$\boldsymbol{\alpha} \leftarrow \arg\min_{\boldsymbol{\alpha}} \left\| \begin{bmatrix} \mathbf{H}_n \\ \bar{\mathbf{G}}_2 \\ \vdots \\ \bar{\mathbf{G}}_J \end{bmatrix} \boldsymbol{\alpha} - \begin{bmatrix} \mathbf{Q}_{n+1}^T(\mathbf{v}_1 + \mathbf{d}_1) \\ \mathbf{Q}_n^T(\mathbf{v}_2 + \mathbf{d}_2) \\ \vdots \\ \mathbf{Q}_n^T(\mathbf{v}_J + \mathbf{d}_J) \end{bmatrix} \right\|_2^2$$
  $\mathbf{v}_1 \leftarrow -\mathbf{A}\mathbf{u}_0 + \text{prox}_{h_1/\mu}(\mathbf{Q}_{n+1}\mathbf{H}_n\boldsymbol{\alpha} - \mathbf{d}_1 + \mathbf{A}\mathbf{u}_0)$
  $\mathbf{d}_1 \leftarrow \mathbf{d}_1 - (\mathbf{Q}_{n+1}\mathbf{H}_n\boldsymbol{\alpha} - \mathbf{v}_1)$
  for $j = 2, \ldots, J$ **do**
    $\mathbf{v}_j \leftarrow -\mathbf{G}_j\mathbf{u}_0 + \text{prox}_{h_j/\mu}(\mathbf{G}_j\mathbf{Q}_n\boldsymbol{\alpha} - \mathbf{d}_j + \mathbf{G}_j\mathbf{u}_0)$
    $\mathbf{d}_j \leftarrow \mathbf{d}_j - (\mathbf{G}_j\mathbf{Q}_n\boldsymbol{\alpha} - \mathbf{v}_j)$
  end for
until stopping criterion is satisfied.
$\mathbf{u} \leftarrow \mathbf{Q}_n\boldsymbol{\alpha} + \mathbf{u}_0$

3.4. Convergence analysis. The KADMM is an application of ADMM for (12) which is a convex optimization problem derived by the Krylov subspace method. Using the projection method on Krylov subspace, the dimension of the search space is changed from image size d to optional truncation order n. Given $n \ll d$, the convergence of KADMM follows the convergence of ADMM [30, 23] of smaller optimization problems. It will be shown in numerical experiments that the practical choice of small n gives fast convergence as well.

4. TV restoration of spatially variant blur.

4.1. Restoration of spatially variant blur. The image acquisition model is given by

$$g(\mathbf{s}) = \sum_{\mathbf{t}} k(\mathbf{s}, \mathbf{t}) f(\mathbf{t}) + n(\mathbf{s}), \tag{19}$$

where g, f, and n are the observed, original, and noise images, respectively, k is the blur kernel, and $\mathbf{s}$ and $\mathbf{t}$ are two-dimensional spatial variables. The blur kernel depends on the spatial location $\mathbf{s}$, and hence the blur in this model is spatially variant. The image acquisition model can be rewritten as

$$\mathbf{g} = \mathbf{A}\mathbf{f} + \mathbf{n}, \tag{20}$$

where $\mathbf{g}$, $\mathbf{f}$, and $\mathbf{n}$ are the lexicographically ordered vectors of the observed, original, and noise images. The matrix $\mathbf{A}$ represents the operation by the spatially variant blur kernel k. For images of size $M \times N$, the sizes of vectors are $MN \times 1$, and the size of the matrix $\mathbf{A}$ is $MN \times MN$. Deblurring or restoration is the process of obtaining the original image $\mathbf{f}$ given an observed noisy blurred image $\mathbf{g}$ and the blur model $\mathbf{A}$. This paper focuses on the development of a restoration method assuming the blur model is given.

When the blur introduced by the image acquisition process is spatially invariant, the blur kernel can be specified by

$$k(\mathbf{s}, \mathbf{t}) = k(\mathbf{s} - \mathbf{t}), \tag{21}$$

and the sum in (19) becomes the two-dimensional convolution operation. For spatially invariant blur, the matrix $\mathbf{A}$ in (20) is Toeplitz [32], and with a periodic boundary condition, the matrix $\mathbf{A}$ becomes BCCB [5]. Block circulant matrices can be diagonalized by the DFT matrix. Many restoration algorithms rely on the diagonalization by the DFT matrix for their efficient implementation [1, 2].

When the blur kernel changes depending on the locations of pixels in an observed image, or a periodic boundary condition breaks down, the matrix $\mathbf{A}$ is no longer BCCB. Restoration algorithms that rely on the inversion of the matrix $\mathbf{A}$ cannot be implemented efficiently using the DFT matrix. Moreover, for spatially variant blur, the matrix $\mathbf{A}$ is nonsymmetric in general. Iterative solvers that assume a symmetric matrix cannot be applied directly but must be applied to the normal equation $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{f} = \mathbf{A}^{\mathrm{T}}\mathbf{g}$ for the restoration of spatially variant blur. When the blur is spatially variant and the image size is large, the least squares problem is difficult to solve with the large nonsymmetric and non-Toeplitz matrix $\mathbf{A}$. Even with images of moderate sizes, the size of the matrix $\mathbf{A}$ is considerably large. Explicit construction of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is not practical. The multiplication by the matrix $\mathbf{A}$ or $\mathbf{A}^{\mathrm{T}}$ can be implemented implicitly

by filtering with spatially varying filter coefficients [52, 50, 29]. Using the piecewise constant convolution function and induced DFT implementation, multiplication by $\mathbf{A}$ or $\mathbf{A}^{\mathrm{T}}$ can be approximated efficiently [39, 36, 22, 25, 48, 40]. However, the approximation of $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ introduces mismatches of blur models in the acquisition and restoration processes.

The KADMM developed in the previous section can efficiently work with non-BCCB general matrices. The restoration of spatially variant blur is a good application of our method.

4.2. TV restoration. TV regularization was introduced in [43] for image denoising problems and has been widely used for image deblurring since it generates visually nice results [1, 2, 18, 4, 51]. However, most of these works are limited to the restoration of spatially invariant blur where the matrix $\mathbf{A}$ has Toeplitz structure. We consider the TV restoration for the restoration of spatially variant blur. The optimization problem can be written as

$$\begin{aligned} &\underset{\mathbf{f}\in\mathbb{R}^{MN}}{\text{minimize}}\ \mathrm{TV}(\mathbf{f}) \\ &\text{subject to } \|\mathbf{A}\mathbf{f}-\mathbf{g}\|_2 \le \epsilon, \\ &\mathbf{0} \le \mathbf{f} \le \mathbf{1}, \end{aligned} \tag{22}$$

where the function $\mathrm{TV}(\mathbf{f})$ is the isotropic discrete total variation. The first constraint is on the deviation between the solution and the observed images. The parameter $\epsilon > 0$ can be predetermined by the discrepancy principle [49]. The second constraint limits the dynamic range of the solution. The TV restoration of spatially variant blur can be written in the form of (9). The problem in (22) can be rewritten as an unconstraint optimization problem as

$$\underset{\mathbf{f}\in\mathbb{R}^{MN}}{\text{minimize}}\ \mathrm{TV}(\mathbf{f}) + \delta(\mathbf{A}\mathbf{f}|B(\epsilon,\mathbf{g})) + \delta(\mathbf{f}|R(0,1)), \tag{23}$$

where the indicator function δ is given by

$$\delta(x|X) = \begin{cases} 0 & \text{if } x \in X, \\ +\infty & \text{otherwise.} \end{cases} \tag{24}$$

The set $B(\gamma,\mathbf{c}) := \{\mathbf{x} : \|\mathbf{x}-\mathbf{c}\|_2 \le \gamma\}$ denotes a ball which has radius of γ and is centered at a point $\mathbf{c}$. The set $R(\beta_1,\beta_2) := \{\mathbf{x} : \beta_1 \le x_i \le \beta_2 \text{ for all } i\}$ denotes a rectangle or a box from α to β, which ensures that the solution image f has meaningful values inside the restricted dynamic range. The unconstrained optimization problem (23) is taken from (9) by setting

$$\begin{aligned} h_1 &= \delta(\cdot|B(\epsilon,\mathbf{g})), \\ h_2 &= \mathrm{TV}, \\ h_3 &= \delta(\cdot|R(0,1)), \\ \mathbf{G}_2 &= \mathbf{I}_{MN}, \\ \mathbf{G}_3 &= \mathbf{I}_{MN}, \end{aligned} \tag{25}$$

where $\mathbf{I}_{MN}$ is the identity matrix of size $MN \times MN$. This problem formulation was solved for the restoration of space invariant blur in [2, 18]. The key idea was a DFT implementation for the least squares problem in line 2 of Algorithm 1, which is not applicable to the restoration of spatially variant blur. We apply the KADMM to the TV restoration of spatially variant blur.

4.3. Application of KADMM. The solution space for the restoration problem is restricted to the nth order Krylov subspace. With the zero vector as the initial solution, i.e., $\mathbf{f}_0 = \mathbf{0}$, the image restoration problem in the nth order Krylov subspace becomes

$$\begin{aligned} &\underset{\boldsymbol{\alpha}\in\mathbb{R}^n}{\text{minimize}}\ \mathrm{TV}(\mathbf{Q}_n\boldsymbol{\alpha}) \\ &\text{subject to } \|\mathbf{A}\mathbf{Q}_n\boldsymbol{\alpha} - \mathbf{g}\| \le \epsilon, \\ &\mathbf{0} \le \mathbf{Q}_n\boldsymbol{\alpha} \le \mathbf{1}. \end{aligned} \tag{26}$$

The objective of the corresponding unconstrained optimization problem can be written as

$$\mathrm{TV}(\mathbf{Q}_n\boldsymbol{\alpha}) + \delta(\mathbf{A}\mathbf{Q}_n\boldsymbol{\alpha}|B(\epsilon,\mathbf{g})) + \delta(\mathbf{Q}_n\boldsymbol{\alpha}|R(0,1)). \tag{27}$$

The KADMM algorithm can be applied by identifying (25). The KADMM for the optimization problem in (26) is given in Algorithm 4.

The proximal operator for TV is studied in [16] with a fixed point iteration scheme. The proximal operator for $\delta(\cdot|S)$ is given as a projection on the set S. Therefore,

$$\mathrm{prox}_{\delta(\cdot|B(\epsilon,\mathbf{c}))}(\mathbf{s}) = \begin{cases} \mathbf{c} + \epsilon\frac{\mathbf{s}-\mathbf{c}}{\|\mathbf{s}-\mathbf{c}\|_2}, & \|\mathbf{s}-\mathbf{c}\|_2 > \gamma, \\ \mathbf{s}, & \|\mathbf{s}-\mathbf{c}\|_2 \le \gamma, \end{cases} \tag{28}$$

and

$$\mathrm{prox}_{\delta(\cdot|R(\alpha,\beta))}(\mathbf{s})_i = \begin{cases} \alpha, & s_i \le \alpha, \\ s_i, & \alpha < s_i < \beta, \\ \beta, & s_i \ge \beta. \end{cases} \tag{29}$$

For image restoration problems, the blur is usually modeled with PSFs with compact supports. For spatially variant blur, the PSF changes depending on spatial location. For a PSF with compact support, for example, $w \times w$, the multiplication by $\mathbf{A}$ alone requires $w^2 \times MN$ operations for $M \times N$ size images. The multiplication by $\mathbf{Q}_n$ requires $n \times MN$ operations. For $n < w^2$, it is computationally cheaper to work with $\mathbf{Q}_n$ than with $\mathbf{A}$.

Another possible variable splitting method for TV restoration is to let $\mathbf{G}_2 = \nabla$, where $\nabla \in \mathbb{R}^{2MN\times MN}$ is a matrix representing the discrete gradient operator, and $h_2(u) = \sum_{i=1}^{MN}(u_i^2 + u_{i+MN}^2)^{1/2}$ for $\mathbf{u} \in \mathbb{R}^{2MN}$. This variable splitting method has an advantage in that the proximal operator for h_2 has a closed form [21]. However, there is a slight increase of computational complexity in the least squares problem dealing with $\mathbf{G}_2$. The computationally dominant part of KADMM is the Arnoldi process that runs in the original space $\mathbb{R}^{MN}$. Since the ADMM runs in the reduced space $\mathbb{R}^n$, the computations for the ADMM iterations take only a small part of the total computational complexity. The two variable splitting methods for KADMM are similar to each other in terms of computational complexity.

Algorithm 4 KADMM for the TV restoration.

$\mathbf{Q}_{n+1}, \mathbf{H}_n \leftarrow \text{Arnoldi}(\mathbf{A}, \mathbf{g}, n, \mathbf{0})$
repeat
  $\boldsymbol{\alpha} \leftarrow \arg\min_{\boldsymbol{\alpha}} \left\| \begin{bmatrix} \mathbf{H}_n \\ \mathbf{I}_n \\ \mathbf{I}_n \end{bmatrix} \boldsymbol{\alpha} - \begin{bmatrix} \mathbf{Q}_{n+1}^T(\mathbf{v}_1 + \mathbf{d}_1) \\ \mathbf{Q}_n^T(\mathbf{v}_2 + \mathbf{d}_2) \\ \mathbf{Q}_n^T(\mathbf{v}_3 + \mathbf{d}_3) \end{bmatrix} \right\|_2^2$
  $\mathbf{v}_1 \leftarrow \text{prox}_{\delta(\cdot|B(\epsilon,\mathbf{g}))}(\mathbf{Q}_{n+1}\mathbf{H}_n\boldsymbol{\alpha} - \mathbf{d}_1)$
  $\mathbf{v}_2 \leftarrow \text{prox}_{\text{TV}/\mu}(\mathbf{Q}_n\boldsymbol{\alpha} - \mathbf{d}_2)$
  $\mathbf{v}_3 \leftarrow \text{prox}_{\delta(\cdot|R(0,1))}(\mathbf{Q}_n\boldsymbol{\alpha} - \mathbf{d}_3)$
  $\mathbf{d}_1 \leftarrow \mathbf{d}_1 - (\mathbf{Q}_{n+1}\mathbf{H}_n - \mathbf{v}_1)$
  $\mathbf{d}_2 \leftarrow \mathbf{d}_2 - (\mathbf{Q}_n\boldsymbol{\alpha} - \mathbf{v}_2)$
  $\mathbf{d}_3 \leftarrow \mathbf{d}_3 - (\mathbf{Q}_n\boldsymbol{\alpha} - \mathbf{v}_3)$
until stopping criterion is satisfied.
$\mathbf{f} \leftarrow \mathbf{Q}_n\boldsymbol{\alpha}$

5. Experiments.

5.1. Spatially variant Gaussian blur. The spatially variant Gaussian blur model in [9] is used for the restoration experiments. The blur kernel at the pixel location $\mathbf{s} = (s_1, s_2)$ in the image acquisition model in (19) is obtained from the two-dimensional separable Gaussian function

$$k(\mathbf{s}, \mathbf{t}) = \frac{1}{\xi} \exp\left(-\frac{1}{2} \left(\frac{(t_1 - s_1)^2}{\sigma_x^2(s_1)} + \frac{(t_2 - s_2)^2}{\sigma_y^2(s_2)} \right) \right), \tag{30}$$

where the normalizing constant is given by

$$\xi = 2\pi\sigma_x(s_1)\sigma_y(s_2). \tag{31}$$

The variances σ_x and σ_y are functions of s_1 and s_2, respectively. The blur kernel changes depending on the location $\mathbf{s}$, and hence the blur is spatially variant.

Three types of spatially variant Gaussian blur are considered. The first type has a smaller amount of blur at the center and a larger amount of blur at the corners of an image. The variances of the spatially variant Gaussian blur kernel are given by

$$\begin{aligned} \sigma_x(s_1) &= \gamma|0.5 - s_1/M| + 0.5, \\ \sigma_y(s_2) &= \gamma|0.5 - s_2/N| + 0.5. \end{aligned} \tag{32}$$

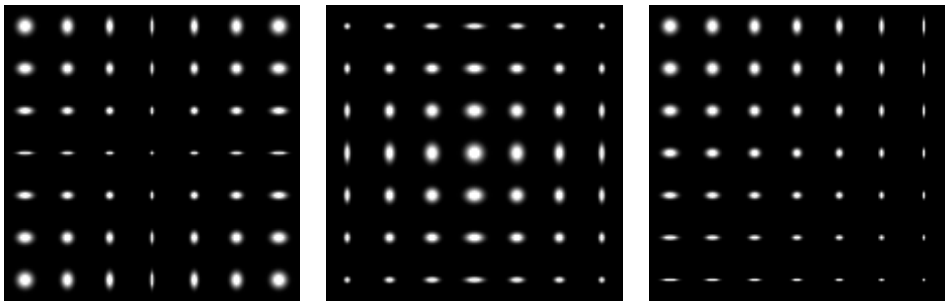
The second type is the opposite of the first case, with more severe blur at the center and milder blur at the corners. The variances are given by

$$\begin{aligned} \sigma_x(s_1) &= -\gamma|0.5 - s_1/M| + 2.5, \\ \sigma_y(s_2) &= -\gamma|0.5 - s_2/N| + 2.5. \end{aligned} \tag{33}$$

The third type has smaller amount of blur at the lower right corner and larger amount of blur at the upper left part of an image. The variances are given by

$$\begin{aligned} \sigma_x(s_1) &= \gamma(0.5 - s_1/M)/2 + 1.25, \\ \sigma_y(s_2) &= \gamma(0.5 - s_2/N)/2 + 1.25. \end{aligned} \tag{34}$$

The scalar γ, which is related to the gradient of the variance, is fixed as 4 in our experiment, and 21×21 kernel size is used for discretization. The spatially variant Gaussian functions sampled at 7×7 locations are shown in Figure 1. Note that the blur kernels in Figure 1 are examples of blur kernels that change their shape pixelwise. Three blur models have severe blurs with large gradient of variances.



(a) (b) (c)

Figure 1. *Examples of spatially variant Gaussian blur used in the experiments. The variances change with the pixel locations in an image. The kernels are sampled at* 7×7 *pixel locations.* (a) *Type* 1: *more severe blur at the corners.* (b) *Type* 2: *more severe blur at the center.* (c) *Type* 3: *more severe blur at the upper left corner.*

5.2. Rotational blur. Rotational blur is another example of spatially variant blur. The rotational blur is a special case of motion blur for which the blurred image $\mathbf{g}$ is formed as a weighted sum of images at incremental times during acquisition. The image acquisition process is modeled by

$$\mathbf{g} = \sum_{t=1}^{T} w_t \mathbf{f}_t + \mathbf{n}, \tag{35}$$

where $\mathbf{f}_t$ is an image at time t, w_t is the normalization weight for the tth image, and $\mathbf{n}$ is the additive noise. Assuming $\mathbf{f}_t$ is a transformed image of $\mathbf{f}$ by a two-dimensional projective transformation and using bilinear interpolation for the transformation, $\mathbf{f}_t$ can be represented as a linear transformation of $\mathbf{f}$, i.e., $\mathbf{f}_t = \mathbf{K}_t\mathbf{f}$, where the matrix $\mathbf{K}_t$ contains a set of coefficients of the bilinear transformation [9]. Then (35) can be written in the form of (20) as

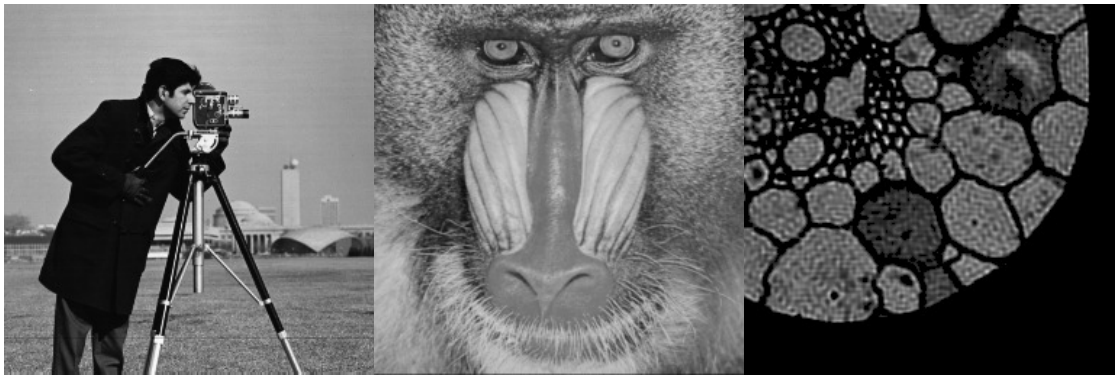
$$\mathbf{g} = \left(\sum_{t=1}^{T} w_t \mathbf{K}_t \right) \mathbf{f} + \mathbf{n} \tag{36}$$

$$= \mathbf{A}\mathbf{f} + \mathbf{n}. \tag{37}$$

In the case of the rotational blur, $\mathbf{f}_t$ is a rotated image centered at the original image center by an angle θ_t. For our experiments, we set $T = 25$ and $w_t = 1/T$ for all t. The three rotational blurs are defined by changing a range parameter ρ, where

$$\theta_t = \rho(t - 13)/12^\circ, \qquad t = 1, 2, \ldots, 25. \tag{38}$$

The range of θ_t is $[-\rho, \rho]$ degrees. The fourth, fifth, and sixth spatially variant blurs are defined by rotational blur with $\rho = 2, 4, 8$, respectively.



(a) (b) (c)

(d) (e)

Figure 2. *Test images of size* 256×256: (a) *Cameraman,* (b) *Baboon,* (c) *Grain,* (d) *HD test image of size* 1920×1080*, and* (e) 36*-megapixel test image of size* 7360×4912.

5.3. Computational environment. *Cameraman, Baboon,* and *Grain* images of size 256×256 pixels, a high definition (HD) image of size 1920×1080 pixels, and a 36-megapixel image of size 7360×4912 pixels used in our experiments are shown in Figure 2. The images are blurred by the spatially variant blur, and Gaussian noise is added to meet given blurred signal-to-noise ratio (BSNR) values. The BSNR is defined by

$$\mathrm{BSNR} = 10 \log \frac{\mathrm{var}(\mathbf{Af})}{\sigma^2}, \tag{39}$$

where $\mathrm{var}(\mathbf{Af})$ is the variance of $\mathbf{Af}$ and σ^2 is the noise variance. The BSNR is set at 60 dB in our experiments.

All the algorithms are implemented with MATLAB. The matrices $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ for spatially variant blur are precomputed and stored as sparse matrices. If not stated otherwise, the matrix vector multiplications are performed with the stored sparse matrices. For larger size images, explicit construction of $\mathbf{A}$ or $\mathbf{A}^{\mathrm{T}}$ is unavailable due to the large dimensionality. In this case, the matrix vector multiplications are performed implicitly as spatially variant filtering operations with CUDA. The computation time for the operation $\mathbf{Ax}$ or $\mathbf{A}^{\mathrm{T}}\mathbf{x}$ includes the time for the memory transfers to and from a GPU.

The value of the augmented Lagrangian parameter μ is set to one, and the value of ϵ is set

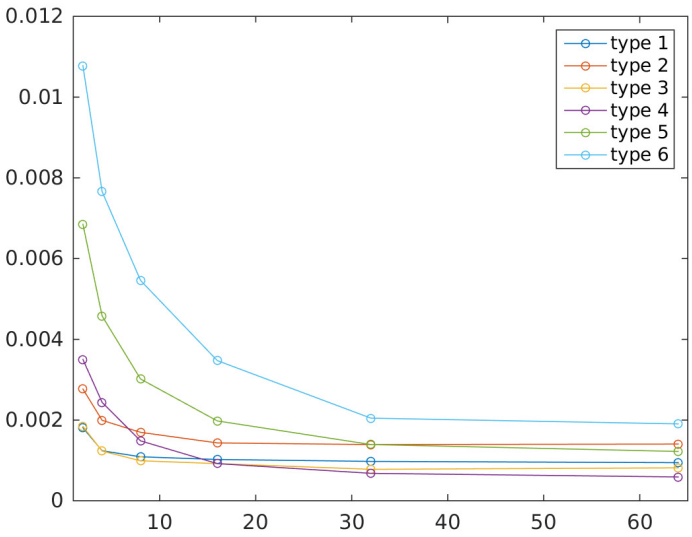


Figure 3. *TV restoration of spatially variant blur using KADMM in terms of the MSE over the number of the basis vectors $n = 2^\zeta$ for $\zeta = 1, 2, \ldots, 6$.*

by Morozov's discrepancy principle [18, 49], $\epsilon = \sqrt{\tau MN}\sigma$, where a rule of thumb is $\tau = 1$. The periodic boundary condition is chosen for the experiments on spatially variant Gaussian blur, and the zero boundary condition is chosen for the experiments on rotational blur. Termination conditions are set by the relative change of the residuals, $\|\mathbf{r}^i - \mathbf{r}^{i-1}\|/\|\mathbf{r}^i\|$, where $\mathbf{r}^i$ is the residual at the ith iteration. They are chosen to be 10^{-5} for all of the experiments with 256×256 size images. For the experiments with larger size images, they are chosen to be 10^{-3}.

5.4. Numerical results. The performance of Krylov subspace methods is affected by the choice of the order of Krylov subspace n. Figure 3 shows the performance of the KADMM when the order of Krylov subspace $n = 2^1, 2^2, \ldots, 2^6$. The average mean square error (MSE) of the three 256×256 size test images for the three different Gaussian blur types and the three different rotational blurs is shown. It can be seen that, in general, the performance increases as the order of Krylov subspace n increases. The decreases of MSEs become incrementally smaller as n increases. Note that as n increases, more memory is needed to store the $\mathbf{Q}_{n+1}$ matrix of size $MN \times (n + 1)$. Moreover, the size of the least squares problem increases with n. In the following experiments, the order of the Krylov subspace is chosen to be 16, considering the tradeoff between the performance and the computational complexity and memory requirement. The choice of the order of the Krylov subspace, $n = 16$ for our experiments, is not an unusually small choice. GMRES and Arnoldi–Tikhonov with $n = 7, 8$ and $n = 6, 7$, respectively, are applied to restoration problems in [14, 42], respectively. Examples of restored images by KADMM with different n are shown in Figure 4.

The performance of the proposed KADMM is compared to that of the NESTA [8] and PA [21] methods. NESTA and PA find the solutions in the original solution space. The matrix

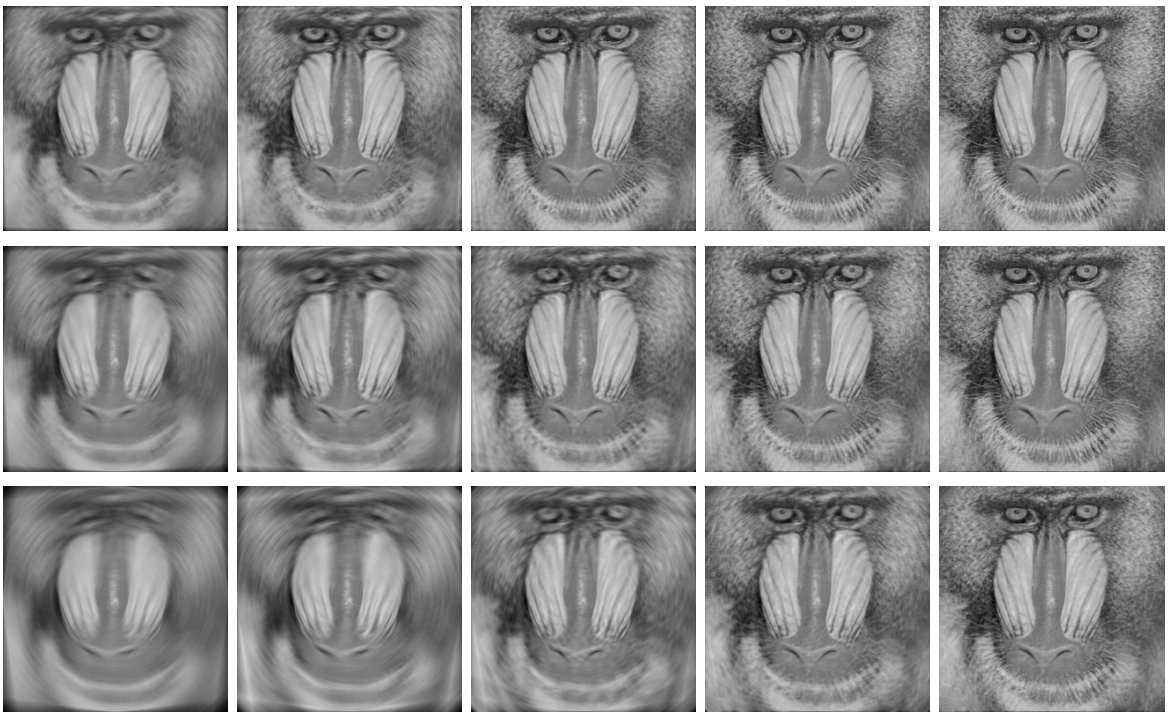


Figure 4. *TV restoration of spatially variant rotational blur using KADMM with the Baboon image. From top to bottom: Blur type* $4, 5, 6$. *From left to right: Observed image followed by images restored with* $n = 2^2, 2^3, 2^4, 2^5$.

vector multiplications of $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ for NESTA and PA are also implemented using sparse matrices.

The quality of the restored images is measured using the MSE in Tables 1 and 2 for the Gaussian and rotational blur, respectively. Table 3 shows the comparison of MSE for the Gaussian blur with the HD test image. NESTA and PA find the solutions in the original solution space, $\mathbb{R}^{MN}$, while the KADMM finds the solutions in the nth order Krylov subspace, $\mathcal{K}_n(\mathbf{A}, \mathbf{r})$. It can be seen that all three methods report similar MSE values. Small differences in the MSE values of NESTA, PA, and KADMM indicate that the restriction of the solution space does not significantly hinder the quality of the restored images.

Table 1

Numerical results of TV restoration of spatially variant Gaussian blur on test images.

Problem		Calls to $\mathbf{A}$, $\mathbf{A}^T$			Dim. of search space			Iteration #			CPU time (sec)			MSE ($\times 10^{-4}$)		
Image	Blur	PA	NESTA	KADMM	PA	NESTA	KADMM	PA	NESTA	KADMM	PA	NESTA	KADMM	PA	NESTA	KADMM
Cameraman	1	4286	3013	17	65536	65536	16	2144	753	102	129.0	85.3	2.7	5.69	8.21	7.01
	2	6000	3269	17	65536	65536	16	3001	817	192	172.0	92.4	4.3	19.7	28.0	22.2
	3	5584	3073	17	65536	65536	16	2793	768	226	159.4	86.3	5.0	5.64	11.5	7.82
Baboon	1	3402	4569	17	65536	65536	16	1702	1142	3001	96.8	137.7	60.1	12.2	13.4	8.41
	2	3612	4717	17	65536	65536	16	1807	1179	3001	110.8	133.4	59.3	11.6	12.4	7.76
	3	3248	4529	17	65536	65536	16	1625	1132	3001	677.4	944.8	43.5	9.82	10.6	6.69
Grain	1	3940	2069	17	65536	65536	16	1971	517	50	113.0	58.7	1.5	3.27	13.1	8.07
	2	4560	2113	17	65536	65536	16	2281	528	101	130.9	64.2	2.5	6.43	19.8	9.70
	3	3754	2377	17	65536	65536	16	1878	594	149	113.6	66.7	3.5	3.71	13.7	6.16

Table 2

Numerical results of TV restoration of spatially variant rotational blur on test images.

Problem		Calls to $\mathbf{A}$, $\mathbf{A}^T$			Dim. of search space			Iteration #			CPU time (sec)			MSE ($\times 10^{-4}$)		
Image	Blur	PA	NESTA	KADMM	PA	NESTA	KADMM	PA	NESTA	KADMM	PA	NESTA	KADMM	PA	NESTA	KADMM
Cameraman	4	3500	2881	17	65536	65536	16	1751	720	205	9.3	6.7	4.1	1.72	8.96	7.47
	5	4556	3321	17	65536	65536	16	2279	830	270	16.0	11.2	5.4	3.21	18.2	16.1
	6	6000	3545	17	65536	65536	16	3001	886	349	32.0	17.5	6.9	5.18	35.2	31.6
Baboon	4	2924	4289	17	65536	65536	16	1463	1072	231	8.0	9.8	4.6	4.03	8.68	6.84
	5	3474	4237	17	65536	65536	16	1738	1059	299	12.2	14.3	6.0	7.55	16.5	13.2
	6	4100	3837	17	65536	65536	16	2051	959	291	22.0	18.9	5.8	12.2	25.8	21.1
Grain	4	3978	2349	17	65536	65536	16	1990	587	261	10.7	5.4	5.2	0.49	14.8	8.78
	5	5040	2977	17	65536	65536	16	2521	744	287	17.8	10.0	5.7	1.64	34.7	25.3
	6	6000	3429	17	65536	65536	16	3001	857	276	32.3	16.7	5.5	5.02	63.0	48.1

Table 3

Numerical results of TV restoration of spatially variant Gaussian blur on HD test images.

Problem		Calls to $\mathbf{A}$, $\mathbf{A}^T$			Dim. of search space			Iteration #			CPU time (sec)			MSE ($\times 10^{-4}$)		
Image	Blur	PA	NESTA	KADMM	PA	NESTA	KADMM	PA	NESTA	KADMM	PA	NESTA	KADMM	PA	NESTA	KADMM
HD	1	636	409	17	2073600	2073600	16	319	102	90	2379.2	1528.2	113.1	12.7	22.8	10.7
	2	648	373	17	2073600	2073600	16	325	93	93	2425.5	1394.2	115.5	21.2	32.3	17.4
	3	594	341	17	2073600	2073600	16	298	85	35	2222.7	1276.1	77.4	6.64	15.7	4.86

The evolutions over time of the quadratic constraint $\|\mathbf{Af}-\mathbf{g}\|_2$ that measures the deviation from a given image, the objective TV($\mathbf{f}$) that measures the roughness of the solution, and the deviation from the range constraint are shown in Figures 5(a), 5(b), and 5(c), respectively. The evolution of the MSE is shown in Figure 5(d). The Cameraman image is used with blur type 1 as an example. The figures use log-log scales. It can be seen that both the objectives and constraints converge over time for NESTA, PA, and KADMM. The algorithms terminate with the quadratic constraint. The convergence for KADMM is much faster than those of NESTA and PA. KADMM requires computation of the basis for the Krylov subspace via the Arnoldi process before it enters into the ADMM iteration. It can be seen in Figure 5 that the values of the objectives and constraints begin to appear after some time, i.e., after the time it takes for the algorithm to find the basis. However, the objectives and constraints converge in a very short time once the iterative optimization process begins. Overall, the speed of KADMM is much faster than the speed of NESTA and PA. The CPU times in terms of wall-clock times in Table 1 show that the proposed KADMM is faster than NESTA and PA.

Tables 1–3 also show a comparison of the computational complexity. The dominant arithmetic operations are the matrix multiplications. The numbers of matrix vector multiplications by $\mathbf{A}$ or $\mathbf{A}^{\mathrm{T}}$ are shown for NESTA, PA, and KADMM, indicating that the computational complexity of KADMM is significantly lower than that of NESTA and PA. Note that the iterations of NESTA and PA involve operations in the original solution space, whereas the iterations of KADMM involve operations in the reduced solution space. By the Krylov subspace method, not only is the dimension of the search space reduced from d to n, but the size of the least squares problem in the ADMM also is reduced from $dJ \times d$ to $(nJ+1) \times n$. For the restoration of a 256×256 size image with $J = 3$, the size of the problem is reduced from $(256 \cdot 256 \cdot 3) \times (256 \cdot 256)$ to $(16 \cdot 3 + 1) \times 16$ for KADMM with $n = 16$. And for the restoration of a 1920×1080 size image with $J = 3$, the size of the problem is reduced from $(1920 \cdot 1080 \cdot 3) \times (1920 \cdot 1080)$ to $(16 \cdot 3 + 1) \times 16$ for KADMM with $n = 16$. The order reduction by the Krylov subspace method is significant. In terms of CPU time, KADMM is much faster than PA and NESTA, especially for larger size images. For the experiments in Table 3, the matrix vector multiplications of $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ are implemented using CUDA. The large size of

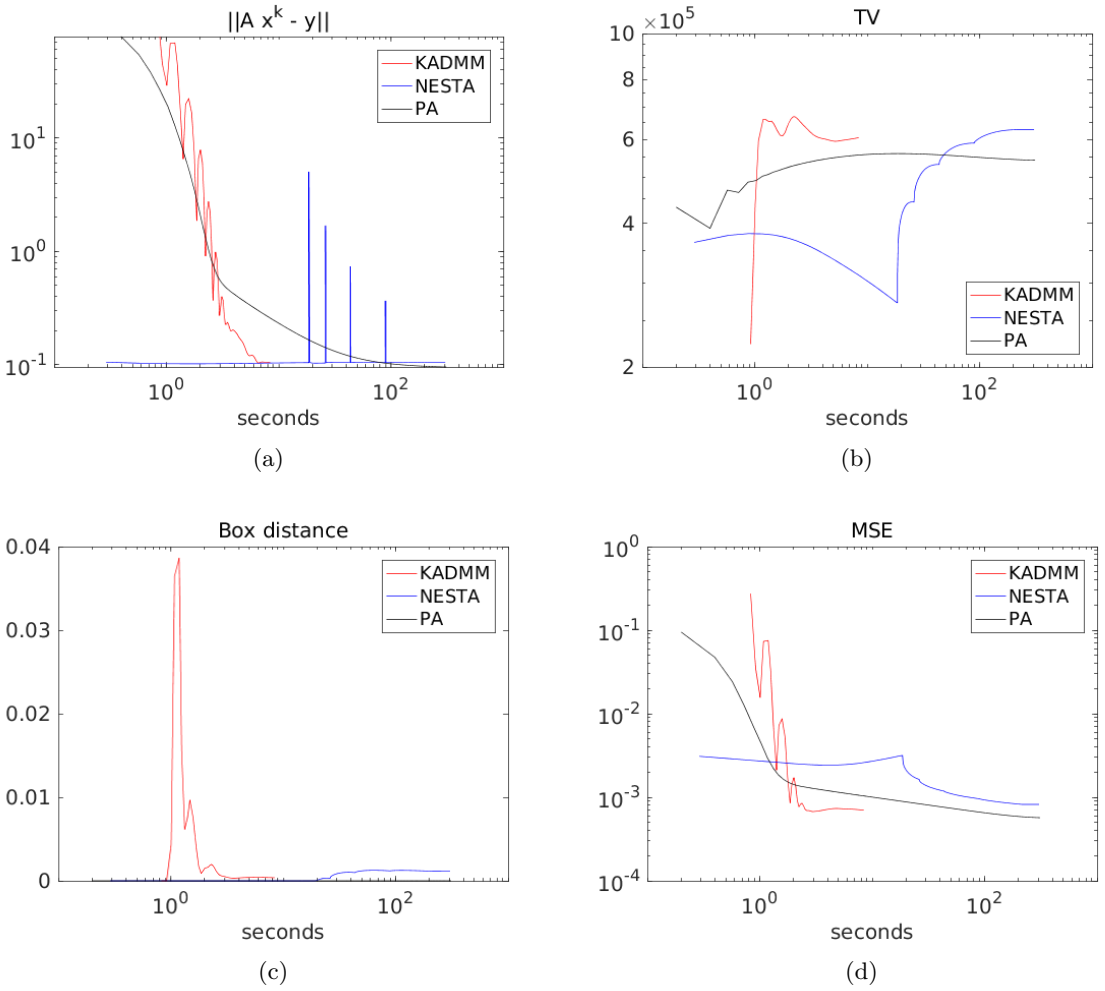


Figure 5. *TV restoration of spatially variant blur using PA (black line), NESTA (blue line), and KADMM (red line) with the Cameraman image.* (a) *Evolution of quadratic constraint* $\|\mathbf{Af}-\mathbf{g}\|_2$ *over time (sec).* (b) *Evolution of the objective* $\mathrm{TV}(\mathbf{f})$ *over time (sec).* (c) *Distance to range constraint over time (sec).* (d) *MSE over time (sec).*

the HD images prohibits explicit implementation of matrix vector multiplications with sparse matrices. The CPU time in Table 3 is not directly comparable to that in Tables 1 and 2.

Table 4 shows the performance of the KADMM for images of different sizes. The 36-megapixel test image shown in Figure 2(e) is scaled to various sizes: 230×154, 460×307, 920×614, 1840×1228, 3680×2456, and 7360×4912 pixels. The Gaussian blur type 2 is applied at the BSNR of 60dB. KADMM, with $n = 16$, is applied to restore the blur. The large sizes of the images prohibit explicit implementation of matrix vector multiplications with sparse matrices. The matrix vector multiplications of $\mathbf{A}$ and $\mathbf{A}^{\mathrm{T}}$ are implemented using CUDA for this experiment. The MSE values show that the KADMM restores the spatially variant blur of various size images effectively. The number of pixels in the images quadruples for the next larger images. The speed of the KADMM increases at about the same rates for larger size images.

Examples of TV restoration by PA, NESTA, and KADMM for the Gaussian blur with the

Table 4

Numerical results of TV restoration of spatially variant Gaussian blur with various size images.

Size	230×154	460×307	920×614	1840×1228	3680×2456	7360×4912
Calls to $\mathbf{A}$, $\mathbf{A}^T$	17	17	17	17	17	17
Dim. of search space	16	16	16	16	16	16
Iteration #	52	61	86	92	64	51
CPU time (sec)	3.2	7.9	31.1	122.2	393.5	1427.0
MSE	4.79E-04	2.65E-04	1.14E-04	3.64E-05	1.69E-05	1.80E-05

Figure 6. *TV restoration of spatially variant Gaussian blur with the Cameraman image. From top to bottom: Blur type* $1, 2, 3$. *From left to right: Observed image followed by images restored by PA, NESTA, and KADMM.*

Cameraman image are presented in Figure 6. The experiments are performed with the three Gaussian blur types. Blurred images and images restored by the three algorithms are shown. Each blur type imposes a different amount of blur depending on the pixel locations. It can be seen that all three algorithms restore sharp images for the three types of spatially variant Gaussian blur. An example of a restored image with HD size 1920×1080 is given in Figure 7, where the observed image and the image restored by KADMM are shown in Figures 7(a) and

(a)

(b)

(c)

Figure 7. *TV restoration of spatially Gaussian variant blur with an HD size* 1920×1080 *image with blur type* 2. (a) *Noisy blurred image.* (b) *Image restored by KADMM.* (c) *Parts of* (a) *and* (b).

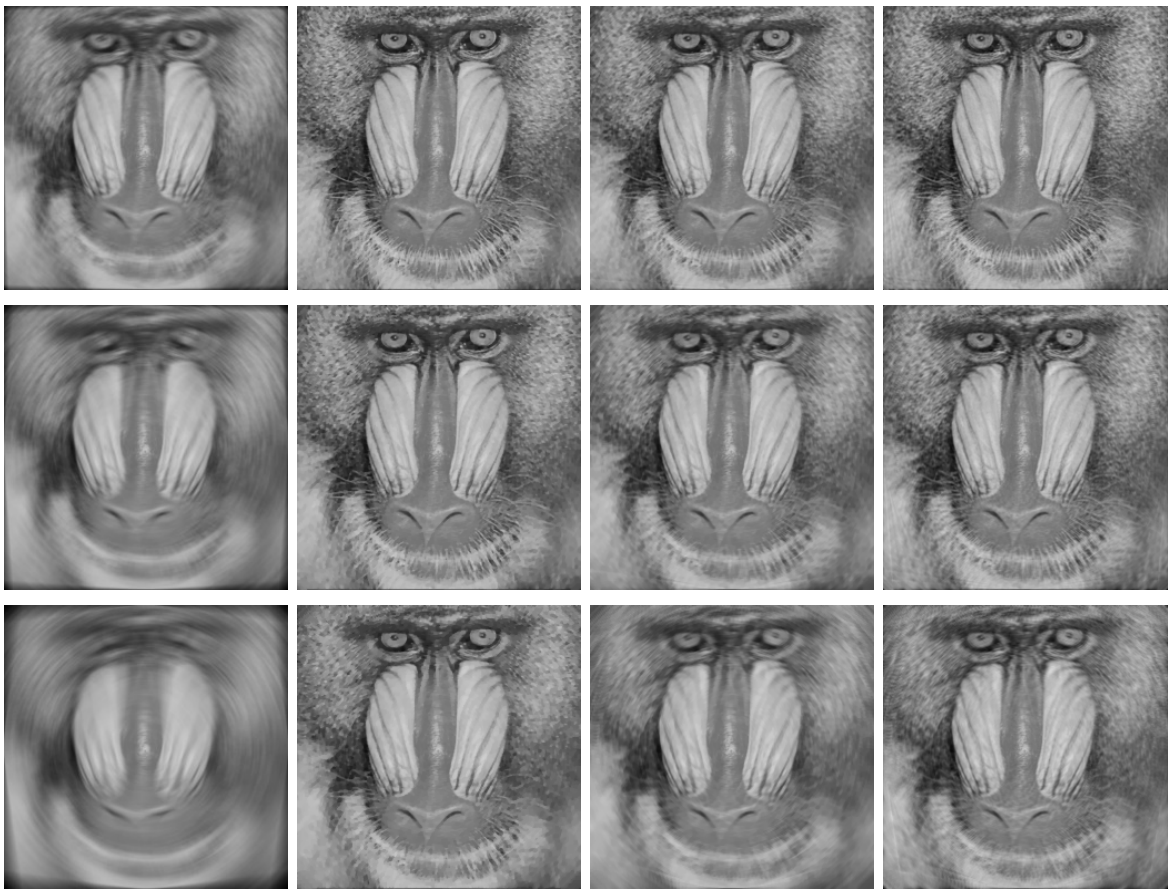


Figure 8. *TV restoration of spatially variant rotational blur with the Cameraman image. From top to bottom: Blur type* $4, 5, 6$*. From left to right: Observed image followed by images restored by PA, NESTA, and KADMM.*

7(b), respectively. The observed image is blurred by type 2, which imposes more severe blur at the center and milder blur at the corners, and is contaminated by the Gaussian noise. It can be seen that the noisy and blurry HD image is effectively restored by KADMM.

Examples of TV restoration by PA, NESTA, and KADMM for the rotational blur with the Baboon image are presented in Figure 8. The experiments are performed with the three rotational blur types. Blurred images and images restored by the three algorithms are shown. Each blur type imposes a different amount of blur depending on the pixel locations. It can be seen that all three algorithms restore sharp images for the three types of spatially variant rotational blur.

6. Conclusion. This work presents a convex optimization framework based on ADMM and Krylov subspace methods. The proposed KADMM finds an inexact solution by restricting the solution space to the Krylov subspace of small order. The basis of the Krylov subspace is found by the Arnoldi process, and ADMM is applied in the Krylov subspace. Experiments on restoration of spatially variant blur with TV show that meaningful solutions can be found

in the restricted solution space in a short time. The quality of images restored by KADMM is comparable to that of other convex optimization restoration methods such as NESTA and PA. The computational complexity of KADMM is significantly reduced by the order reduction with the Krylov subspace method. The proposed methods can be applied to various inverse problems represented by nonsymmetric non-Toeplitz matrices.

REFERENCES

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, *Fast image recovery using variable splitting and constrained optimization*, IEEE Trans. Image Process., 19 (2010), pp. 2345–2356, https://doi.org/10.1109/TIP.2010.2047910.

[2] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo, *An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems*, IEEE Trans. Image Process., 20 (2011), pp. 681–695, https://doi.org/10.1109/TIP.2010.2076294.

[3] A. Agrawal, Y. Xu, and R. Raskar, *Invertible motion blur in video*, ACM Trans. Graph., 28 (2009), 95, https://doi.org/10.1145/1531326.1531401.

[4] M. S. Almeida and M. A. Figueiredo, *Deconvolving images with unknown boundaries using the alternating direction method of multipliers*, IEEE Trans. Image Process., 22 (2013), pp. 3074–3086, https://doi.org/10.1109/TIP.2013.2258354.

[5] H. C. Andrews and B. R. Hunt, *Digital Image Restoration*, Prentice-Hall Signal Processing Series 1, Prentice-Hall, Englewood Cliffs, NJ, 1977.

[6] M. R. Banham and A. K. Katsaggelos, *Digital image restoration*, IEEE Signal Processing Mag., 14 (1997), pp. 24–41, https://doi.org/10.1109/79.581363.

[7] A. Beck and M. Teboulle, *Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems*, IEEE Trans. Image Process., 18 (2009), pp. 2419–2434, https://doi.org/10.1109/TIP.2009.2028250.

[8] S. Becker, J. Bobin, and E. J. Candès, *NESTA: A fast and accurate first-order method for sparse recovery*, SIAM J. Imaging Sci., 4 (2011), pp. 1–39, https://doi.org/10.1137/090756855.

[9] S. Berisha and J. G. Nagy, *Iterative methods for image restoration*, in Academic Press Library in Signal Processing: Volume 4—Image, Video Processing and Analysis, Hardware, Audio, Acoustic and Speech Processing, Academic Press, Waltham, MA, 2014, pp. 193–247, https://doi.org/10.1016/B978-0-12-396501-1.00007-8.

[10] D. P. Bertsekas, *Convex Optimization Theory*, Athena Scientific, Belmont, MA, 2009.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Vol. 23, Prentice Hall, Englewood Cliffs, NJ, 1989.

[12] 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., 3 (2011), pp. 1–122, https://doi.org/10.1561/2200000016.

[13] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004, https://doi.org/10.1017/CBO9780511804441.

[14] D. Calvetti, B. Lewis, and L. Reichel, *Restoration of images with spatially variant blur by the GMRES method*, in Advanced Signal Processing Algorithms, Architectures, and Implementations X, Proc. SPIE 4116, International Society for Optical Engineering, Bellingham, WA, 2000, pp. 364–374.

[15] D. Calvetti, S. Morigi, L. Reichel, and F. Sgallari, *Tikhonov regularization and the L-curve for large discrete ill-posed problems*, J. Comput. Appl. Math., 123 (2000), pp. 423–446, https://doi.org/10.1016/S0377-0427(00)00414-3.

[16] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20 (2004), pp. 89–97.

[17] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, *An augmented Lagrangian method for total variation video restoration*, IEEE Trans. Image Process., 20 (2011), pp. 3097–3111, https://doi.org/10.1109/TIP.2011.2158229.

[18] H. Chuan, H. Chang-Hua, W. Zhang, and S. Biao, *Box-constrained total-variation image restoration with automatic parameter estimation*, Acta Automat. Sinica, 40 (2014), pp. 1804–1811.

[19] P. L. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing*, in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer, New York, 2011, pp. 185–212, https://doi.org/10.1007/978-1-4419-9569-8_10.

[20] P. L. Combettes and V. R. Wajs, *Signal recovery by proximal forward-backward splitting*, Multiscale Model. Simul., 4 (2005), pp. 1168–1200, https://doi.org/10.1137/050626090.

[21] L. Condat, *A generic proximal algorithm for convex optimization—application to total variation minimization*, IEEE Signal Process. Lett., 21 (2014), pp. 985–989.

[22] L. Denis, E. Thiébaut, F. Soulez, J.-M. Becker, and R. Mourya, *Fast approximations of shift-variant blur*, Int. J. Comput. Vis., 115 (2015), pp. 253–278, https://doi.org/10.1007/s11263-015-0817-x.

[23] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Programming, 55 (1992), pp. 293–318, https://doi.org/10.1007/BF01581204.

[24] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Math. Appl. 375, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[25] P. Escande and P. Weiss, *Sparse wavelet representations of spatially varying blurring operators*, SIAM J. Imaging Sci., 8 (2015), pp. 2976–3014, https://doi.org/10.1137/151003465.

[26] M. Fortin and R. Glowinski, *Decomposition-coordination methods using an augmented Lagrangian*, in Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems, Stud. Math. Appl. 15, North-Holland, Amsterdam, 1983, pp. 97–146.

[27] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite element approximation*, Comput. Math. Appl., 2 (1976), pp. 17–40, https://doi.org/10.1016/0898-1221(76)90003-1.

[28] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, 2013.

[29] N. Hajlaoui, C. Chaux, G. Perrin, F. Falzon, and A. Benazza-Benyahia, *Satellite image restoration in the context of a spatially varying point spread function*, J. Opt. Soc. Amer. A, 27 (2010), pp. 1473–1481, https://doi.org/10.1364/JOSAA.27.001473.

[30] D. Han and X. Yuan, *A note on the alternating direction method of multipliers*, J. Optim. Theory Appl., 155 (2012), pp. 227–238, https://doi.org/10.1007/s10957-012-0003-z.

[31] P. C. Hansen, *Deconvolution and regularization with Toeplitz matrices*, Numer. Algorithms, 29 (2002), pp. 323–378.

[32] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: Matrices, Spectra, and Filtering*, Fundam. Algorithms 3, SIAM, Philadelphia, 2006, https://doi.org/10.1137/1.9780898718874.

[33] C. He, C. Hu, W. Zhang, B. Shi, and X. Hu, *Fast total-variation image deconvolution with adaptive parameter estimation via split Bregman method*, Math. Probl. Eng., 2014 (2014), 617026, https://doi.org/10.1155/2014/617026.

[34] F. Heide, M. Rouf, M. B. Hullin, B. Labitzke, W. Heidrich, and A. Kolb, *High-quality computational imaging through simple lenses*, ACM Trans. Graph., 32 (2013), 149, https://doi.org/10.1145/2516971.2516974.

[35] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf, *Fast removal of non-uniform camera shake*, in Proceedings of the 2011 IEEE International Conference on Computer Vision (ICCV), 2011, pp. 463–470, https://doi.org/10.1109/ICCV.2011.6126276.

[36] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling, *Efficient filter flow for space-variant multiframe blind deconvolution*, in Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 607–614, https://doi.org/10.1109/CVPR.2010.5540158.

[37] J. Jang, J. D. Yun, and S. Yang, *Modeling non-stationary asymmetric lens blur by normal sinh-arcsinh model*, IEEE Trans. Image Process., 25 (2016), pp. 2184–2195, https://doi.org/10.1109/TIP.2016.2539685.

[38] E. Kee, S. Paris, S. Chen, and J. Wang, *Modeling and removing spatially-varying optical blur*, in Proceedings of the 2011 IEEE International Conference on Computational Photography (ICCP), 2011, pp. 1–8, https://doi.org/10.1109/ICCPHOT.2011.5753120.

[39] J. G. Nagy and D. P. O'Leary, *Restoring images degraded by spatially variant blur*, SIAM J. Sci. Comput., 19 (1998), pp. 1063–1082, https://doi.org/10.1137/S106482759528507X.

[40] D. O'Connor and L. Vandenberghe, *Total variation image deblurring with space-varying kernel*, Comput. Optim. Appl., (2017), https://doi.org/10.1007/s10589-017-9901-1.
[41] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629, https://doi.org/10.1137/0712047.
[42] L. Reichel, F. Sgallari, and Q. Ye, *Tikhonov regularization based on generalized Krylov subspace methods*, Appl. Numer. Math., 62 (2012), pp. 1215–1228, https://doi.org/10.1016/j.apnum.2010.10.002.
[43] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268, https://doi.org/10.1016/0167-2789(92)90242-F.
[44] Y. Saad, *Krylov subspace methods for solving large unsymmetric linear systems*, Math. Comp., 37 (1981), pp. 105–126, https://doi.org/10.1090/S0025-5718-1981-0616364-6.
[45] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003, https://doi.org/10.1137/1.9780898718003.
[46] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869, https://doi.org/10.1137/0907058.
[47] W. J. Smith, *Modern Optical Engineering: The Design of Optical Systems*, McGraw-Hill, New York, 1966.
[48] J. Wei, C. A. Bouman, and J. P. Allebach, *Fast space-varying convolution using matrix source coding with applications to camera stray light reduction*, IEEE Trans. Image Process., 23 (2014), pp. 1965–1979, https://doi.org/10.1109/tip.2014.2311657.
[49] Y.-W. Wen and R. H. Chan, *Parameter selection for total-variation-based image restoration using discrepancy principle*, IEEE Trans. Image Process., 21 (2012), pp. 1770–1781, https://doi.org/10.1109/tip.2011.2181401.
[50] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, *Non-uniform deblurring for shaken images*, Int. J. Comput. Vis., 98 (2012), pp. 168–186, https://doi.org/10.1007/s11263-011-0502-7.
[51] S. Xie and S. Rahardja, *Alternating direction method for balanced image restoration*, IEEE Trans. Image Process., 21 (2012), pp. 4557–4567, https://doi.org/10.1109/tip.2012.2206043.
[52] X. Zhou, F. Zhou, X. Bai, and B. Xue, *A boundary condition based deconvolution framework for image deblurring*, J. Comput. Appl. Math., 261 (2014), pp. 14–29, https://doi.org/10.1016/j.cam.2013.10.028.
[53] S. Zhuo and T. Sim, *Defocus map estimation from a single image*, Pattern Recogn., 44 (2011), pp. 1852–1858, https://doi.org/10.1016/j.patcog.2011.03.009.