SHORT-AND-SPARSE DECONVOLUTION VIA RANK-ONE CONSTRAINED OPTIMIZATION (ROCO)

Cheng Cheng and Wei Dai

Department of Electrical and Electronic Engineering, Imperial College London, UK

ABSTRACT

Short-and-sparse deconvolution (SaSD) aims to recover a short kernel and a long and sparse signal from their convolution. In the literature, formulation of blind deconvolution is either a convex programming via a matrix lifting of convolution, or a bilinear Lasso. Optimization solvers are typically based on bilinear factorizations. In this paper, we formulate SaSD as a non-convex optimization with a rank-one matrix constraint, hence referred to as Rank-One Constrained Optimization (ROCO). The solver is based on alternating direction method of multipliers (ADMM). It operates on the full rank-one matrix rather than bilinear factorizations. Closed form updates are derived for the efficiency of ADMM. Simulations include both synthetic data and real images. Results show substantial improvements in recovery accuracy (at least 19dB in PSNR for real images) and comparable runtime compared with benchmark algorithms based on bilinear factorization.

***Index Terms*—** Alternating direction method of multipliers (ADMM), blind deconvolution, image deblurring, non-convex optimization, rank-one constraint

1. INTRODUCTION

Blind deconvolution is a problem to recover two unknown sequences from their convolution. This problem arises in many applications, including astronomy image [1, 2], remote sensing [3, 4], medical ultrasound image [5, 6] and so on. Blind deconvolution problems are fundamentally ill-posed as the number of unknown variables is more than the number of observations. Extra assumptions (typically in the form of prior knowledge) must be imposed to reduce the number of unknowns. Early methods for blind deconvolution include iterative Fourier transform [7], maximum likelihood (ML) estimation [8] and maximum-a-posteriori (MAP) estimation [9, 10, 11, 12].

In the era of compressive sensing (CS) [13] and sparse recovery, modern techniques including [14, 15, 16, 17, 18, 19] assume that either the two unknown sequences lie in fixed low-dimensional spaces, or one is in a fixed low-dimensional space and the other is sparse. The seminal work [14] by Ahmed et. al. studies on the former case and formulates blind deconvolution as a low-rank matrix recovery problem. It is based on the observation that convolution can be viewed as a linear operator acting on a rank-one matrix formed by the outer product of the two unknown sequences. By this matrix lifting technique, a convex optimization is then developed where the nuclear norm is employed to promote low-rankness. As an efficient solver, Burer-Monteiro factorization of low-rank matrices is adopted where the full low-rank matrix is replaced with two factor matrices of much smaller dimensions [14, Section 2.1].

Recently, short-and-sparse deconvolution (SaSD) has received considerable attention [20, 21, 22, 23]. It assumes that the convolution is from a short kernel and a sparse signal. Instead of matrix lifting, a popular approach [20, 21, 22, 23] is utilized to cast blind deconvolution as a bilinear Lasso problem and solve it by alternating minimization, where the two unknown sequences are updated alternatively by fixing the other. The bilinear Lasso formulation is non-convex. Nevertheless, analysis in [20, 21] shows that under certain conditions, every local minimum is close to some shift of the ground truth. The popularity of bilinear Lasso is based on its good empirical performance and advantages in computational complexity: it has been widely believed that it is expensive to directly operate on the full low-rank matrix in matrix lifting methods.

This paper focuses on the SaSD problem. Our optimization formulation follows the matrix lifting approach but is with a non-convex constraint that the lifted matrix is of rank-one, hence referred to as Rank-One Constrained Optimization (ROCO). An alternating direction method of multipliers (ADMM) algorithm is developed to solve this non-convex optimization problem. It works on the full rank-one matrix directly instead of the bilinear Burer-Monteiro factorization. To address potential computational complexity issues, closed forms are derived to evaluate the convolution operator and to update the full rank-one matrix in ADMM. The numerical tests include both synthetic data and real images from MNIST dataset, and compare ROCO with multiple benchmark algorithms based on bilinear Lasso. Simulations show that ROCO not only achieves substantial improvements in recovery accuracy (at least 19dB in PSNR for real images) but also has comparable runtime (sometimes less than) with that of benchmark algorithms.

2. BACKGROUND

The circular convolution of two N-point periodic sequences $x[n]$ and $h[n]$ is defined as

$$y[n] = x[n] \circledast h[n] = \sum_{m=0}^{N-1} x[m]h[n-m].$$

Convolution of two finite-length sequences can be represented by a circular convolution by zero-padding the end of the sequences to a common length N.

Blind deconvolution is to recover both the sequences $x[n]$ and $h[n]$ from their circular convolution $y[n]$. It is well known that the solutions to blind deconvolution subject to scaling and shifting ambiguities. That is, if $y[n] = x[n] \circledast h[n]$, then $y[n] = (ax[n]) \circledast (h[n]/a)$ and $y[n] = S_\tau(x[n]) \circledast S_{-\tau}(h[n])$, where $a \neq 0$ is a scaling constant, $S_\tau(x[n]) = x[n-\tau]$ denotes a shift of the sequence $x[n]$ by τ positions, and $S_{-\tau}(h[n]) = h[n+\tau]$ denotes a shift in the other direction.

This paper focuses on the problem of short-and-sparse blind deconvolution [21, 22]. Consider a convolution of two finite-length sequences: without loss of generality assume that $x[n]$ is sparse and

$h[n]$ is short. Write the sequences as vectors, i.e., $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ and $\boldsymbol{h} \in \mathbb{R}^M$ with $M \ll N$. Blind deconvolution can be then written as

$$\min_{\boldsymbol{x}, \boldsymbol{h}} \ \|\boldsymbol{x}\|_0 \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{x} \circledast \boldsymbol{h}, \tag{1}$$

where $\| \cdot \|_0$ denotes ℓ_0 pseudo-norm which counts the number of non-zero elements. As a convex relaxation of the sparsity, the ℓ_1-norm can replace the ℓ_0 pseudo-norm in (1). To partially address the scaling ambiguity, an extra constraint $\|\boldsymbol{h}\|_2 = 1$ can be added. These yield the well-known bilinear Lasso form [22]

$$\min_{\boldsymbol{x}, \boldsymbol{h}} \ \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x} \circledast \boldsymbol{h}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \quad \text{s.t. } \|\boldsymbol{h}\|_2 = 1. \tag{2}$$

Note that there is a basic trade-off between the sparsity level of $\boldsymbol{x}$ and the shift-coherence of $\boldsymbol{h}$ (coherence between $\boldsymbol{h}$ and its cyclic-shifts) [21]. The work [22] summarizes several popular methods to solve (2). The basic approach is the Alternating Descent Method (ADM) where in each iteration, one updates $\boldsymbol{x}$ by fixing $\boldsymbol{h}$ and then alternatively updates $\boldsymbol{h}$ (using a Riemannian gradient descent [24]) by fixing $\boldsymbol{x}$. It can be shown that the objective function decreases monotonically and hence a convergence is guaranteed [21]. To speed up the convergence, two other variations are studied in [22]: the in-ertial Alternating Descent Method (iADM) adds momentum to the descent direction to mitigate possible oscillations in the optimiza-tion process, and the homotopy-ADM method applies a homotopy continuation method on top of ADM.

The optimization (2) is still non-convex due to the bilinear term in the objective function. On the other hand, SaSD problem (1) can be relaxed into a convex optimization problem by using the matrix lifting technique [14]. This is based on the observations that $\boldsymbol{Z} := \boldsymbol{x}\boldsymbol{h}^\top$ is a rank-one matrix, that the convolution can be written as a linear operator acting on $\boldsymbol{Z}$, i.e., $\boldsymbol{y} = \mathcal{A}(\boldsymbol{Z})$, and that the sparsity in $\boldsymbol{x}$ can be translated into row sparsity of $\boldsymbol{Z}$. Hence, SaSD problem can be reformulated as

$$\min_{\boldsymbol{Z}} \ \|\boldsymbol{Z}\|_* + \lambda_1\|\boldsymbol{Z}^\top\|_{2,1} + \frac{\lambda_2}{2}\|\boldsymbol{y} - \mathcal{A}(\boldsymbol{Z})\|_2^2, \tag{3}$$

where the nuclear norm $\| \cdot \|_*$ promotes low-rank solutions and the $\ell_{2,1}$-norm $\| \cdot \|_{2,1}$ promotes a solution $\boldsymbol{Z}$ with a small number of non-zero rows. However, there is not much discussion of directly solving (3) in the literature. This is partly because the performance of (3) is typically not as good as that of the non-convex counter-part (2), partly due to the widespread belief that it is much more computationally efficient to operate on the low-rank Burer-Monteiro factorization rather than the full matrix directly [20, 25].

3. SASD VIA ROCO

Our approach to addressing SaSD problem is based on the matrix lifting in (3) but uses a non-convex optimization formulation.

3.1. Optimization Formulation

Consider the rank-one matrix lifting of a convolution [14, 15]. In particular,

$$\boldsymbol{y} = \boldsymbol{x} \circledast \boldsymbol{h} = \mathcal{A}\left(\boldsymbol{x}\boldsymbol{h}^\top\right)$$
$$= \mathcal{A}(\boldsymbol{Z}) = \boldsymbol{A}\text{vec}(\boldsymbol{Z}), \tag{4}$$

where $\boldsymbol{Z} := \boldsymbol{x}\boldsymbol{h}^\top$ is clearly a rank-one matrix, the linear operator $\mathcal{A} : \mathbb{R}^{N \times M} \to \mathbb{R}^N$ can be represented by the matrix

$$\boldsymbol{A} := [\boldsymbol{I}_N, \mathcal{S}_1(\boldsymbol{I}_N), \cdots, \mathcal{S}_{M-1}(\boldsymbol{I}_N)] \in \mathbb{R}^{N \times MN}, \tag{5}$$

$\boldsymbol{I}_N$ is the $N \times N$ identity matrix, and $\mathcal{S}_\tau(\cdot)$ is the *downwards* cir-cular shift operator that circularly moves the rows of the input vec-tor/matrix downwards by τ many positions.

Our optimization formulation of blind deconvolution is based on the matrix lifting technique in (3). Suppose that $\boldsymbol{x}$ is sparse. A zero entry in $\boldsymbol{x}$, say x_n, leads to a zero row in $\boldsymbol{Z}$, i.e., $\boldsymbol{Z}_{n,:} = x_n\boldsymbol{h}^\top = \boldsymbol{0}^\top$. Hence, promoting sparsity in $\boldsymbol{x}$ is equivalent to enforcing row-sparsity in the matrix $\boldsymbol{Z}$. Define row-wise $\ell_{2,0}$-norm as

$$\|\boldsymbol{Z}\|_{r,2,0} := \left\| \left[\|\boldsymbol{Z}_{1,:}\|_2, \|\boldsymbol{Z}_{2,:}\|_2, \cdots, \|\boldsymbol{Z}_{N,:}\|_2 \right]^\top \right\|_0, \tag{6}$$

which counts the number of non-zero rows of the input matrix $\boldsymbol{Z}$. Further define the set of rank-one matrices as

$$\mathcal{R}1 = \left\{ \boldsymbol{Z} \in \mathbb{R}^{N \times M} : \text{rank}(\boldsymbol{Z}) \leq 1 \right\}. \tag{7}$$

Then the SaSD problem can be formulated as

$$\min_{\boldsymbol{Z}} \ \|\boldsymbol{Z}\|_{r,2,0}, \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\text{vec}(\boldsymbol{Z}), \ \boldsymbol{Z} \in \mathcal{R}1. \tag{8}$$

Further relax the non-convex ℓ_0 pseudo-norm into the convex ℓ_1-norm, yielding

$$\min_{\boldsymbol{Z}} \ \|\boldsymbol{Z}\|_{r,2,1}, \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{A}\text{vec}(\boldsymbol{Z}), \ \boldsymbol{Z} \in \mathcal{R}1, \tag{9}$$

where $\| \cdot \|_{r,2,1}$ is defined by replacing the ℓ_0 pseudo-norm in (6) by ℓ_1-nrom. It is important to note the non-convex rank constraint in (9), which marks the key difference between our approach and the convex counterpart (3).

The convolution operator $\boldsymbol{y} = \mathcal{A}(\boldsymbol{Z}) = \boldsymbol{A}\text{vec}(\boldsymbol{Z})$ can be eval-uated efficiently. The straightforward way is to write the matrix $\boldsymbol{A}$ explicitly, store it using a sparse matrix data type (for example in Matlab), and then compute $\boldsymbol{A}\text{vec}(\boldsymbol{Z})$. By contrast, we calculate $\boldsymbol{A}\text{vec}(\boldsymbol{Z})$ by introducing an auxiliary matrix

$$\boldsymbol{Z}_{\text{CL}} = \mathcal{S}_{\text{CL}}(\boldsymbol{Z}) := [\boldsymbol{Z}_{:,1}, \mathcal{S}_1(\boldsymbol{Z}_{:,2}), \cdots, \mathcal{S}_{M-1}(\boldsymbol{Z}_{:,M})],$$

where the subscript CL stands for *Cyclic Lifting*, meaning that $\boldsymbol{Z}_{\text{CL}}$ is not a direct matrix lifting $\boldsymbol{x}\boldsymbol{h}^\top$ but a cyclic shifted version of it. Then the convolution can be computed by summing the elements in each row of the auxiliary matrix $\boldsymbol{Z}_{\text{CL}}$, i.e.,

$$\boldsymbol{y} = \boldsymbol{x} \circledast \boldsymbol{h} = \mathcal{A}(\boldsymbol{Z}) = \mathcal{S}_{\text{CL}}(\boldsymbol{Z})\boldsymbol{1} = \boldsymbol{Z}_{\text{CL}}\boldsymbol{1}, \tag{10}$$

where $\boldsymbol{1} \in \mathbb{R}^M$ is the vector of which all elements are one. This method minimizes the computation and storage costs of evaluating $\mathcal{A}(\boldsymbol{Z})$, and also leads to the efficient update (18) of the ADMM solver developed in Section 3.2.

An ADMM solver is developed to solve (9) in Section 3.2. Note the developed ADMM operates on the full matrix $\boldsymbol{Z} \in \mathbb{R}^{N \times M}$. By the shortness of $\boldsymbol{h}$, $M \ll N$. Each iteration of ADMM has complexity $O(MN^2)$, which is comparable to that of iteration in methods based on sparse Lasso when $M = O(1)$.

After solving (9), SVD is used to extract $\boldsymbol{x}$ and $\boldsymbol{h}$ from $\boldsymbol{Z}$. Con-sider the SVD $\boldsymbol{Z} = \sigma\boldsymbol{u}\boldsymbol{v}^\top$ (as $\boldsymbol{Z}$ has rank one). Let $s \in \{-1, 1\}$ be the sign of the first nonzero entry in the vector $\boldsymbol{v}$. We set $\boldsymbol{x} = s\sigma\boldsymbol{u}$ and $\boldsymbol{h} = s\boldsymbol{v}$.

3.2. An ADMM solver for ROCO

We solve the ROCO optimization problem (9) by using an ADMM algorithm. As will be shown soon, each step of ADMM iterations admits a closed form solution, and hence the overall ADMM algo-rithm is computationally efficient.

The ADMM reformulation of (9) is given by

$$\min_{\boldsymbol{P},\boldsymbol{Q},\boldsymbol{Z}} \ \|\boldsymbol{P}\|_{r,2,1} + \mathbb{1}_{\mathrm{rank}(\cdot)\leq 1}(\boldsymbol{Q}) \qquad (11)$$

$$\text{s.t. } \boldsymbol{y} = \mathcal{S}_{\mathrm{CL}}(\boldsymbol{Z})\mathbf{1}, \ \boldsymbol{P} = \boldsymbol{Z}, \ \boldsymbol{Q} = \boldsymbol{Z},$$

where

$$\mathbb{1}_{\mathrm{rank}(\cdot)\leq 1}(\boldsymbol{Q}) := \begin{cases} 0, & \text{if } \mathrm{rank}(\boldsymbol{Z}) \leq 1, \\ +\infty, & \text{otherwise}, \end{cases}$$

is the indicator function of a matrix of which the rank is at most one. The corresponding augmented Lagrangian can be written as

$$\mathcal{L}_\rho = \|\boldsymbol{P}\|_{2,1} + \mathbb{1}_{\mathrm{rank}(\cdot)\leq 1}(\boldsymbol{Q}) + \frac{\rho}{2}\|\boldsymbol{Z} - \boldsymbol{P} + \boldsymbol{\Lambda}_1\|_F^2$$

$$+ \frac{\rho}{2}\|\boldsymbol{Z} - \boldsymbol{Q} + \boldsymbol{\Lambda}_2\|_F^2 + \frac{\rho}{2}\|\mathcal{S}_{CL}(\boldsymbol{Z})\mathbf{1} - \boldsymbol{y} + \boldsymbol{\lambda}_0\|_2^2$$

$$- \frac{\rho}{2}\|\boldsymbol{\Lambda}_1\|_F^2 - \frac{\rho}{2}\|\boldsymbol{\Lambda}_2\|_F^2 - \frac{\rho}{2}\|\boldsymbol{\lambda}_0\|_2^2,$$

where $\rho > 0$ denotes the penalty parameter, and $\boldsymbol{\Lambda}_1 \in \mathbb{R}^{N\times M}$, $\boldsymbol{\Lambda}_2 \in \mathbb{R}^{N\times M}$, and $\boldsymbol{\lambda}_0 \in \mathbb{R}^N$ are *normalized* Lagrange multipliers (for the purpose of notational simplification). Hence the ADMM iterations are given by

$$\boldsymbol{P}^{l+1} = \arg\min_{\boldsymbol{P}} \|\boldsymbol{P}\|_{r,2,1} + \frac{\rho}{2}\|\boldsymbol{Z}^l - \boldsymbol{P} + \boldsymbol{\Lambda}_1^l\|_F^2, \qquad (12)$$

$$\boldsymbol{Q}^{l+1} = \arg\min_{\boldsymbol{Q}} \mathbb{1}_{\mathrm{rank}(\cdot)\leq 1}(\boldsymbol{Q}) + \frac{\rho}{2}\|\boldsymbol{Z}^l - \boldsymbol{Q} + \boldsymbol{\Lambda}_2^l\|_F^2, \qquad (13)$$

$$\boldsymbol{Z}^{l+1} = \arg\min_{\boldsymbol{Z}} \|\boldsymbol{Z} - \boldsymbol{P}^{l+1} + \boldsymbol{\Lambda}_1^l\|_F^2 + \|\boldsymbol{Z} - \boldsymbol{Q}^{l+1} + \boldsymbol{\Lambda}_2^l\|_F^2$$

$$+ \|\mathcal{S}_{CL}(\boldsymbol{Z})\mathbf{1} - \boldsymbol{y} + \boldsymbol{\lambda}_0^l\|_2^2 \qquad (14)$$

$$\boldsymbol{\Lambda}_1^{l+1} = \boldsymbol{\Lambda}_1^l + \boldsymbol{Z}^{l+1} - \boldsymbol{P}^{l+1}, \qquad (15)$$

$$\boldsymbol{\Lambda}_2^{l+1} = \boldsymbol{\Lambda}_2^l + \boldsymbol{Z}^{l+1} - \boldsymbol{Q}^{l+1}, \qquad (16)$$

$$\boldsymbol{\lambda}_0^{l+1} = \boldsymbol{\lambda}_0^l + \mathcal{S}_{CL}(\boldsymbol{Z})\mathbf{1} - \boldsymbol{y}, \qquad (17)$$

where l is the index of iterations.

All the three sub-problems (12)-(14) involved in ADMM iterations admit closed form solutions. In particular, the closed form solution of (12) can be obtained by group soft-thresholding [26]. Define $\hat{\boldsymbol{P}} := \boldsymbol{Z}^l + \boldsymbol{\Lambda}_1^l$. Then the solution $\boldsymbol{P}^{l+1}$ to (12) is given by

$$\boldsymbol{P}_{n,:}^{l+1} = \left(1 - \frac{1}{\rho\|\hat{\boldsymbol{P}}_{n,:}\|_2}\right)_+ \hat{\boldsymbol{P}}_{n,:},$$

where $(x)_+ := \max(0, x)$.

Though the sub-problem (13) is non-convex, its optimal solution can be obtained based on Eckart-Young-Mirsky theorem [27]. Define $\hat{\boldsymbol{Q}} = \boldsymbol{Z}^l + \boldsymbol{\Lambda}_2^l$. Consider the Singular Value Decomposition (SVD) $\hat{\boldsymbol{Q}} = \boldsymbol{U}\mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{V}^\mathsf{T}$. The optimal solution of (13) is then given by

$$\boldsymbol{Q}^{l+1} = \boldsymbol{\sigma}_1 \boldsymbol{U}_{:,1} \boldsymbol{V}_{:,1}^T.$$

The sub-problem (14) is a quadratic optimization, of which the optimal solution can be obtained via pseudo-inverse or conjugate gradient method [28] in principle. On the other hand, the specific structure of this quadratic optimization problem allows a much more efficient way to compute its optimal solution. Define $\boldsymbol{C}_1^l := \boldsymbol{P}^{l+1} - \boldsymbol{\Lambda}_1^l$, $\boldsymbol{C}_2^l := \boldsymbol{Q}^{l+1} - \boldsymbol{\Lambda}_2^l$, $\boldsymbol{c}_3^l := \boldsymbol{y} - \boldsymbol{\lambda}_0^l$, and $\boldsymbol{C}_3^l := [\boldsymbol{c}_3^l, \cdots, \boldsymbol{c}_3^l] \in \mathbb{R}^{N\times M}$ (M repeated columns). Define $\mathcal{S}_{-\tau}(\cdot)$ as the *upwards* circular shift operator that circularly moves the rows of the input vector/matrix upwards by τ many positions. Define

$$\mathcal{S}_{\text{-CL}}(\boldsymbol{C}) = \left[\boldsymbol{C}_{:,1}, \mathcal{S}_{-1}(\boldsymbol{C}_{:,2}), \cdots, \mathcal{S}_{-(M-1)}(\boldsymbol{C}_{:,M})\right].$$

It can be verified by linear algebra and Woodbury matrix identity that the optimal solution to (14) is given by

$$\boldsymbol{Z}^{l+1} = \frac{1}{2}\mathcal{S}_{\text{-CL}}\left(\left(\mathcal{S}_{\mathrm{CL}}(\boldsymbol{C}_1^l + \boldsymbol{C}_2^l) + \boldsymbol{C}_3^l\right)\left(\boldsymbol{I}_M - \frac{1}{M+2}\mathbf{1}\mathbf{1}^\mathsf{T}\right)\right). \qquad (18)$$

It is noteworthy that the problem (11) that ADMM is applied to is non-convex. Generally speaking, ADMM algorithm for non-convex problems may not converge. Nevertheless, the above developed ADMM algorithm with sufficient large ρ converges in all of our numerical tests. Interested readers may refer to [29] for recent work on adapting ADMM to non-convex and non-smooth optimizations so that global convergence is guaranteed.

4. NUMERICAL TESTS

We compare the ROCO method with four benchmark algorithms: ADM, iADM, homotopy-ADM and homotopy-iADM [22].

4.1. Synthetic data tests

The setup for synthetic data tests follows the typical way similar to that in [22] and is specified as follows. We assume that the observed data $\boldsymbol{y} \in \mathbb{R}^N$ is generated from a convolution of a ground-truth sparse signal $\boldsymbol{x}^0 \in \mathbb{R}^N$ and a ground-truth kernel $\boldsymbol{h}^0 \in \mathbb{R}^M$ via $\boldsymbol{y} = \boldsymbol{x}^0 \circledast \boldsymbol{h}^0$. The kernel $\boldsymbol{h}^0$ is generated by first filling it with independent realizations of the standard Gaussian variable and then normalizing it to have unit ℓ_2-norm. Here we choose i.i.d. Gaussian entries to fill $\boldsymbol{h}$ for a relatively small shift-coherence. The sparse vector $\boldsymbol{x}^0$ is generated by Bernoulli-Gaussian distribution $\mathcal{BG}(\theta)$, i.e. the occurrence of the non-zero entries in $\boldsymbol{x}^0$ is under Bernoulli distribution with probability θ, and the value of non-zeros are under Gaussian distribution. In particular, we set the dimension of the kernel $\boldsymbol{h}^0$ as $M = 10$ and vary the sparsity ratio θ and number of samples N from 0.025 to 0.2 and from 100 to 800 respectively, where the values of θ chosen are adequate for the lower bound stated in [21]. The tests are repeated for 100 random trials. For ROCO, we set the penalty parameter $\rho = 2$. For all tested algorithms, we stop the optimization process either at 1000 iterations, or when the difference between two consecutive iterations in ℓ_2-norms is smaller than threshold 10^{-6}.

Performance metric includes recovery failure rate and average ℓ_2-error for successful recoveries. Denote $\hat{\boldsymbol{h}}$ as the estimated kernel. The recovery error is defined as

$$e_{\mathrm{Rec}} = \min \ \{\|s_\tau(\hat{\boldsymbol{h}}) - \boldsymbol{h}^0\|_2, \|s_\tau(\hat{\boldsymbol{h}}) + \boldsymbol{h}^0\|_2 : \ \tau \in [N-1]\}$$

to address both sign and shift ambiguities. Let $e_{\mathrm{Thr}} = 10^{-2}$. We call a trial with $e_{\mathrm{Rec}} \leq e_{\mathrm{Thr}}$ a successful recovery and otherwise a failed recovery. The failure recovery rate is depicted in Figure 1, where the number of trials is 100. A darker colour means a lower failure rate. It is straightforward to see that ROCO has a lower failure rate in the tested range of sparsity ratios θ and signal lengths N. The improvement of ROCO becomes more significant when the signal length N is relatively small.

We are also interested in the average error of successful recoveries, as it gives more detailed performance information. We focus on the case where $N = 800$. Results are presented in Figure 2. It is clear that ROCO substantially outperforms all other benchmark algorithms. Even for less sparse signals $\boldsymbol{x}$ where $\theta = 0.2$, the average error of successful recoveries is approximately only $1/10$ of that of other algorithms.

Table 1: Comparison of different SaSD algorithms applied to image deblurring. Note that the PSNR (in dB) values for the benchmark algorithms are computed using the shifts of their results.

Ground truth digits	Blurred digits	ROCO	ADM	iADM	homotopy-ADM	homotopy-iADM
PSNR for digit 5 (dB)		89.0427	20.3754	15.4896	20.0855	15.1360
PSNR for digit 0 (dB)		38.6332	17.8752	19.0489	16.4804	14.7691
PSNR for digit 9 (dB)		51.4815	22.8923	15.5526	16.1069	15.7989
PSNR for digit 3 (dB)		42.1422	13.5715	12.6990	12.7768	12.6003
Average running time (s)		2.198	1.836	1.565	2.776	2.978

4.2. Image deblurring

This test is based on real images from MNIST dataset. MNIST dataset contains images of handwritten digits from 0 to 9, each of which is of the size 28×28 pixels. We randomly choose 4 images (corresponding to 4 different digits) from MNIST dataset as ground truth images, denoted as $\boldsymbol{I}^0$. We also generate a convolutional kernel $\boldsymbol{h}^0 \in \mathbb{R}^9$ using the same way as in the synthetic data test. The observation $\boldsymbol{y}$ is a blurred image generated by a convolution of $\text{vec}(\boldsymbol{I}^0)$ and $\boldsymbol{h}^0$. The SaSD problem is then a single image deblurring problem. In simulations, we set $\rho = 20$ for ROCO method.

Table 1 summarises the simulation results. The performance metric is peak signal-to-noise ratio (PSNR) defined as

$$\text{PSNR} = \max \left\{ 10\log_{10} \frac{N}{\|S_\tau(\hat{\boldsymbol{I}}) - \boldsymbol{I}^0\|_F^2} : \forall \tau \in [\sqrt{N} - 1] \right\},$$

where $N = 28 \times 28$, $\sqrt{N} - 1 = 27$, and the term S_τ is introduced to address shift ambiguity. From the results, the recovered images via ROCO are visually much sharper than those from other methods, and actually look identical to the ground truth ones. In terms of PSNR, the performance of ROCO is at least 19dB better than those of other methods. It is important to note that the run time of ROCO is comparable to benchmark algorithms (sometimes less). This shows that matrix lifting by using our cyclic lifting method may not sacrifice computational complexity compared with factorization-based methods.

5. CONCLUSION

In this paper, we develop the ROCO method for the SaSD problem. The distinct characteristic is that ROCO operates on full rank-one matrices in both formulation and its ADMM solver. For an efficient ADMM implementation, closed forms are derived for the convolution operator and the ADMM update of the full rank-one matrix. Numerical tests are performed to compare ROCO with four benchmark algorithms based on bilinear Lasso. Results demonstrate substantial improvement in recovery accuracy and comparable runtime of ROCO compared with benchmark algorithms.
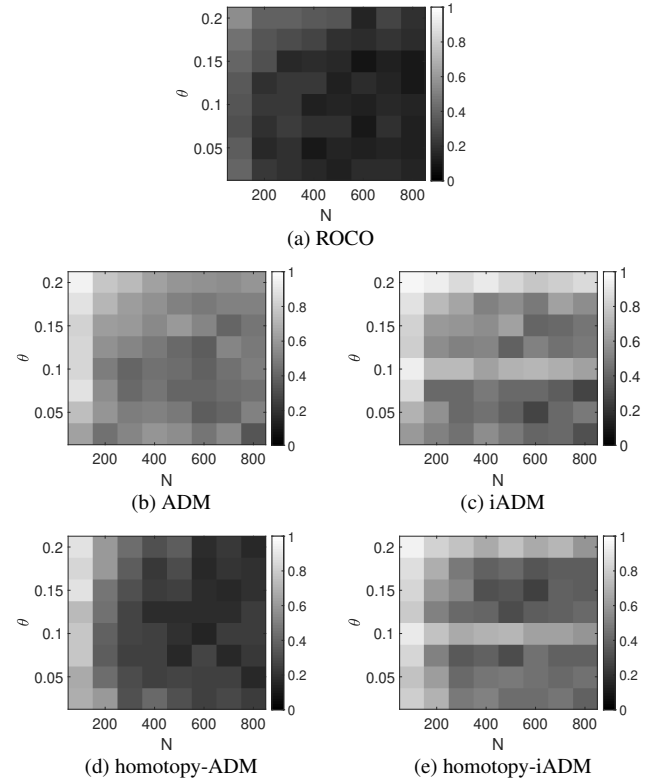

(a) ROCO

(b) ADM

(c) iADM

(d) homotopy-ADM

(e) homotopy-iADM

Fig. 1: Comparison of SaSD methods based on the failure rate of reconstructing $\boldsymbol{h}$. $M = 10$, $N = [100, 800]$, $\theta = [0.025, 0.2]$.

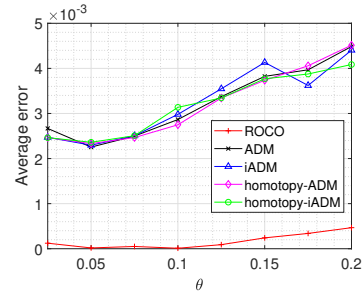**Fig. 2**: Comparison of SaSD methods based on the average error of successful reconstruction of $\boldsymbol{h}$. $M = 10$, $N = 800$, $\theta = [0.025, 0.2]$.

6. REFERENCES

[1] S. M. Jefferies and J. C. Christou, "Restoration of astronomical images by iterative blind deconvolution," *The Astrophysical Journal*, vol. 415, pp. 862, 1993.

[2] R. Molina, J. Núñez, F. J. Cortijo, and J. Mateos, "Image restoration in astronomy: a bayesian perspective," *IEEE Signal Processing Magazine*, vol. 18, no. 2, pp. 11–29, 2001.

[3] A. Jalobeanu, L. Blanc-Feraud, and J. Zerubia, "An adaptive gaussian model for satellite image deblurring," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 613–621, 2004.

[4] L. M. G. Fonseca, L. M. Namikawa, and E. F. Castejon, "Digital image processing in remote sensing," in *2009 Tutorials of the XXII Brazilian Symposium on Computer Graphics and Image Processing*. IEEE, 2009, pp. 59–71.

[5] T. Taxt and G. V. Frolova, "Noise robust one-dimensional blind deconvolution of medical ultrasound images," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 46, no. 2, pp. 291–299, 1999.

[6] O. Michailovich and A. Tannenbaum, "Blind deconvolution of medical ultrasound images: A parametric inverse filtering approach," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 3005–3019, 2007.

[7] G. R. Ayers and J. C. Dainty, "Iterative blind deconvolution method and its applications," *Optics letters*, vol. 13, no. 7, pp. 547–549, 1988.

[8] A. K. Katsaggelos and K. T. Lay, "Maximum likelihood blur identification and image restoration using the em algorithm," in *IEEE Trans. Sig. Proc*, 1980, vol. 39.

[9] A. C. Likas and N. P. Galatsanos, "A variational approach for bayesian blind image deconvolution," *IEEE transactions on signal processing*, vol. 52, no. 8, pp. 2222–2233, 2004.

[10] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *ACM SIGGRAPH 2006 Papers*, pp. 787–794. 2006.

[11] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1964–1971.

[12] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *European conference on computer vision*. Springer, 2010, pp. 157–170.

[13] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[14] A. Ahmed, B. Recht, and J. Romberg, "Blind deconvolution using convex programming," *IEEE Transactions on Information Theory*, vol. 60, no. 3, pp. 1711–1732, 2013.

[15] S. Choudhary and U. Mitra, "Sparse blind deconvolution: What cannot be done," in *2014 IEEE International Symposium on Information Theory*. IEEE, 2014, pp. 3002–3006.

[16] Daniele Perrone and Paolo Favaro, "Total variation blind deconvolution: The devil is in the details," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2909–2916.

[17] S. Ling and T. Strohmer, "Self-calibration and biconvex compressive sensing," *Inverse Problems*, vol. 31, no. 11, pp. 115002, 2015.

[18] Y. Chi, "Guaranteed blind sparse spikes deconvolution via lifting and convex optimization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 782–794, 2016.

[19] Y. Li, K. Lee, and Y. Bresler, "Identifiability in blind deconvolution with subspace or sparsity constraints," *IEEE Transactions on information Theory*, vol. 62, no. 7, pp. 4266–4275, 2016.

[20] Y. Zhang, Y. Lau, H. Kuo, S. Cheung, A. Pasupathy, and J. Wright, "On the global geometry of sphere-constrained sparse blind deconvolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4894–4902.

[21] H. Kuo, Y. Lau, Y. Zhang, and J. Wright, "Geometry and symmetry in short-and-sparse deconvolution," in *International Conference on Machine Learning*. PMLR, 2019, pp. 3570–3580.

[22] Y. Lau, Q. Qu, H. Kuo, P. Zhou, Y. Zhang, and J. Wright, "Short-and-sparse deconvolution–a geometric approach," *arXiv preprint arXiv:1908.10959*, 2019.

[23] Y. Zhang, H. Kuo, and J. Wright, "Structured local optima in sparse blind deconvolution," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 419–452, 2019.

[24] P. A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2009.

[25] S. Ling and T. Strohmer, "Regularized gradient descent: a nonconvex recipe for fast joint blind deconvolution and demixing," *Information and Inference: A Journal of the IMA*, vol. 8, no. 1, pp. 1–49, 2019.

[26] Ming Yuan and Yi Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.

[27] G. H. Golub, A. Hoffman, and G. W. Stewart, "A generalization of the eckart-young-mirsky matrix approximation theorem," *Linear Algebra and its applications*, vol. 88, pp. 317–327, 1987.

[28] J. Nocedal and S. J. Wright, "Conjugate gradient methods," *Numerical optimization*, pp. 101–134, 2006.

[29] Y. Wang, W. Yin, and J. Zeng, "Global convergence of admm in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, 2019.