MPTV: Matching Pursuit-Based Total Variation Minimization for Image Deconvolution

Dong Gong, Mingkui Tan, Qinfeng Shi, Anton van den Hengel, and Yanning Zhang

Abstract— **Total variation (TV) regularization has proven effective for a range of computer vision tasks through its preferential weighting of sharp image edges. Existing TV-based methods, however, often suffer from the over-smoothing issue and solution bias caused by the homogeneous penalization. In this paper, we consider addressing these issues by applying inhomogeneous regularization on different image components. We formulate the inhomogeneous TV minimization problem as a convex quadratic constrained linear programming problem. Relying on this new model, we propose a matching pursuit-based total variation minimization method (MPTV), specifically for image deconvolution. The proposed MPTV method is essentially a cutting-plane method that iteratively activates a subset of nonzero image gradients and then solves a subproblem focusing on those activated gradients only. Compared with existing methods, the MPTV is less sensitive to the choice of the trade-off parameter between data fitting and regularization. Moreover, the inhomogeneity of MPTV alleviates the over-smoothing and ringing artifacts and improves the robustness to errors in blur kernel. Extensive experiments on different tasks demonstrate the superiority of the proposed method over the current state of the art.**

Index Terms— **Total variation, image deconvolution, matching pursuit, convex programming.**

I. Introduction

MANY image restoration tasks can be formulated as an inverse problem, which allows the recovery of the latent image $\mathbf{x}$ from a measured image $\mathbf{y}$. The imaging model can be formulated as:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ denotes the latent image, $\mathbf{y} \in \mathbb{R}^n$ denotes the measured image, $\mathbf{A} \in \mathbb{R}^{n\times n}$ is a linear operator which models the measurement process, and $\mathbf{n} \in \mathbb{R}^n$ is an additive noise vector. Directly recovering $\mathbf{x}$ from model (1) is an ill-posed problem, because there are usually too many solutions. A proper prior or regularizer for $\mathbf{x}$ is important for reducing the ill-posedness of the problem. Given $\mathbf{y}$ and $\mathbf{A}$, and assuming $\mathbf{n}$ is sampled from i.i.d. Gaussian white noise, image restoration can be achieved by solving the following for $\mathbf{x}$:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda\Omega(\mathbf{x}), \tag{2}$$

where $\Omega(\cdot)$ is the regularizer on $\mathbf{x}$, and $\lambda > 0$ is a trade-off parameter. Many kinds of regularizer $\Omega(\cdot)$, such as the total variation (TV) norm [1], wavelet frame-based sparse priors [2] and Gaussian mixture models [3], have been proposed to handle different image restoration tasks, *e.g.* denoising, inpainting and deconvolution. In this paper, we focus on one particular task, namely, *non-blind image deconvolution*.

For image deconvolution [4], [5], the matrix $\mathbf{A}$ in model (1) and (2) represents a convolution matrix, with an embedded blur kernel (a.k.a. point spread function, PSF) $\mathbf{k}$. The degenerate image $\mathbf{y}$ is usually modeled by

$$\mathbf{y} = \mathbf{x} * \mathbf{k} + \mathbf{n},$$

where $*$ denotes the 2D convolution operator. *Non-blind deconvolution* seeks to recover $\mathbf{x}$ from the blurred image $\mathbf{y}$ given a known blur kernel $\mathbf{k}$ (and thus $\mathbf{A}$). The problem is ill-posed and non-trivial to address. The practical importance of the problem has motivated significant research attention [4], [6]–[8]. In particular, TV regularized methods have been intensively exploited and have demonstrated great success in image deconvolution [1], [9]–[11].

The TV-based model was first proposed by Rudin *et al.* [1] for image denoising [4], [9], [12]. It has subsequently been applied to a variety of other tasks including deconvolution [4], [13], [14], super-resolution [15], and inpainting [16]. By assuming that the image gradients are sparse, TV-based methods reflect the tendency of clear, sharp images towards piecewise smoothness. Although ℓ_0-norm regularization can be applied to promote sparse solution directly, the resultant optimization problem is difficult to solve due to its non-convexity. In practice, ℓ_1-norm or $\ell_{2,1}$-norm based TV regularization is more commonly used [4], [11]. In this paper, we consider the $\ell_{2,1}$-norm based

Manuscript received June 23, 2017; revised March 28, 2018, June 4, 2018, and August 15, 2018; accepted September 25, 2018. Date of publication October 10, 2018; date of current version December 5, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61231016 and Grant 61602185, and in part by the ARC Discovery Project under Grant DP160100703. The work of M. Tan was supported by the Recruitment Program for Young Professionals. The work of Y. Zhang was supported by the Chang Jiang Scholars Program of China under Grant 100017GH030150 and Grant 15GH0301. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jean-Francois Aujol. *(Dong Gong and Mingkui Tan contributed equally to this work.) (Corresponding author: Qinfeng Shi.)*

D. Gong and Y. Zhang are with the School of Computer Science and Engineering, Northwestern Polytechnical University, Xi’an 710129, China (e-mail: edgong01@gmail.com; ynzhang@nwpu.edu.cn).

M. Tan is with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China (e-mail: mingkuitan@scut.edu.cn).

Q. Shi and A. van den Hengel are with the School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: javen.shi@adelaide.edu.au; anton.vandenhengel@adelaide.edu.au).

Digital Object Identifier 10.1109/TIP.2018.2875352

isotropic TV norm [1]:

$$\Omega_{\mathrm{TV}}(\mathbf{x}) = \|\mathbf{D}\mathbf{x}\|_{2,1} = \sum\nolimits_{i=1}^{n} \| \; [(\mathbf{D}_v\mathbf{x})_i, (\mathbf{D}_h\mathbf{x})_i] \; \|_2, \tag{3}$$

where $\mathbf{D} = [\mathbf{D}_v^\mathsf{T}, \mathbf{D}_h^\mathsf{T}]^\mathsf{T}$, and $\mathbf{D}_v \in \mathbb{R}^{n\times n}$ and $\mathbf{D}_h \in \mathbb{R}^{n\times n}$ denote the first-order difference matrices in vertical and horizontal directions, respectively. $\|\cdot\|_{2,1}$ denotes the $\ell_{2,1}$-norm. Specifically, $[(\mathbf{D}_v\mathbf{x})_i, (\mathbf{D}_v\mathbf{x})_i]$ denotes a 1-by-2 vector concatenating the i-th element of $\mathbf{D}_v\mathbf{x}$ and $\mathbf{D}_h\mathbf{x}$.

A variety of TV-based methods for reducing the impact of both image noise and blur have been devised. These methods tend to suffer from a common set of deficiencies, however. Firstly, the homogeneous penalty may cause over-smoothing of high-frequency image components (such as edges and corners). Secondly, due to the regularization bias [17]–[19] rooted in the specific ℓ_1-norm or $\ell_{2,1}$-norm, it is difficult to select an appropriate trade-off hyper-parameter λ to achieve a solution that has sparse gradients and suffers less bias issue. Thirdly, existing TV-based deconvolution methods are usually sensitive to errors or noise in $\mathbf{k}$ (*i.e.* $\mathbf{A}$), and this often causes heavy ringing artifacts in the latent image $\mathbf{x}$ [20], [21]. For existing TV methods, it is non-trivial to achieve a balance between suppressing artifacts and preserving image details.

In this paper, we propose a matching pursuit algorithm, called MPTV, for solving TV regularization based image deconvolution. The paper extends our conference paper [22] where we addressed the generalized lasso problem. The new contributions of this paper are summarized as follows.

- We reformulate the TV minimization problem as a quadratically constrained linear programming (QCLP) problem and then propose a matching pursuit algorithm called MPTV to address the resultant problem. Instead of focusing on all the image gradients at the beginning, we iteratively invoke the most beneficial gradient subset, followed by solving a subproblem constrained on selected subsets only.
- The proposed MPTV method is able to alleviate the over-smoothing issue while yielding a solution with sparse gradients due to the matching pursuit like optimization strategy. Compared to existing methods, MPTV can help to reduce the solution bias. Moreover, the proposed optimization scheme also helps to improve the robustness of the method to noise and/or errors in $\mathbf{A}$. In particular, for the task of image deconvolution, MPTV helps to suppress the ringing artifacts and recover the image details.
- Many TV-based methods require an extensive and often imprecise selection of hyper-parameters to balance the sparsity and the fitness to observation. This issue can be significantly alleviated in the proposed MPTV method due to the new formulation and optimization strategy, particularly the early stopping strategy that will be introduced. In other words, MPTV is less sensitive to the selection of the regularization parameter.

The remainder of this paper is organized as follows. In Section II, we review related work in TV models and image deconvolution. In Section III, we introduce our matching pursuit based TV minimization (MPTV) algorithm and discuss the details, including stopping conditions, parameter setting, and convergence analysis. Empirical studies on various datasets are presented in Section V. We conclude in Section VI.

II. Notation and Related Work

A. Notation

Let $\mathbf{A} = [A_{i,j}] \in \mathbb{R}^{m\times n}$ and $\mathbf{v} = [v_1, \ldots, v_n]^\mathsf{T} \in \mathbb{R}^n$ denote a matrix and a vector, receptively, where $^\mathsf{T}$ denotes the transpose of a vector/matrix. Let $\mathbf{0}$ and $\mathbf{1}$ be vectors with all zeros and all ones, respectively, and let $\mathbf{I}$ denote the identity matrix. Let $\mathbf{A}_i$ or $\mathbf{A}^i$ be a matrix indexed for some purpose, and let $\mathbf{v}^i$ or $\mathbf{v}_i$ be a vector indexed for some purpose.

Given a vector $\mathbf{v}$, let $\mathrm{diag}(\mathbf{v})$ be a diagonal matrix with diagonal elements equal to the vector $\mathbf{v}$, and $\|\mathbf{v}\|_p$ be the ℓ_p-norm. Let $\odot$ denote the element-wise (Hadamard) product, $\otimes$ denote the Kronecker product and $\mathrm{supp}(\mathbf{v})$ denote the support set of $\mathbf{v}$. Given a positive integer n, let $[n] = \{1, \ldots, n\}$. Given any index set $\mathcal{T} \subseteq [n]$, let $\mathcal{T}^c$ be the complementary set of $\mathcal{T}$, *i.e.* $\mathcal{T}^c = [n] \setminus \mathcal{T}$, and $\mathrm{card}(\mathcal{T})$ be the cardinality of $\mathcal{T}$. For a vector $\mathbf{v} \in \mathbb{R}^n$, let v_i denote the i-th element of $\mathbf{v}$, and $\mathbf{v}_\mathcal{T}$ denote the subvector indexed by $\mathcal{T}$. For a matrix $\mathbf{A}$, let $\mathbf{A}_\mathcal{T}$ denote the columns of $\mathbf{A}$ with indeces in $\mathcal{T}$.

B. Total Variation Models

Total variation regularization has been the subject of much attention over the past two decades, not least due to its effectiveness in image processing [4], [12], compressive sensing [9] and machine learning tasks [22], [23]. Considering the piecewise smooth property of images, Rudin *et al.* [1] proposed the TV regularizer for image denoising. Following that, many variants of the original TV regularizer have been extensively studied. Beside the isotropic TV model proposed in [1], anisotropic TV has also been wildly explored [4], [9].

To alleviate the optimization difficulties caused by the non-differentiability of TV models, some approximate TV models have been proposed, *e.g.* smooth TV [24] and Huber-norm based TV [25]. To preserve the sparse nature of the image gradients, the ℓ_0-norm based TV regularizer was proposed for edge-preserving image editing tasks [26] and blur kernel estimation [27]. Motivated by the $\ell_1 - \ell_2$-norm in compressive sensing, Lou *et al.* [11] proposed a weighted difference of anisotropic and isotropic TV regularizers to alleviate the over-smoothing incurred by classical TV methods. For these methods, achieving acceptable performance is critically dependent on careful turning of hyper-parameters. In fact, due to the bias nature in regularization, it is non-trivial to reduce bias issue while pursing sparsity. To handle different types of noises, different data fitting functions are also studied. Classical TV models [1], [4] use ℓ_2 loss to fit the Gaussian noise in observation. To improve the robustness of the model to outliers, non-smooth loss functions, including ℓ_1 loss [28], [29], ℓ_∞ loss [30] and ℓ_0 loss [12], have been used to handle Laplacian noise, uniform noise and impulse noise, respectively.

Various optimization strategies for TV regularized problems have also been extensively investigated. Rudin *et al.* [1] minimize the TV model using a gradient projection method, which, converges slowly due to the non-smoothness of the problem. Osher *et al.* [31] proposed an iterative regularization method based on Bregman distance minimization for solving the TV based image restoration problem. Burger [32] discussed a series of Bregman distance based approaches for inverse problems and pointed out that the Bregman iteration scheme is beneficial for alleviating the bias caused by the TV regularizer. In the past two decades, many other techniques have been proposed to accelerate convergence, including the primal-dual interior point method [24], the splitting Bregman method [9], the half-quadratic formulation based method [4], and alternating direction multiplier methods [10]. Specifically, to handle the spatially inhomogeneous structural and textural information in the image, some approaches proposed to use spatially varying constraints with the TV regularization [33], [34]. Gilboa *et al.* [33] proposed a variational framework for image denoising, in which spatially varying constraints are used for different image local areas. In [34], a spatially dependent parameter selection scheme is proposed to achieve spatially adaptive TV regularization for image restoration.

C. Image Deconvolution

Abundant studies have been conducted on (non-blind) image deconvolution, starting from the classical ones including Wiener filter and Richadson-Lucy (RL) algorithms [6]. Based on the piecewise smooth property of natural images and statistical studies of image gradients, many methods perform image deconvolution by preserving significant edges or sparse gradients. To this end, a series of TV based methods have been extensively exploited [4], [11], [12]. Additionally, in [36], an iterative reweighed least square (IRLS) has been used to achieve excellent results by encouraging the piecewise smooth property in image deconvolution. Krishnan and Fergus [7] proposed to perform image deconvolution using a hyper-Laplacian prior. Moreover, instead of preserving sparse gradients, many other methods also perform well by relying on patch similarity [37], kernel similarity [38], progressive multi-scale deconvolution [39], group based sparse representation [40], or stochastic optimization [41].

Besides the empirically designed models above, some methods have been proposed to learn models for image deconvolution [3], [42]–[45]. The generative prior learning methods [3], [46] focus on learning image priors/regularizers (*e.g.* a Gaussian Mixture Model based prior in [3]) from a set of clear and sharp images, which usually leads to non-convex problems and time-consuming optimization. For the sake of computational efficiency, some discriminative learning based methods [42]–[44] learn the deconvolution models (*e.g.* a convolutional neural network based model [43]) relying on pairs of sharp and blurred images. The learning based deconvolution methods are effective to utilize more information from training examples than the empirically designed approaches. However, suffering from limited flexibility on the model design, existing learning based methods [42], [43] often require customized training for specific blur kernels or noise, which limits their practicability.

In practice, image deconvolution results often suffer from wave-like ringing artifacts, especially in regions near strong edges. This issue may be caused by the unavoidable error in blur kernel estimation [20], the Gibbs phenomenon [47], the zero values in the frequency spectrum of the blur kernel [21] and/or the mismatch between the data and the model (*e.g.* non-conforming noise, saturation) [48], [49]. To reduce these artifacts in deconvolution, Yuan *et al.* [39] performed the Richardson-Lucy algorithm in a residual inter-and-intra-scale scheme with edge-preserving bilateral filters. Shan *et al.* [20] proposed to combin a smoothness constraint with the ℓ_1-norm regularizer. Mosleh *et al.* [21] proposed a post processing method to detect and suppress ringing based on frequency analysis. Some methods tried to avoid artifacts using more accurate imaging models and data fitting terms [29], [48], [49].

III. Matching Pursuit Total Variation for Image Deconvolution

In this section, we present a matching pursuit based TV minimization method for non-blind image deconvolution.

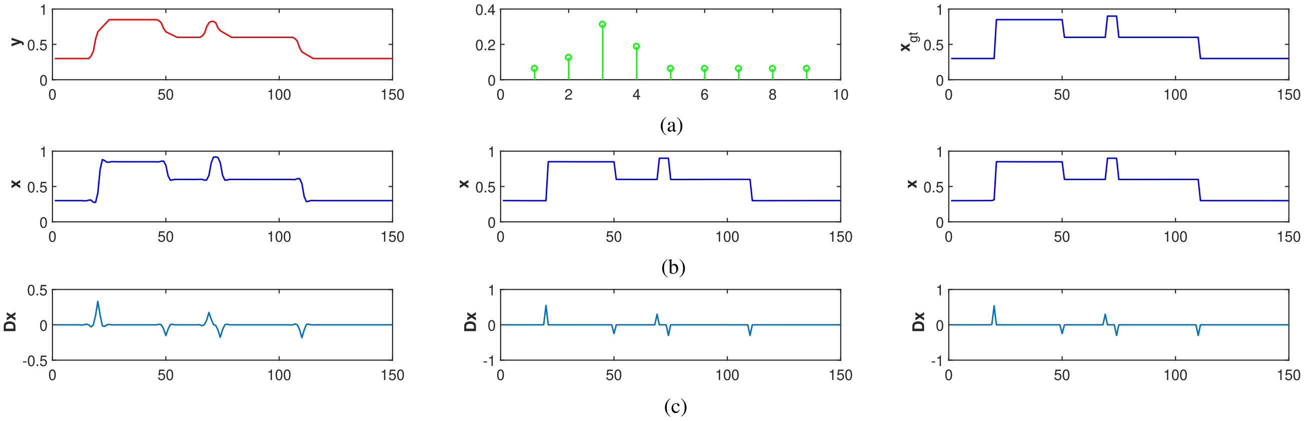
A. A QCLP Reformulation of TV Model

Image deconvolution is rendered more challenging by the ill-posed nature of the problem and the noise in both the blurred image and the estimated blur kernel. With the TV regularizer introduced in (3), the clear and sharp image can be recovered by solving the following classic TV-norm regularized optimization problem:

$$\min_{\mathbf{x}} \ \frac{1}{2}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2+\lambda\Omega_{\mathrm{TV}}(\mathbf{x}), \tag{4}$$

where $\Omega_{\mathrm{TV}}(\mathbf{x}) = \|\mathbf{D}\mathbf{x}\|_{2,1} = \sum_{i=1}^{n} \| \ [(\mathbf{D}_v\mathbf{x})_i, (\mathbf{D}_h\mathbf{x})_i] \ \|_2$. In problem (4), the TV regularizer treats all elements in $\mathbf{x}$ homogeneously. Although the $\ell_{2,1}$-norm is used for inducing sparsity on image gradients, it tends to shrink the large elements in image gradients towards zero [19]. As a result, while the TV regularizer in (4) favors a piecewise smooth solution, it may incur over-smoothing deconvolution results [11] due to the bias issue [18], [19]. We thus consider minimizing the total variation value by explicitly detecting and only preserving the sparse nonzero gradients, while suppressing the insignificant components. In this way, we can maintain the significant high-frequency elements in the image and suppress the noise. To illustrate the above issue, we show an example for 1D signal deconvolution in Fig. 1. This experiment shows that minimizing TV with active gradient detection can reduce the regularization bias significantly, and thus alleviates the over-smoothness and increases estimation accuracy accordingly.

To explicitly detect the sparse nonzero image gradients in image deconvolution, we will formulate the TV model with a new binary vector to indicate the nonzero gradients, which will be estimated with the image simultaneously. Firstly, we introduce $\mathbf{z} = [\mathbf{z}_v^\mathsf{T}, \mathbf{z}_h^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{2n}$ to denote the concatenation of the gradients of two directions, namely $\mathbf{D}_v\mathbf{x}$ and $\mathbf{D}_h\mathbf{x}$. To find the most significant image gradients which contribute



(a)

(b)

(c)

Fig. 1. Deconvolution results on a 1D signal example. (a) The blurred signal $\mathbf{y}$ is generated by blurring the sharp signal $\mathbf{x}_{\text{gt}}$ with the blur kernel $\mathbf{k}$. In (b) and (c), the results from left to right are generated by solving the TV regularized model using ADMM [35] (TV-ADMM), solving problem (6) using the proposed MPTV method ($\kappa = 1$), and solving problem (6) using ADMM given the support of the ground truth signal $\mathbf{x}_{\text{gt}}$, respectively. (c) shows the gradients of the recovered signal in (b). In each case, $\lambda = 0.01$.

the most to the quality of image deconvolution, we introduce two binary vectors $\boldsymbol{\tau}_v \in \{0,1\}^n$ and $\boldsymbol{\tau}_h \in \{0,1\}^n$ to indicate the nonzero ones in $\mathbf{D}_v\mathbf{x}$ and $\mathbf{D}_h\mathbf{x}$, respectively. We further define $\boldsymbol{\tau}_v = \boldsymbol{\tau}_h \triangleq \boldsymbol{\tau}$, since each term in the $\ell_{2,1}$-norm based TV regularizer consists of two elements over vertical and horizontal directions, *e.g.* $(\mathbf{D}_v\mathbf{x})_i$ and $(\mathbf{D}_h\mathbf{x})_i$ shown in (3). We then let $\widetilde{\boldsymbol{\tau}} = [\boldsymbol{\tau}^\top, \boldsymbol{\tau}^\top]^\top$ indicate the nonzero components in $\mathbf{Dx}$ via $(\mathbf{z} \odot \widetilde{\boldsymbol{\tau}})$. To induce the sparsity, we impose an ℓ_0-norm constraint $\|\boldsymbol{\tau}\|_0 \leq \kappa$. For simplicity, let $\Lambda = \{\boldsymbol{\tau} | \boldsymbol{\tau} \in \{0,1\}^n, \|\boldsymbol{\tau}\|_0 \leq \kappa\}$ be the feasible domain of $\boldsymbol{\tau}$.

With the introduction of $\boldsymbol{\tau}$ and the auxiliary variable $\mathbf{z}$, we propose to address the following TV-norm regularized problem:

$$\begin{aligned} &\min_{\boldsymbol{\tau}\in\Lambda} \min_{\mathbf{x},\boldsymbol{\xi},\mathbf{z}} \frac{1}{2}\|\boldsymbol{\xi}\|_2^2 + \lambda \sum\nolimits_{i=1}^{n} \| \, [(\mathbf{z}_v)_i, (\mathbf{z}_h)_i] \, \|_2 \\ &\text{s.t.} \boldsymbol{\xi} = \mathbf{y} - \mathbf{Ax}, \quad \mathbf{Dx} = \mathbf{z} \odot \widetilde{\boldsymbol{\tau}}, \end{aligned} \tag{5}$$

where $(\mathbf{z}_v)_i$ and $(\mathbf{z}_h)_i$ denote the i-th elements of $\mathbf{z}_v$ and $\mathbf{z}_h$, respectively. The constraint $\|\boldsymbol{\tau}\|_0 \leq \kappa$ explicitly constrains the sparsity of gradients via $\mathbf{Dx} = \mathbf{z} \odot \widetilde{\boldsymbol{\tau}}$, leading to an image $\mathbf{x}$ with sparse $\mathbf{Dx}$ and a small total variation.

To simplify the computation of $\sum_{i=1}^{n} \| \, [(\mathbf{z}_v)_i, (\mathbf{z}_h)_i] \, \|_2$, we define binary matrices $\mathbf{C}_i \in \{0,1\}^{2\times 2n}, \forall i \in [n]$ to select both $(\mathbf{z}_v)_i$ and $(\mathbf{z}_h)_i$, $\forall i \in [n]$ from $\mathbf{z}$ via multiplication between $\mathbf{C}_i$ and $\mathbf{z}$, *i.e.* $\mathbf{C}_i\mathbf{z}$. As an example, there is $\mathbf{C}_i\mathbf{z} = [(\mathbf{z}_v)_i, (\mathbf{z}_h)_i]^\top$. In each $\mathbf{C}_i$, the i-th and $(i+n)$-th elements on the two rows, respectively, are 1, and the rest are 0. Then, the $\ell_{2,1}$-norm in problem (5) can be rewritten as $\sum_{i=1}^{n} \|\mathbf{C}_i\mathbf{z}\|_2$. Moreover, we introduce a new vector $\mathbf{d}$ to represent $\mathbf{z}$, and let its subvectors $\mathbf{d}_i = \mathbf{C}_i\mathbf{z}, \forall i \in [n]$. Then, we can obtain an equivalent formulation of problem (5):

$$\begin{aligned} &\min_{\boldsymbol{\tau}\in\Lambda} \min_{\mathbf{x},\boldsymbol{\xi},\mathbf{z},\mathbf{d}} \frac{1}{2}\|\boldsymbol{\xi}\|_2^2 + \lambda \sum\nolimits_{i=1}^{n} \|\mathbf{d}_i\|_2 \\ &\text{s.t. } \boldsymbol{\xi} = \mathbf{y} - \mathbf{Ax}, \quad \mathbf{Dx} = \mathbf{z} \odot \widetilde{\boldsymbol{\tau}}, \\ &\qquad \mathbf{d}_i = \mathbf{C}_i\mathbf{z}, \quad \forall i \in [n]. \end{aligned} \tag{6}$$

In problem (6), the integer κ reflects our rough knowledge of the sparsity of $\mathbf{Dx}$. Note that there are $|\Lambda| = \sum_{i=0}^{\kappa} \binom{n}{i}$ feasible $\boldsymbol{\tau}$'s in Λ. Problem (6) tends to find the optimal $\boldsymbol{\tau}$ from Λ to minimize the objective in (6). Even though the sparsity constraint $\|\boldsymbol{\tau}\|_0 \leq \kappa$ explicitly induces sparsity in image gradients, the regularizer $\lambda \sum_{i=1}^{n} \|\mathbf{d}_i\|_2$ is still necessary for the robust recovery of $\mathbf{z}$ due to possible noise in $\mathbf{y}$. Meanwhile, benefiting from the sparsity constraint and the optimization scheme that will be introduced, a small λ can be used to reduce the bias induced by the regularization [50], [51], which is beneficial in alleviating over-smoothness.

Unfortunately, problem (6) is a mixed integer programming problem, and thus hard to address. We consider a *convex relaxation* to this problem. To achieve this, we first introduce the following proposition.

Proposition 1: By deriving the dual form of the inner minimization problem in (6) *w.r.t.* $\mathbf{x}$, $\boldsymbol{\xi}$, $\mathbf{z}$ *and* $\mathbf{d}$ *given fixed* $\boldsymbol{\tau} \in \Lambda$, *problem* (6) *can be transformed into*

$$\begin{aligned} &\min_{\boldsymbol{\tau}\in\Lambda} \max_{\boldsymbol{\alpha}} \; -\frac{1}{2}\|\boldsymbol{\alpha}\|_2^2 + \boldsymbol{\alpha}^\top\mathbf{y} \\ &\textit{s.t. } \mathbf{A}^\top\boldsymbol{\alpha} = \mathbf{D}^\top\boldsymbol{\beta}, \\ &\qquad \tau_i\|\boldsymbol{\beta}_i\|_2 \leq \lambda, \quad \boldsymbol{\beta}_i = \mathbf{C}_i\boldsymbol{\beta}, \; \forall i \in [n]. \end{aligned} \tag{7}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^n$, $\boldsymbol{\beta}_v \in \mathbb{R}^n$, $\boldsymbol{\beta}_h \in \mathbb{R}^n$, $\boldsymbol{\beta} = [\boldsymbol{\beta}_v^\top, \boldsymbol{\beta}_h^\top]^\top$ *are the Lagrangian dual variables. For any* $\boldsymbol{\tau} \in \Lambda$, $\boldsymbol{\alpha}^* = \boldsymbol{\xi}^*$ *at the optimum of the inner problem.*

The proof can be found in the supplementary material.

From Proposition 1, $\boldsymbol{\alpha}^*$ equals to the fitting error $\boldsymbol{\xi}^*$. Without loss of generality, we assume $\boldsymbol{\alpha} \in \mathcal{A} = [-l, l]^n$, where l is a sufficiently large positive number and $\mathcal{A}$ is a compact domain. Based on (7), the feasible domain of $\boldsymbol{\alpha}$ w.r.t. each $\boldsymbol{\tau}$ is $\mathcal{A}_{\boldsymbol{\tau}} = \{\boldsymbol{\alpha} | \mathbf{A}^\top\boldsymbol{\alpha} = \mathbf{D}^\top\boldsymbol{\beta}, \tau_i\|\boldsymbol{\beta}_i\|_2 \leq \lambda, \forall i \in [n], \boldsymbol{\alpha} \in [-l, l]^n\}$. Note that, according to the constraints in (7), all $\mathcal{A}_{\boldsymbol{\tau}}$'s share the same $\boldsymbol{\beta}$. We further define

$$\phi(\boldsymbol{\alpha}, \boldsymbol{\tau}) = \frac{1}{2}\|\boldsymbol{\alpha}\|_2^2 - \boldsymbol{\alpha}^\top\mathbf{y}, \quad \boldsymbol{\alpha} \in \mathcal{A}_{\boldsymbol{\tau}}. \tag{8}$$

By applying the minimax inequality in [52], we have

$$\min_{\boldsymbol{\tau}\in\Lambda} \max_{\boldsymbol{\alpha}\in\mathcal{A}} -\phi(\boldsymbol{\alpha}, \boldsymbol{\tau}) \geq \max_{\boldsymbol{\alpha}\in\mathcal{A}} \min_{\boldsymbol{\tau}\in\Lambda} -\phi(\boldsymbol{\alpha}, \boldsymbol{\tau}). \tag{9}$$

Algorithm 1: Cutting-Plane for the QCLP Problem (10)

```
**Input**: Observation $\mathbf{y}$, $\mathbf{A}$, parameter $\lambda$.
Initializing $\boldsymbol{\alpha}^0 = \mathbf{y} - (\mathbf{1} \otimes (\sum_{i=1}^{n} y_i/n))$, and $\boldsymbol{\tau}_0 = \mathbf{0}$;
Set $\Lambda_0 = \emptyset$ and $t = 1$;
**while** *Stopping conditions are not achieved* **do**
    Find the $\boldsymbol{\tau}_t$ corresponding to **the most violated constraint** based on $\boldsymbol{\alpha}^{t-1}$;
    Set $\Lambda_t = \Lambda_{t-1} \cup \{\boldsymbol{\tau}_t\}$;
    **Solve the subproblem** corresponding to $\Lambda_t$:
        $\min_{\boldsymbol{\alpha} \in \mathcal{A}, \theta \in \mathbb{R}} \ \theta, \quad \text{s.t. } \phi(\boldsymbol{\alpha}, \boldsymbol{\tau}) - \theta \leq 0, \forall \boldsymbol{\tau} \in \Lambda_t, \quad (11)$
    obtaining the solution $\boldsymbol{\alpha}^t$. Let $t = t + 1$;
```

According to (9), $\max_{\boldsymbol{\alpha} \in \mathcal{A}} \min_{\boldsymbol{\tau} \in \Lambda} -\phi(\boldsymbol{\alpha}, \boldsymbol{\tau})$ is a lower bound of problem (7) and it is a convex problem. By introducing a new variable $\theta \in \mathbb{R}$, $\max_{\boldsymbol{\alpha} \in \mathcal{A}} \min_{\boldsymbol{\tau} \in \Lambda} -\phi(\boldsymbol{\alpha}, \boldsymbol{\tau})$ can be equivalently written as a *quadratically constrained linear programming* (QCLP) problem [53]:

$$\min_{\boldsymbol{\alpha} \in \mathcal{A}, \theta} \ \theta, \quad \text{s.t. } \phi(\boldsymbol{\alpha}, \boldsymbol{\tau}) \leq \theta, \quad \forall \boldsymbol{\tau} \in \Lambda, \tag{10}$$

which is a convex relaxation of problem (7). Note that each feasible $\boldsymbol{\tau}$ corresponds to a constraint. Note that there are $T = \sum_{i=0}^{\kappa} \binom{n}{i}$ elements in Λ. In other words, problem (10) has exponentially many constraints, which makes it difficult to address directly.

B. Optimization of the QCLP TV Model

Though there are exponentially many constraints in (10), most of them are inactive at the optimum, since only a subset of components are relevant in fitting the observation. Accordingly, we seek to address problem (10) using a *cutting-plane method* [50], [54] as shown in Algorithm 1. Instead of handling all constraints at the same time, Algorithm 1 iteratively finds the active constraints and then solves a subproblem (11) with the selected active constraints only.

Algorithm 1 involves two main steps: finding the most violated constraints and the subproblem optimization step. Since each $\boldsymbol{\tau} \in \Lambda$ corresponds to a constraint, in the t-th iteration, we find the most active $\boldsymbol{\tau}_t$ based on $\boldsymbol{\alpha}^{t-1}$ and add it into the active constraint set Λ_t, which is initialized as an empty set $\emptyset$. Then we update $\boldsymbol{\alpha}^t$ by solving the subproblem (11) with the constraints defined in Λ_t. The algorithm terminates when the stopping conditions are achieved.

As will be shown later, in each iteration t, the activated $\boldsymbol{\tau}_t$ indicates at most κ nonzero elements in $\mathbf{D}_v\mathbf{x}$ and $\mathbf{D}_h\mathbf{x}$. Activating the most active $\boldsymbol{\tau}_t$ is equivalent to finding κ indices of the corresponding nonzero elements, which can be recorded in an index set $\mathcal{C}_t \subseteq [n]$, *i.e.* $\mathcal{C}_t = \text{supp}(\boldsymbol{\tau}_t)$. For convenience, we define an index set $\mathcal{S}_t = \cup_{i=1}^{t} \mathcal{C}_i$ to record the indices indicated by all $\boldsymbol{\tau}$'s in Λ_t. As shown in Section III-C, due to the difficulty of directly solving the subproblem (11) w.r.t. the dual variable $\boldsymbol{\alpha}$ [50], [55], we will investigate an efficient way to solve the subproblem w.r.t. to the primal variable $\mathbf{x}$ instead.

C. Solving QCLP Subproblem in Primal Form

In each iteration of Algorithm 1, after updating the active constraint set, we address subproblem (11) with a subset of activated $\boldsymbol{\tau}$'s in Λ_t. Although the number of constraints in (11) is significantly smaller than the original problem (10), directly optimizing it w.r.t. the dual variable $\boldsymbol{\alpha}$ is not easy [50], [55]. However, considering that the activated $\boldsymbol{\tau}$'s in Λ_t indicate only a subset of nonzero elements in $\mathbf{z}$ and $\mathbf{Dx}$, we will transform problem (11) to an equivalent problem (w.r.t. the primal variables $\mathbf{x}$) that can be solved much faster [22], [55].

Recall that $\mathcal{S}_t \subseteq [n]$ is defined to record the indices of the nonzero elements indicated by $\boldsymbol{\tau}$'s in Λ_t, and $\widetilde{\boldsymbol{\tau}} = [\boldsymbol{\tau}^\mathsf{T}, \boldsymbol{\tau}^\mathsf{T}]^\mathsf{T} \in \{0, 1\}^{2n}$ in (5) is defined to indicate nonzero elements in $\mathbf{Dx} \in \mathbb{R}^{2n}$. For convenience, we extend the definition of $\mathcal{S}_t$ for $\boldsymbol{\tau}$ to a notation $\widetilde{\mathcal{S}}_t$ for $\widetilde{\boldsymbol{\tau}}$, *i.e.* , for an $\mathcal{S}_t$, there is $\widetilde{\mathcal{S}}_t = \mathcal{S}_t \cup (\mathcal{S}_t + n)$ and $(\mathcal{S}_t + n) = \{i | i = j + n, j \in \mathcal{S}_t\}$. Armed with the definition of $\widetilde{\mathcal{S}}_t$, we transform problem (11) into problem (12) w.r.t. the primal variables $\mathbf{x}$ and $\mathbf{z}_{\widetilde{\mathcal{S}}_t}$ as shown in Proposition 2.

Proposition 2: Let $\mathcal{S} = \cup_{i=1}^{t} \mathcal{C}_i$. Assume there is no overlapping element among $\mathcal{C}_i$'s, problem (11) *can be addressed by solving*

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}_{\widetilde{\mathcal{S}}}} \ & \frac{1}{2}\|\mathbf{y} - \mathbf{Ax}\|_2^2 + \lambda \sum\nolimits_{i \in \mathcal{S}} \|\mathbf{C}_{i\widetilde{\mathcal{S}}}\mathbf{z}_{\widetilde{\mathcal{S}}}\|_2 \\ \textit{s.t. } & (\mathbf{Dx})_{\widetilde{\mathcal{S}}} = \mathbf{z}_{\widetilde{\mathcal{S}}}, \quad (\mathbf{Dx})_{\widetilde{\mathcal{S}}^c} = \mathbf{0}. \end{aligned} \tag{12}$$

Additionally, the optimal value of $\boldsymbol{\alpha}^*$ under problem (11) *can be recovered by $\boldsymbol{\alpha}^* = \boldsymbol{\xi}^*$ where $\boldsymbol{\xi}^* = \mathbf{y} - \mathbf{Ax}^*$.*
The proof can be found in the supplementary material. In (12), the subscript t of $\widetilde{\mathcal{S}}_t$ is omitted for simplifying the representation.

Proposition 2 implies that activating $\boldsymbol{\tau}$'s in the QCLP problem in (10) corresponds to activating a subset of nonzero image gradients in the TV minimization problem in primal, which are indexed by $\mathcal{S}_t$. The nonzero gradients activated in each iteration of Algorithm 1 are recorded in $\mathcal{C}_t$.

In problem (12), only a subset of image gradients (*i.e.* $\mathbf{Dx}_{\widetilde{\mathcal{S}}}$ and $\mathbf{z}_{\widetilde{\mathcal{S}}}$) indicated by $\widetilde{\mathcal{S}}$ can be nonzero, the constraint $(\mathbf{Dx})_{\widetilde{\mathcal{S}}^c} = \mathbf{0}$ thus reduces the uncertainty of the problem, making the optimization easy. We will introduce an efficient alternating direction method of multipliers (ADMM) based algorithm to handle subproblem (12) in Section IV.

D. Finding the Most Violated Constraint

In each iteration of Algorithm 1, we need to find the most active $\boldsymbol{\tau}$ within a large number of elements in Λ based on the updated $\boldsymbol{\alpha}^{t-1}$ and the corresponding $\boldsymbol{\beta}$.

At the optimum of problem (10), the following condition should hold for all $\boldsymbol{\tau}$'s:

$$\mathbf{A}^\mathsf{T}\boldsymbol{\alpha} = \mathbf{D}^\mathsf{T}\boldsymbol{\beta}, \quad \tau_i \|\boldsymbol{\beta}_i\|_2 \leq \lambda, \ \forall i \in [n]. \tag{13}$$

Given an $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, a subvector $\boldsymbol{\beta}_i$ with the larger $\|\boldsymbol{\beta}_i\|_2$ (and $\tau_i = 1$) violates the optimality condition (13) the more. In this sense, the most active $\boldsymbol{\tau}$ indicates *the largest number* of $\|\boldsymbol{\beta}_i\|_2$ with *the largest values*. Due to the constraint $\|\boldsymbol{\tau}\|_0 \leq \kappa$, with a given $\boldsymbol{\beta}$, we construct the most active $\boldsymbol{\tau}$ by finding the κ largest $\|\boldsymbol{\beta}_i\|_2$, and then setting the corresponding τ_i to 1 and the rest to 0.

```
Algorithm 2: Finding the Most Violated Constraint
Input: $\boldsymbol{\alpha}$, $\kappa$, $\mathbf{A}$ and regularizer parameter $r$.
Recover $\boldsymbol{\beta}$ from $\boldsymbol{\alpha}$ by solving problem (14);
Let $\boldsymbol{\beta}_i = \mathbf{C}_i\boldsymbol{\beta}$ and calculate $g_i = \|\boldsymbol{\beta}_i\|_2, \forall i \in [n]$;
Initialize $\boldsymbol{\tau} = \mathbf{0}$, and find the $\kappa$ largest $g_i$'s;
Set $\tau_i$ corresponding to the $\kappa$ largest $g_i$'s to 1;
Form $\mathcal{C}$, and return $\boldsymbol{\tau}$ and $\mathcal{C}$.
```

However, we cannot directly obtain the solution of the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, since the optimization of subproblem (11) is achieved by solving the problem in (12) w.r.t. the primal variable $\mathbf{x}$, as introduced in Section III-C. Thus, in each iteration, after solving the subproblem in primal (*i.e.* problem (12)), we firstly recover $\boldsymbol{\alpha}$ from the solution $\mathbf{x}$ via $\boldsymbol{\alpha} = \boldsymbol{\xi} = \mathbf{y} - \mathbf{A}\mathbf{x}$, and then reconstruct $\boldsymbol{\beta}$ based on the equality constraint $\mathbf{A}^\top\boldsymbol{\alpha} = \mathbf{D}^\top\boldsymbol{\beta}$ in (14). Recall that $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\boldsymbol{\beta} = [\boldsymbol{\beta}_v^\top, \boldsymbol{\beta}_h^\top]^\top \in \mathbb{R}^{2n}$. It is thus highly ill-posed to recover $\boldsymbol{\beta}$ from $\boldsymbol{\alpha}$ by solving the linear system $\mathbf{A}^\top\boldsymbol{\alpha} = \mathbf{D}^\top\boldsymbol{\beta}$. We use the standard ℓ_2-norm based regularizer $\|\boldsymbol{\beta}\|_2^2$ to alleviate the ill-posed nature. Relying on the ℓ_2-norm based regularizer, we can obtain a closed-form solution of $\boldsymbol{\beta}$ efficiently, which simplifies the optimization. We thus try to obtain $\boldsymbol{\beta}$ approximately by solving:

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\|\mathbf{D}^\top\boldsymbol{\beta} - \mathbf{A}^\top\boldsymbol{\alpha}\|_2^2 + \frac{r}{2}\|\boldsymbol{\beta}\|_2^2, \tag{14}$$

where $r > 0$ is a penalty parameter. Even though $\mathbf{D}$ is a concatenation of two parts, *i.e.* $\mathbf{D} = [\mathbf{D}_v^\top, \mathbf{D}_h^\top]^\top$, we can still efficiently solve (14) using Fast Fourier Transforms (FFTs) (see details in the supplementary material). Then we can easily obtain $\boldsymbol{\beta}_i, \forall i \in [n]$ based on $\boldsymbol{\beta}_i = \mathbf{C}_i\boldsymbol{\beta}$. For convenience, we define a vector $\mathbf{g}$, where $g_i = \|\boldsymbol{\beta}_i\|_2$.

In the t-th iteration of Algorithm 1, with the recovered $\boldsymbol{\beta}$, we can find the most active $\boldsymbol{\tau}_t$. As introduced in Section III-B, in practice, we recode the κ indices indicated by $\boldsymbol{\tau}_t$ into a set $\mathcal{C}_t$, *i.e.* $\mathcal{C}_t = \text{supp}(\boldsymbol{\tau}_t)$ and let $\mathcal{S}_t = \cup_{i=1}^t \mathcal{C}_i$ record the indices of the all activated $\boldsymbol{\tau}$'s. In general, once an element has been activated and added into $\mathcal{S}_{t-1}$, it is unlikely to be activated again and selected in $\mathcal{C}_t$ in the subsequent iterations. However, if the subproblem is not solved accurately, some of the activated elements may have large values of g_i (*i.e.* $\|\boldsymbol{\beta}_i\|_2$) and be activated again. To avoid overlapping components among $\mathcal{C}_t$ in practice, we form $\mathcal{C}_t$ from $[n] \setminus \mathcal{S}_{t-1}$. The algorithm for finding the most violated constraint is summarized in Algorithm 2.

E. Matching Pursuit for Minimizing TV Model

Based on Proposition 2 and Algorithm 2, we can implement our algorithm in primal form as summarized in Algorithm 3, which is referred as the *matching pursuit total variation (MPTV)* algorithm. In each iteration of Algorithm 3, we recover the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as well as find the most active $\boldsymbol{\tau}_t$ (relying on Algorithm 2), and then record all indices indicated by the activated $\boldsymbol{\tau}$'s into $\mathcal{S}_t$. As shown in Fig. 2 (h), the nonzero gradients are activated according to the $\mathbf{g}^t$ (*i.e.* $\|\boldsymbol{\beta}_i\|_2, \forall i \in [n]$), in which the elements with the largest values indicate the most significant gradients in the image. During the iterations, the image $\mathbf{x}$ is recovered based on the gradually activated gradients (indicated by $\mathcal{S}_t$). Since no $\boldsymbol{\tau}$ is involved at the initial stage, *e.g.* $\mathcal{S}_0 = \emptyset$, we initialize $\mathbf{x}$ via solving (12) given $\mathcal{S} = \emptyset$, *i.e.* letting $\mathbf{x}^0 = \mathbf{1} \otimes (\sum_{i=1}^n y_i/n)$.

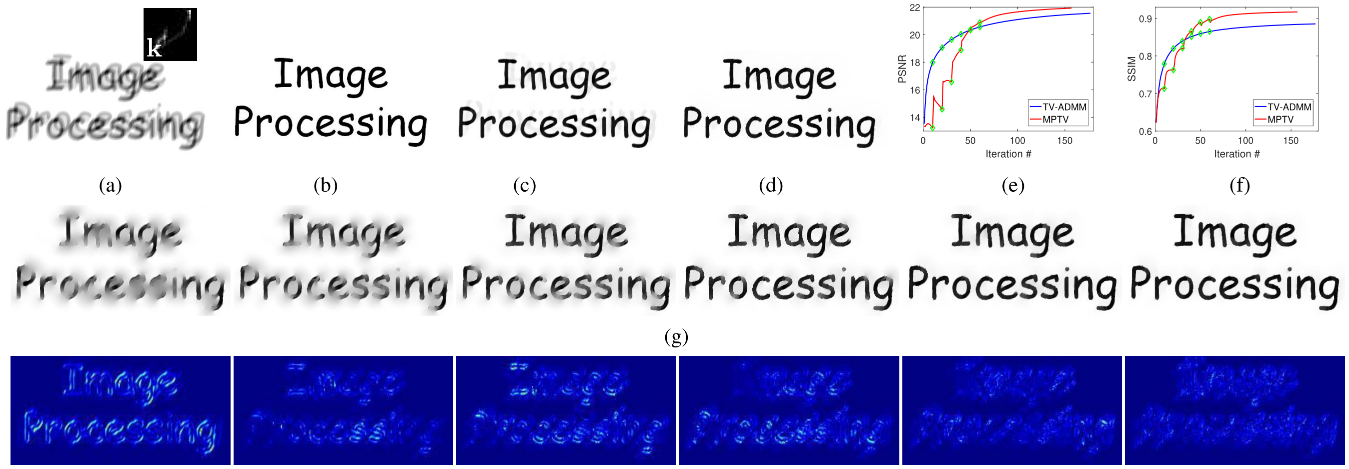
```
Algorithm 3: MPTV for Image Deconvolution
Input: Observation $\mathbf{y}$, parameter $\lambda$ and $\rho$.
Initialize $\mathbf{x}^0 = \mathbf{1} \otimes (\sum_{i=1}^n y_i/n)$, $\boldsymbol{\alpha}^0 = \mathbf{y} - \mathbf{A}\mathbf{x}^0$, $\mathcal{S}_0 = \emptyset$;
Set iteration index $t = 1$;
while Stopping conditions are not achieved do
    Finding the most violated constraint: Find the most active $\boldsymbol{\tau}_t$ based on $\boldsymbol{\alpha}^{t-1}$ via Algorithm 2, and record the corresponding indices into $\mathcal{C}_t$;
    Let $\mathcal{S}_t = \mathcal{S}_{t-1} \cup \mathcal{C}_t$;
    Subproblem optimization: Solve subproblem (12) via the ADMM in Algorithm 4, obtaining $\mathbf{x}^t$. Let $\boldsymbol{\alpha}^t = \mathbf{y} - \mathbf{A}\mathbf{x}^t$, and let $t = t + 1$;
```

Refinement for $\mathcal{S}$. To boost the robustness of the MPTV algorithm for real-world images in practice, we can take several additional simple steps in Algorithm 3 to refine the $\mathcal{S}$, which ensures we can make a conservative estimate of the image pixels. To conduct refinement in 2D image coordinates, we define a 2D map $\mathbf{M}$ with $\mathbf{M}(i, j) = 1$ for $(i, j) \in \mathcal{S}$ and 0 for others, where by a slight abuse of notation, we let $\mathcal{S}$ denote the locations of activated components in 2D coordinate system. Firstly, we perform a binary morphology dilation after an erosion operation, such that $\mathbf{M} = (\mathbf{M} \ominus \mathbf{R}_e) \oplus \mathbf{R}_d$, where $\ominus$ and $\oplus$ denote the binary erosion and dilation, respectively. The structuring elements for both $\mathbf{R}_e$ and $\mathbf{R}_d$ are disks with radius 3. This step helps to remove isolated active components in $\mathcal{S}$. Fewer artifacts thus arise from fragmented active components. Since the ideally sharp edges rarely appear in real-world images, this step does not remove the useful and significant structures while removing the isolated components. Secondly, similar to [49], to improve robustness for natural images and alleviate the visible boundaries between the active and inactive components, we blur the mask $\mathbf{M}$ slightly using a Gaussian filter with standard deviation 3 pixels. After refining the $\mathbf{M}$, we update $\mathcal{S}$ by recording the indices of the nonzero elements of $\mathbf{M}$ in $\mathcal{S}$. Note that the above operations are not necessary for images with ideally sparse gradients.

F. Setting of Parameter κ

The integer κ parameter controls the number of activated gradients in each iteration. In general, a large κ can decrease the iteration number of MPTV algorithm, and a very small κ helps to prevent activating too many gradient components. Thus, the value of κ may affect the quality of the recovered image. A sensitive study of κ is conducted in Section V-B.3. To balance between the efficiency and the performance, we here provide a strategy for setting κ.

Recall that κ reflects a rough knowledge of the support of $\mathbf{D}_v\mathbf{x}$ and $\mathbf{D}_h\mathbf{x}$ together, that is, $K = \text{card}(\text{supp}(\mathbf{D}_v\mathbf{x}) \cup \text{supp}(\mathbf{D}_h\mathbf{x}))$. As K is unknown, we first reconstruct $\boldsymbol{\beta}^0$ from the initialization $\boldsymbol{\alpha}^0$, then obtain $\mathbf{g}^0$ by letting $g_i^0 = \|\boldsymbol{\beta}_i^0\|_2$ and



(a) (b) (c) (d) (e) (f)

(g)

(h)

Fig. 2. An example for the MPTV method that is presented in Algorithm 3. (a) Input blurred image and blur kernel containing estimation error. (b) Ground truth image. (c) and (d) Results of TV-ADMM and the proposed MPTV, respectively. (e) and (f) show the convergence of TV-ADMM and MPTV. MPTV converges to a result with higher PSNR and SSIM values. (g) Intermediate results of MPTV, $\mathbf{x}^t$ in Algorithm 3, corresponding to the green dots in (e) and (f). (h) shows intermediate $\mathbf{g}^t$ in MPTV, corresponding to the $\mathbf{x}^t$ in (g). Note that MPTV is terminated after 6 iterations.

normalize $\mathbf{g}^0$ as $\mathbf{g}^0/\|\mathbf{g}^0\|_2$. Following the thresholding strategy in previous work [22], we set κ as the number of of elements in $\mathbf{g}^0$ larger than $\zeta\|\mathbf{g}^0\|_\infty$. In practice, $\zeta \geq 0.6$ works well for image deconvolution.

G. Solution Bias and Early Stopping

In general TV based image deconvolution, many existing methods are sensitive to the choice of the λ and face a dilemma – a too big a λ produces a solution with sparse gradients and helps to resist noise, but underfits the data and causes over-smoothness in the recovered image, while a small λ fits the observation well but reduces the regularization strength and may result in an estimate which suffers from ringing artifacts and/or noise. As a result it is often necessary to intensively search for an appropriate λ for a specific observation by considering the trade-off between the regularization strength and the solution bias. This is often impractical and inconvenient for users without specific knowledge. Due to the nature of our optimization scheme, we bypass this dilemma to a large degree, by proposing an early stopping condition below.

Recall that the MPTV algorithm incrementally activates the nonzero elements in $\mathbf{Dx}$, the value of $\Omega_{\text{TV}}(\mathbf{x})$ increases from 0 iteratively. By defining $\psi(\mathbf{x}) = \|\mathbf{y} - \mathbf{Ax}\|_2^2 + \lambda\Omega_{\text{TV}}(\mathbf{x})$, the algorithm will be stopped when

$$|\psi(\mathbf{x}^{t-1}) - \psi(\mathbf{x}^t)|/\psi(\mathbf{x}^0) \leq \epsilon,$$

where ϵ is a tolerance variable.

A too small ϵ may lead to too many iterations and activate too many $\boldsymbol{\tau}$'s due to the noise in $\mathbf{y}$, resulting in a non-sparse solution. We thus use an early stopping condition with a relatively large $\epsilon = 1 \times 10^{-3}$ (and maximum iteration number as 7) to prevent MPTV from too many iterations, which helps to obtain a solution $\mathbf{x}$ with small total variation, *i.e.* small $\Omega_{\text{TV}}(\mathbf{x})$, and sparse gradients. Since the early stopping strategy helps to ensure the sparsity of image gradients, we thus can use a relatively small λ to reduce the regularization bias in solution. As a result, MPTV performs well for a wide range of λ, which saves a lot of computational cost for hyper-parameter tuning (see Fig. 6). The sensitive study of ϵ can be found in Section V-B.3.

H. Discussions on Ringing Suppression by MPTV

As is outlined above, the deconvolution results often suffer from ringing artifacts in flat areas and near strong edges (Fig. 2 (c)), which may be caused by errors in the blur kernel estimate [20], the Gibbs phenomenon [47], etc, as discussed in Section II-C. The classical TV based model, unfortunately, is inadequate for avoid the undesired ringing artifacts.

For the deconvolution process given a fixed blur kernel $\mathbf{k}$ and corresponding convolution matrix $\mathbf{A}$, an image $\widetilde{\mathbf{x}}$ containing ringing artifacts often lies in a domain $\widetilde{\mathbb{X}} = \{\mathbf{x} \mid \mathbf{x} = \widehat{\mathbf{x}}+\mathbf{r}, \|\widehat{\mathbf{x}}-\mathbf{x}^*\|_2 \approx 0, \|\mathbf{Ar}\|_2 \approx 0\}$, where $\mathbf{x}^*$ denotes the ground truth image, and $\mathbf{r}$ is an error term for artifacts. An artifact term $\mathbf{r}$ corresponding to medium frequency ripples [49] lies close to $\mathbf{A}$'s nullspace. Since for any $\widetilde{\mathbf{x}} \in \widetilde{\mathbb{X}}$, there is $\|\mathbf{y} - \mathbf{A}\widetilde{\mathbf{x}}\|_2 \approx 0$, a solution only minimizing the ℓ_2-norm based data fidelity term is very likely in $\widetilde{\mathbb{X}}$. In the conventional model (5), the strength of the TV regularizer is only controlled by the parameter λ. Although a large weight of the TV regularizer can suppress the potential ringing artifacts, it also over smooths the significant edges and textures in the recovered image [4], [21]. When λ is not large enough, minimizing the TV based objective cannot produce images with sparse gradients and leads to results lying in $\widetilde{\mathbb{X}}$ with visible ringing. It is hard to find an accurate solution with less ringing artifacts only by tuning the parameter λ.

Instead of directly optimizing the objective in (2), MPTV gradually activates the most significant gradients according to the fitting errors as shown in Fig. 2. As shown in [47], the ringing artifacts are less significant in the fitting error, due to the smaller magnitudes. The possible artifacts can be naturally suppressed by the gradient activation step and the early stopping condition. Thus, benefiting from the optimization scheme, the proposed MPTV can suppress the ringing artifacts without the need of large λ. Recall that MPTV can also produce the desired sparse gradients with a small λ, which is consistent with that for ringing suppression. The example in Fig. 2 and the experimental results in Section V demonstrate that by selectively, and inhomogeneously, activating the appropriate gradients the intended effect of the regularizer can be maintained without causing the unwanted ringing.

I. Convergence Analysis

Similar to the analysis in [22] and [55], it can be proved that the θ's generated through progressive iterations of Algorithm 1 increase monotonically. As a result, the following proposition shows that MPTV converges to a global solution of (10).

Proposition 3: Let $\{(\boldsymbol{\alpha}_t, \theta_t)\}$ be the sequence generated by Algorithm 1. If the most violated constraint finding problem and subproblem (11) *can be solved, the sequence $\{(\boldsymbol{\alpha}_t, \theta_t)\}$ will converge to an optimal solution of problem* (10).

The proof can be found in the supplementary file.

In practice, we may use an early stopping to prevent MPTV from activating too many constraints, which will help to obtain images with sparse gradients.

IV. Efficient Optimization of Subproblem (12)

In MPTV algorithm (*i.e.* Algorithm 3), after updating $\mathcal{S}_t$, we need to solve the subproblem (12). To handle the equality constraints, we apply the alternating direction method of multipliers (ADMM) [52] which can be up to a few orders of magnitude faster than solving (11) directly [55]. Specifically, we iteratively update the primal and dual variables of the augmented Lagrangian function [35] of (12). Let subvectors $\boldsymbol{\gamma}_{\tilde{\mathcal{S}}}$ and $\boldsymbol{\gamma}_{\tilde{\mathcal{S}}^c}$ be the dual variables w.r.t. the two constraints, respectively. By introducing a dual variable $\boldsymbol{\gamma} \in \mathbb{R}^{2n}$ and a positive penalty parameter $\rho > 0$, we obtain the augmented Lagrangian for (12):

$$\begin{aligned}\mathcal{L}(\mathbf{x}, \mathbf{z}_{\tilde{\mathcal{S}}}, \boldsymbol{\gamma}) = {} & \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \textstyle\sum_{i\in\mathcal{S}} \|\mathbf{C}_{i\tilde{\mathcal{S}}}\mathbf{z}_{\tilde{\mathcal{S}}}\|_2 \\ & + \boldsymbol{\gamma}_{\tilde{\mathcal{S}}}^{\mathsf{T}}((\mathbf{D}\mathbf{x})_{\tilde{\mathcal{S}}} - \mathbf{z}_{\tilde{\mathcal{S}}}) + \boldsymbol{\gamma}_{\tilde{\mathcal{S}}^c}^{\mathsf{T}}(\mathbf{D}\mathbf{x})_{\tilde{\mathcal{S}}^c} \\ & + \frac{\rho}{2}\|(\mathbf{D}\mathbf{x})_{\tilde{\mathcal{S}}} - \mathbf{z}_{\tilde{\mathcal{S}}}\|_2^2 + \frac{\rho}{2}\|(\mathbf{D}\mathbf{x})_{\tilde{\mathcal{S}}^c}\|_2^2. \qquad (15)\end{aligned}$$

Then, we can solve for $\mathbf{x}$ by iteratively minimizing $\mathcal{L}(\cdot)$ w.r.t. $\mathbf{x}$ and $\mathbf{z}_{\tilde{\mathcal{S}}}$, and updating $\boldsymbol{\gamma}$. Let k denote the iteration number. Applying ADMM, we carry out the following steps at each iteration:

Step 1: Compute $\mathbf{z}_{\tilde{\mathcal{S}}}^{k+1}$ with fixed $\mathbf{x}^k$ and $\boldsymbol{\gamma}^k$ by solving:

$$\min_{\mathbf{z}_{\tilde{\mathcal{S}}}} \lambda \sum_{i\in\mathcal{S}} \|\mathbf{C}_{i\tilde{\mathcal{S}}}\mathbf{z}_{\tilde{\mathcal{S}}}\|_2 + \frac{\rho}{2}\|(\mathbf{D}\mathbf{x}^k)_{\tilde{\mathcal{S}}} - \mathbf{z}_{\tilde{\mathcal{S}}} + \frac{1}{\rho}\boldsymbol{\gamma}_{\tilde{\mathcal{S}}}^k\|_2^2, \qquad (16)$$

which has an unique closed-form solution. For $\forall i \in \mathcal{S}$, let $\boldsymbol{\mu}_i = \mathbf{C}_{i\tilde{\mathcal{S}}}(\mathbf{D}\mathbf{x}^k + \frac{1}{\rho}\boldsymbol{\gamma}^k)$. We can obtain the solution of (16) by a two-dimensional shrinkage [4] for $\forall i \in \mathcal{S}$:

$$\mathbf{C}_{i\tilde{\mathcal{S}}}\mathbf{z}_{\tilde{\mathcal{S}}}^{k+1} = \max\left\{\|\boldsymbol{\mu}_i\|_2 - \frac{\lambda}{\rho}, 0\right\} \frac{\boldsymbol{\mu}_i}{\|\boldsymbol{\mu}_i\|_2}, \quad \forall i \in \mathcal{S}, \qquad (17)$$

where a convention $0 \cdot (0/0) = 0$ is followed. Following this, for convenience, we define a temporary variable $\mathbf{z}' = [\mathbf{z}'^{\mathsf{T}}_v, \mathbf{z}'^{\mathsf{T}}_h]^{\mathsf{T}}$, and let $\mathbf{z}'_{\tilde{\mathcal{S}}} = \mathbf{z}_{\tilde{\mathcal{S}}}^{k+1}$ and $\mathbf{z}'_{\tilde{\mathcal{S}}^c} = \mathbf{0}$.

Algorithm 4: ADMM for Solving Subproblem (12)

```
Input: Observation $\mathbf{y}$, parameter $\lambda$ and $\rho$, initialization of
       image $\mathbf{x}^0$, index set $\mathcal{S}_t$.
Initialize $\boldsymbol{\beta}^0 = \mathbf{0}$. Set iteration number as $k = 0$;
while Stopping conditions are not achieved do
    Compute $\mathbf{z}_{\tilde{\mathcal{S}}}^{k+1}$ according to (17);
    Generate $\mathbf{z}'$ by letting $\mathbf{z}'_{\tilde{\mathcal{S}}} = \mathbf{z}_{\tilde{\mathcal{S}}}^{k+1}$ and $\mathbf{z}'_{\tilde{\mathcal{S}}^c} = \mathbf{0}$;
    Compute $\mathbf{x}^{k+1}$ by solving problem (18);
    Update $\boldsymbol{\gamma}^{k+1}$ according to (20);
    If the stopping condition is achieved, stop;
    Let $k = k + 1$;
```

Step 2: Update $\mathbf{x}^{k+1}$ by solving $\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{z}_{\tilde{\mathcal{S}}}^{k+1}, \boldsymbol{\gamma}^k)$ w.r.t. $\mathbf{x}$:

$$\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \frac{\rho}{2}\|\mathbf{D}\mathbf{x} - \mathbf{z}' + \frac{1}{\rho}\boldsymbol{\gamma}^k\|_2^2, \qquad (18)$$

which is quadratic in $\mathbf{x}$. Let $\boldsymbol{\nu} = [\boldsymbol{\nu}_v^{\mathsf{T}}, \boldsymbol{\nu}_h^{\mathsf{T}}]^{\mathsf{T}}$ where $\boldsymbol{\nu}_v = \mathbf{z}'_v - \frac{1}{\rho}\boldsymbol{\gamma}_v^k$ and $\boldsymbol{\nu}_h = \mathbf{z}'_h - \frac{1}{\rho}\boldsymbol{\gamma}_h^k$. The minimizer can be recovered from the normal equation:

$$\left[\mathbf{A}^{\mathsf{T}}\mathbf{A} + \rho(\mathbf{D}^{\mathsf{T}}\mathbf{D})\right]\mathbf{x} = \mathbf{A}^{\mathsf{T}}\mathbf{y} + \rho\mathbf{D}^{\mathsf{T}}\boldsymbol{\nu}. \qquad (19)$$

Under the periodic boundary condition, we can obtain the solution via FFTs [4]:

$$\mathbf{x}^{k+1} = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(\mathbf{A})}\mathcal{F}(\mathbf{y}) + \rho(\overline{\mathcal{F}(\mathbf{D}_v)}\mathcal{F}(\boldsymbol{\nu}_v) + \overline{\mathcal{F}(\mathbf{D}_h)}\mathcal{F}(\boldsymbol{\nu}_h))}{\overline{\mathcal{F}(\mathbf{A})}\mathcal{F}(\mathbf{A}) + \rho(\overline{\mathcal{F}(\mathbf{D}_v)}\mathcal{F}(\mathbf{D}_v) + \overline{\mathcal{F}(\mathbf{D}_h)}\mathcal{F}(\mathbf{D}_h))}\right),$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ denote the Fourier transform and the inverse transform, respectively, $\overline{\mathcal{F}(\cdot)}$ denotes the complex conjugate of $\mathcal{F}(\cdot)$, and the multiplication and division are all component-wise operators.

Step 3: Update the dual variable $\boldsymbol{\gamma}$:

$$\boldsymbol{\gamma}^{k+1} = \boldsymbol{\gamma}^k + \rho(\mathbf{D}\mathbf{x}^{k+1} - \mathbf{z}'). \qquad (20)$$

The ADMM method for solving problem (12) is summarized in Algorithm 4. As shown in Proposition 2, the dual variable can be recovered by $\boldsymbol{\alpha}^* = \boldsymbol{\xi}^* = \mathbf{y} - \mathbf{A}\mathbf{x}^*$.

Stopping conditions of Algorithm 4. For Algorithm 4 applied in solving subproblem (12), by defining $\varphi(\mathbf{x}) = \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2$, the algorithm is stopped when

$$|\varphi(\mathbf{x}^{t-1}) - \varphi(\mathbf{x}^t)|/\varphi(\mathbf{x}^0) \leq \epsilon_{\text{in}},$$

where ϵ_{in} is a tolerance value. This stopping condition is adapted from [9]. In practice, a sufficiently small ϵ_{in} is enough to find the most-violated constraint. We suggest setting $\epsilon_{\text{in}} = 0.001$ and the maximum iteration to 100. To avoid the possible early stopping issue in ADMM at the very beginning iterations, we can let the algorithm be stopped only after a minimal number of iterations.

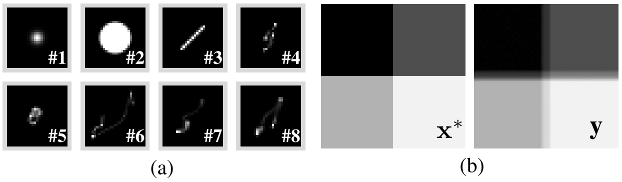


Fig. 3. (a) Blur kernels (PSFs) for generating synthetic data. #1: 25 × 25 Gaussian blur kernel with standard derivation 1.6; #2: 15×15 disk blur kernel; #3: 11 × 11 linear motion blur with length 15 and angle 45°; #4-#8: Motion blur kernel from Levin *et al.* 's dataset [56] (squared kernels with side length 19, 15, 27, 21 and 23). All blur kernels are padded as same size for better illustration. (b) An example of the synthetic image with sparse gradients, where kernel #6 is used.

V. Experiments

In this section, we evaluate the performance of our method against the state-of-the-art image deconvolution methods on both synthetic data and real-world images. To make a comprehensive study of the proposed method, we consider different types of blur kernels (PSFs) and images. We will also study the sensitivity of the MPTV algorithm to noises, blur level and different parameter settings, and demonstrate its ability to suppress ringing artifacts. All experiments are conducted using MATLAB on a desktop computer with an Intel Core i5 CPU with 8GB of RAM. We use brute-force search for the parameters to obtain the best performance for every method.

A. Experimental Settings

1) Synthetic Dataset Generation: Three synthetic datasets are generated based on sharp images $\mathbf{x}^*$ belonging to three different types – images with sparse gradients, text images with near sparse gradients and natural images. We use a set of different blur kernels to construct different $\mathbf{A}$'s. For each $\mathbf{A}$ we generate the blurry testing images $\mathbf{y}$'s by the imaging model (1) $\mathbf{y} = \mathbf{A}\mathbf{x}^* + \mathbf{n}$, where $\mathbf{n}$ denotes the additive Gaussian white noise with noise level[1] 0.3% unless stated otherwise. All synthetic blurry images in these three datasets are generated using a set of *different blur kernels (PSFs)*, including the Gaussian blur kernel, disk blur kernel, linear motion blur kernel and motion blur kernels with different sizes from Levin *et al.* [56] dataset as shown in Fig. 3 (a). A nonperiodic boundary condition [57] is used for imitating the real blurred image. We will discuss more details of the datasets in the following.

2) Evaluation Metrics: Since we have the ground truth for synthetic datasets, we use the peak-signal-noise-ratio (PSNR) and structural similarity index (SSIM) [58] as the metric.

3) Comparison With Other Algorithms: We compare the proposed MPTV method with several state-of-the-art competitors for image deconvolution, including the methods by promoting sparsity on image gradients (such as *e.g.* FTVd [4], L0-Abs [59], WDTV [11], IRLS [36], hyper-Laplacian prior based method [7] and L0TV [12]), and other methods (*e.g.* BM3D [37] and a kernel similarity based method [38]). Moreover, we also compare with a TV-ADMM method by minimizing the isotropic TV model using ADMM method, which is a direct baseline of the proposed method.

B. Experiments on Synthetic Image With Sparse Gradients

Recall that the proposed MPTV is based on the identification of non-zero subsets of the sparse gradient. We thus first evaluate the performance of the proposed method on an image with sparse gradients.

Fig. 3 (b) shows a 256 × 256 sharp image with sparse gradients for synthesizing blurred testing images. We perform image deconvolution using different methods and record the averaged values of PSNR and SSIM with 8 different kernels in Table I. From Table I, the proposed MPTV performs the best in terms of PSNR and SSIM. Here, the execution time of MPTV for handling a 256 × 256 image is within 1 second.

1) Sensitivity Study on Noise: Based on the synthetic data above, we conduct a noise level sensitivity study for the proposed MPTV. We first generate two blurred images, one with the Gaussian blur kernel (#1 in Fig. 3 (a)) and the other with a motion blur kernel (#6 in Fig. 3 (a)), and then add Gaussian noise with noise with levels varying from 1% to 10% with interval 1% to the two blurred images. Fig. 4 records the PSNR and SSIM values of different methods, which show that the performance of many previous methods decreases as noise level increases. The performance of MPTV is high and stable since the cutting-plane optimization method restricts the elements of $\mathbf{D}\mathbf{x}$ corresponding to the inactive $\boldsymbol{\tau}$ to be zero.

2) Sensitivity Study on Blur Level: The deconvolution task becomes more difficult as the level of blur increases. In order to test the robustness of the various methods to increasing blur level, we generate synthetic images using both the Gaussian and linear motion blur kernels, as they exhibit a natural parametrization for this purpose. We thus generate 8 squared Gaussian blur kernels where the side lengths range from 15 to 85 pixels with an interval of 10, and standard derivations 1.6, 3, 5 and 7, and 8 linear motion blur kernels with 45° angle and lengths from 7 to 63 with an interval of 8. Fig. 5 shows that the performance of all methods decreases with increasing blur level for both Gaussian and motion blur. Nevertheless, the proposed MPTV performs the best for all blur levels.

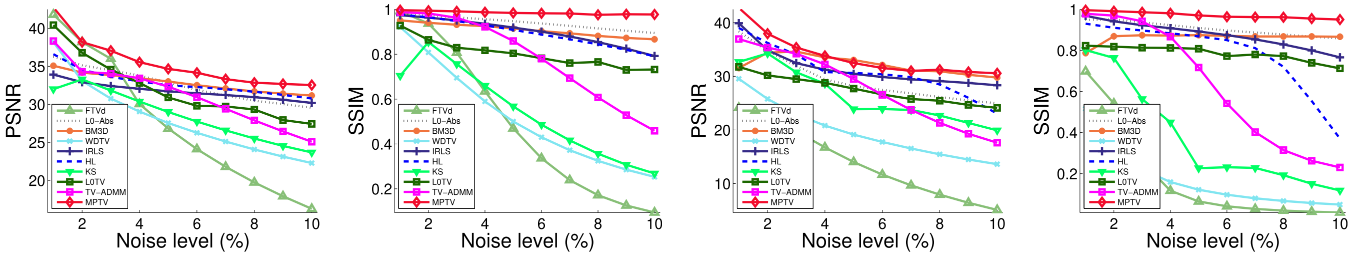
3) Sensitivity Study on Parameters: Firstly, we conduct a sensitivity study for the parameter λ. MPTV with a proper λ achieves solutions with high accuracy and less regularizer bias. Due to the cutting-plane method and the early stopping strategy, MPTV is not sensitive to the value of λ. In this experiment, we primarily compare MPTV with the TV-ADMM method that shares a similar problem formulation and the same implementation details apart from the binary indicator $\boldsymbol{\tau}$ and the cutting-plane based optimization scheme. We perform deconvolution using the two methods given 20 λ's with values from 1×10^{-5} to 0.96×10^{-3} with an interval of 5×10^{-5}, and record the average PSNR and SSIM values of all testing images in Fig. 6. According to evaluation based on both PSNR and SSIM, the proposed MPTV method is more robust to the setting of λ and outperforms the TV-ADMM for all λ's, which

[1] An image $\mathbf{y}$ with $(100 \times \sigma)$% Gaussian noise is generated by adding noise with zero mean and standard derivation σ for image $\mathbf{A}\mathbf{x}^*$ with [0, 1] intensity range.

TABLE I

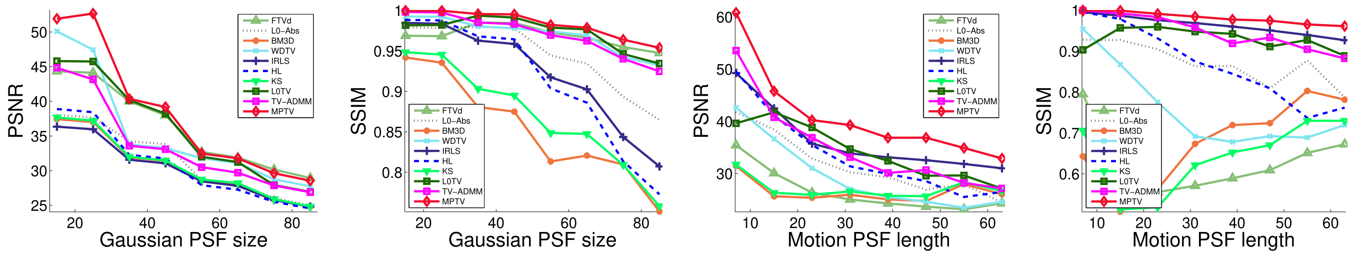
Comparison on the Synthetic Image With Sparse Gradients (PSNR/SSIM)

PSF index	1	2	3	4	5	6	7	8	Avg.
Input	31.37/0.9622	27.05/0.9178	27.41/0.9236	28.22/0.9324	29.03/0.9466	22.92/0.8610	22.69/0.8911	23.81/0.8881	26.56/0.9154
FTVd [4]	47.36/0.9820	27.81/0.4727	27.54/0.4416	34.34/0.7328	32.99/0.6801	27.68/0.4767	34.32/0.7375	31.07/0.5823	32.89/0.6382
L0-Abs [60]	37.69/0.9670	33.83/0.9221	35.99/0.8572	39.22/0.8641	39.37/0.8673	33.52/0.7786	38.32/0.8595	37.76/0.8522	36.96/0.8710
BM3D [38]	37.00/0.9355	25.68/0.4194	24.61/0.3763	40.62/0.9190	46.88/0.9814	23.14/0.3071	40.06/0.9435	35.57/0.8428	34.20/0.7156
WDTV [11]	47.44/0.9876	35.17/0.8668	34.42/0.7964	39.54/0.8953	40.45/0.9159	32.13/0.7047	40.29/0.9141	36.42/0.8256	38.23/0.8633
IRLS [37]	35.97/0.9826	33.96/0.9717	42.48/0.9831	49.24/0.9905	48.52/0.9929	48.82/0.9885	50.68/0.9940	45.66/0.9813	44.42/0.9856
HL [7]	36.76/0.9874	33.72/0.9572	39.90/0.9764	49.23/0.9921	48.97/0.9943	40.54/0.9214	51.27/0.9935	44.88/0.9805	43.16/0.9754
KS [39]	37.21/0.9455	25.91/0.3960	25.06/0.3593	39.98/0.9229	46.80/0.9832	23.60/0.3020	40.14/0.9487	35.87/0.8555	34.32/0.7141
L0TV [12]	46.69/0.9821	41.09/0.9890	41.76/0.9568	30.95/0.6408	30.84/0.7854	31.88/0.8409	37.43/0.8709	21.87/0.4777	35.31/0.8180
TV-ADMM	44.14/0.9703	37.24/0.9693	40.78/0.9726	51.48/0.9790	52.93/0.9793	44.05/0.9567	53.59/0.9795	43.60/0.9707	45.98/0.9722
MPTV	**50.41/0.9997**	**41.09/0.9973**	**47.56/0.9979**	**56.98/0.9996**	**57.13/0.9997**	**51.80/0.9955**	**59.85/0.9998**	**50.51/0.9965**	**51.92/0.9983**



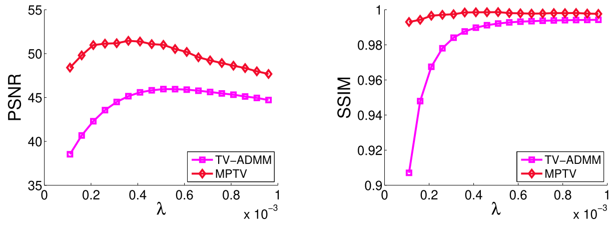
(a) (b) (c) (d)

Fig. 4. Study of noise sensitivity. PSNR and SSIM of the deconvolution results are evaluated for the blurry images contaminated by Gaussian noise with increasing noise level. (a) and (b) Results on the blurry images with Gaussian blur. (c) and (d) Results on the blurry images with motion blur.



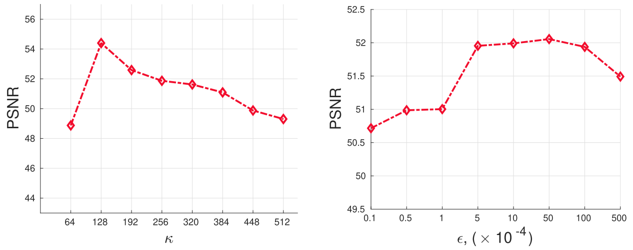
(a) (b) (c) (d)

Fig. 5. Blur level sensitivity study of various methods. PSNR and SSIM of the deconvolution results for 8 blurry images contaminated by Gaussian blur or linear motion blur with increasing blur level. (a) and (b) Results on the images blurred by the Gaussian PSFs. (c) and (d) Results on the images blurred by the linear motion PSFs.



(a) (b)

Fig. 6. Sensitivity study of parameter λ. The deconvolution results of TV-ADMM and MPTV with varying λ are evaluated and compared. (a) Evaluation based on PSNR. (b) Evaluation based on SSIM.



(a) (b)

Fig. 7. Sensitivity studies of κ and ϵ for MPTV. PSNR of the deconvolution results with different values for the parameters are shown. (a) Sensitivity of κ. (b) Sensitivity of the stopping condition parameter ϵ. *In (b), the x axis is scaled for visualization.*

proves the effectiveness of the proposed MPTV formulation and the cutting-plane based optimization method.

In the second experiment, we study the sensitivity of the parameter κ for MPTV. The PSNR values of the image results and the corresponding κ are reported in Fig. 7 (a). The same stopping condition described in Section III-G is used for different κ's (from 64 to 512). From Fig. 7 (a), a proper κ helps to prevent MPTV from activating too many or too less nonzero gradients, resulting in high-quality results. If κ is too small, MPTV may not activate enough nonzero

TABLE II

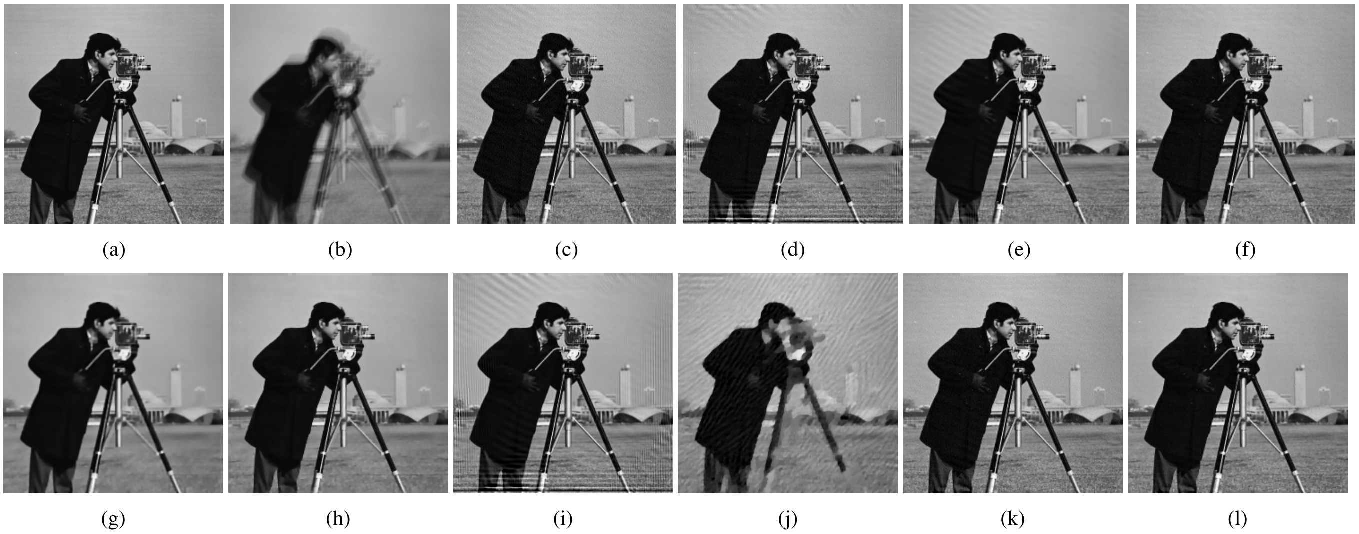
QUALITATIVE COMPARISON ON TEXT IMAGES WITH NEAR SPARSE GRADIENTS (PSNR/SSIM)

PSF index	1	2	3	4	5	6	7	8	Avg.
Input	20.04/0.8009	15.97/0.5684	16.88/0.6312	17.51/0.6632	17.54/0.6945	13.49/0.4328	13.86/0.4882	14.63/0.4938	16.24/0.5966
FTVd [4]	28.91/0.9573	27.03/0.9347	32.58/0.9398	37.00/0.9423	35.81/0.9490	31.88/0.8716	36.95/0.9492	34.18/0.9337	33.04/0.9347
BM3D [38]	27.14/0.9517	25.52/0.8942	28.76/0.8926	37.93/0.9866	37.01/**0.9891**	28.86/0.8562	38.49/0.9891	33.18/0.9499	32.11/0.9387
WDTV [11]	27.38/0.9459	25.40/0.9103	32.06/0.9302	37.25/0.9427	35.95/0.9501	31.96/0.8720	37.25/0.9514	34.23/0.9339	32.68/0.9296
IRLS [37]	25.78/0.9384	22.96/0.8808	30.16/0.9569	38.49/0.9820	35.63/0.9833	37.22/0.9790	38.11/0.9859	35.00/0.9802	32.92/0.9608
HL [7]	25.60/0.9342	21.03/0.8210	27.63/0.9418	35.06/0.9825	33.65/0.9818	32.60/0.9667	36.26/0.9871	32.72/0.9751	30.57/0.9488
KS [39]	27.52/0.9422	25.70/0.8948	29.28/0.8912	37.95/0.9738	37.12/0.9804	29.27/0.8535	38.67/0.9879	33.55/0.9502	32.38/0.9343
L0TV [12]	29.24/0.9502	25.62/0.9018	31.99/0.9152	13.91/0.5324	14.38/0.5769	14.94/0.5111	14.09/0.5202	11.62/0.4153	19.47/0.6654
TV-ADMM	29.98/0.9673	26.91/0.9396	33.64/0.9635	38.98/0.9738	37.46/0.9760	37.10/0.9642	39.13/0.9775	36.68/0.9718	34.99/0.9667
MPTV	**30.15/0.9681**	**27.82/0.9434**	**34.27/0.9719**	**39.94/0.9873**	**38.04**/0.9830	**38.05/0.9815**	**39.96/0.9893**	**37.30/0.9823**	**35.69/0.9759**

TABLE III

QUALITATIVE COMPARISON ON NATURAL IMAGES (PSNR/SSIM)

PSF index	1	2	3	4	5	6	7	8	Avg.
Input	25.75/0.7437	21.53/0.4991	22.42/0.5822	23.18/0.6131	23.22/0.6203	18.66/0.3782	19.17/0.4138	20.05/0.4371	21.75/0.5359
FTVd [4]	29.74/0.8459	23.93/0.5673	23.71/0.5770	31.07/0.8197	29.67/0.7663	25.00/0.6734	31.53/0.8321	27.25/0.7236	27.74/0.7257
L0-Abs [60]	30.95/**0.8911**	19.36/0.5190	19.69/0.5368	26.52/0.7709	29.53/0.8588	22.58/0.6414	31.74/0.8913	22.69/0.6637	25.38/0.7216
BM3D [38]	30.76/0.8799	24.79/0.6477	24.44/0.6576	33.07/0.8916	34.50/0.9281	21.77/0.5983	35.33/0.9392	29.55/0.8434	29.28/0.7982
WDTV [11]	30.60/0.8818	27.62/0.7784	29.37/0.8323	34.86/0.9228	34.57/0.9216	29.87/0.8496	35.68/0.9359	30.90/0.8742	31.69/0.8746
IRLS [37]	29.02/0.8423	25.27/0.6933	27.96/0.8019	30.35/0.8609	30.25/0.8582	29.42/0.8394	31.97/0.8938	30.09/0.8576	29.29/0.8309
HL [7]	29.59/0.8540	25.49/0.6980	28.56/0.8164	31.89/0.8889	31.85/0.8836	30.10/0.8614	33.59/0.9161	31.04/0.8773	30.26/0.8495
KS [39]	30.48/0.8780	11.67/0.1524	11.85/0.1807	19.79/0.4958	22.92/0.7444	15.53/0.3038	26.98/0.8139	15.09/0.3183	19.29/0.4859
L0TV [12]	29.28/0.8615	26.17/0.7345	28.37/0.8189	21.52/0.4082	21.97/0.4506	22.73/0.5399	23.59/0.5455	18.80/0.2801	24.05/0.5799
TV-ADMM	30.93/0.8881	27.42/0.7753	28.25/0.8119	35.06/0.9263	34.67/0.9230	31.70/0.8869	35.74/0.9363	30.76/0.8744	31.82/0.8778
MPTV	**30.96**/0.8869	**27.89/0.7918**	**30.08/0.8627**	**35.53/0.9394**	**35.02/0.9330**	**32.85/0.9188**	**36.32/0.9489**	**32.67/0.9161**	**32.67/0.8997**


(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

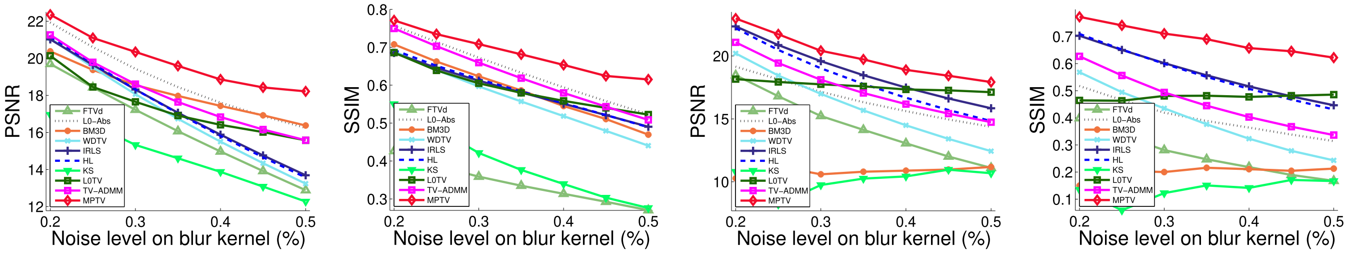
Fig. 8. An example of the results on synthetically blurred natural images. The blurred image $\mathbf{y}$ is generated using the sharp image `cameraman` and the #6 blur kernel in Fig. 3 (a). (a) Ground truth sharp image $\mathbf{x}^*$. (b) Input blurred image $\mathbf{y}$. (c)-(l) Deconvolution results of different methods in comparison.

elements in a limited number of iterations. When κ increases, the performance slightly degrades.

In the third experiment, we study the influence of the parameter ϵ on the performance of MPTV. Different ϵ values result in different iteration numbers. Fig. 7 (b) records the PSNR values with 8 different ϵ's from 1×10^{-5} to 5×10^{-2}. As shown in Fig. 7 (b), MPTV works robust to a wide range of ϵ. When ϵ is large enough, *e.g.* $\epsilon = 1 \times 10^{-3}$ in Fig 7 (b), the early stopping helps MPTV to produce images with sparse gradients and satisfactory quality. A too small ϵ leads to too many iterations and thus activates too many nonzero gradients, which slightly degrades the performance.

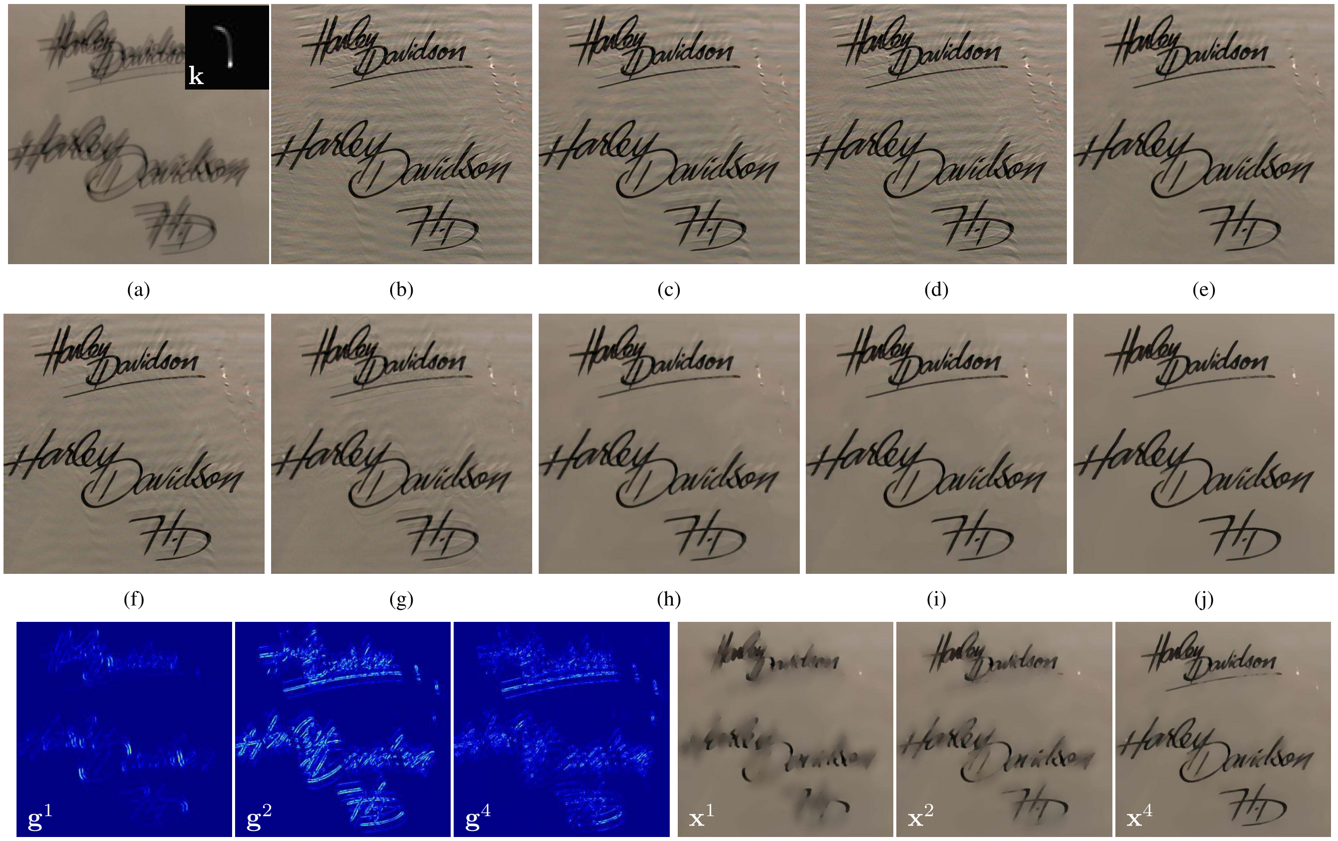
C. Quantitative Studies on Text Images

A text image often contains near sparse gradients and represents an important class of real images. We thus study the performance of MPTV on text images using a dataset containing 14 ground truth text images and 8 kernels (in Fig. 3 (a)).



(a) (b) (c) (d)

Fig. 9. Study of sensitivity to the error in the blur kernel estimate. Blur kernels with increasing noise level are given for deconvolution. PSNR and SSIM of the deconvolution results are evaluated for 7 blurry images contaminated by Gaussian blur or motion blur. (a) and (b) Results on the images blurred by Gaussian blur kernel. (c) and (d) Results on the images blurred by the motion blur kernel.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l)

Fig. 10. Experimental results on real text image. (a) Input blurred image $\mathbf{y}$ and blur kernel $\mathbf{k}$ estimated using the method in [60]. (b)-(j) Deconvolution results of different methods. The results using TV_2 in (h) are estimated using the method in [60]. We show the intermediate results during the iterations of the proposed MPTV in (k) and (l). (k) $\mathbf{g}^t$ for activating gradients. (l) Intermediate $\mathbf{x}^t$. In this example, MPTV terminates after 6 iterations.

Here, we perform deconvolution with ground-truth blur kernels. For each blur kernel, we show the averaged PSNR and SSIM of both input blurred images and the deblurring results in Table II.[2] From the table, MPTV consistently performs better than state-of-the-art methods. In particular, the results of MPTV appear sharper than others.

D. Experiments on Natural Images

In this section, we evaluate the proposed MPTV algorithm and other methods using natural images. We construct a dataset including 192 blurred natural images from 24 sharp images and the 8 kernels shown in Fig. 3 (a). Since the gradients of natural images are much denser, MPTV is required to activate many gradients to fit the observation, leading to lower superiority over the comparator methods, as shown in Table III. Nevertheless, the proposed MPTV performs better than others in terms of PSNR and SSIM values. Fig. 8 provides a visual comparison which demonstrates that the latent images estimated by MPTV exhibit sharper edges and less ringing artifacts than those of its comparators. We also note that MPTV failed to model some of the subtle textures in the background due to the binary indicator for the activated gradients.

[2] The comparison with L0-ABS [59] is absent since the corresponding code only works on square images.

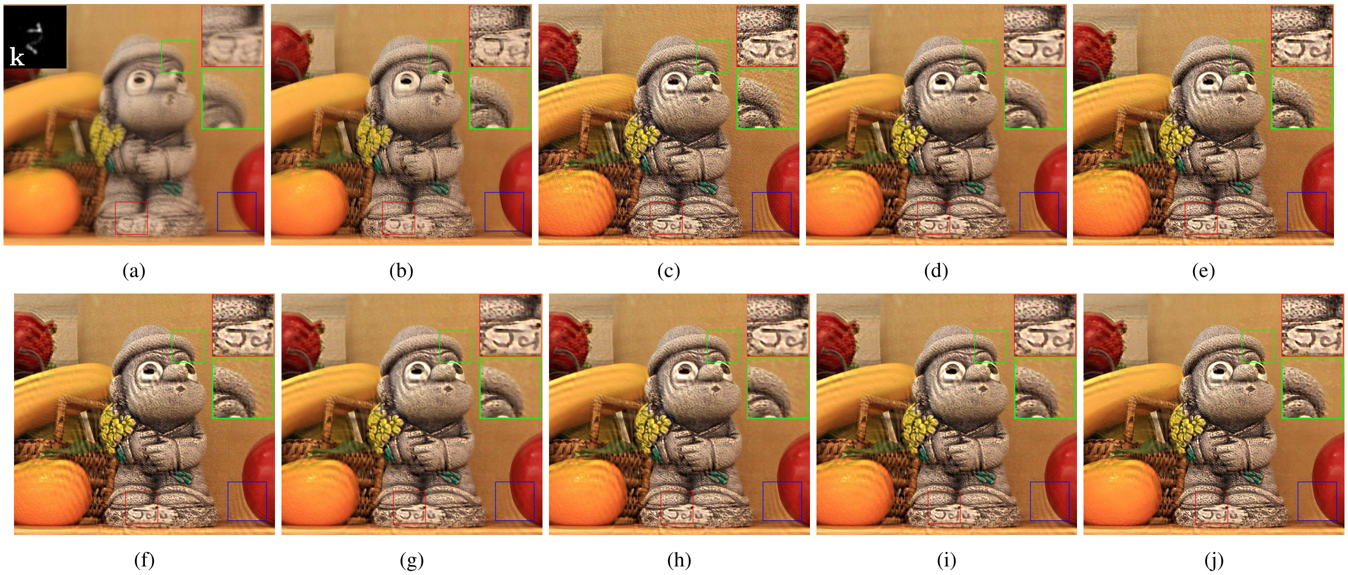


Fig. 11. Experimental results on a real-world image. (a) Input blurred image **y** and blur kernel **k**. (b) Deblurring result of the software in [29] is used as a baseline. (c)-(j) Deconvolution results of different methods. The proposed MPTV recovers more sharp details (see red box) and introduces fewer ringing artifacts (see blue box).

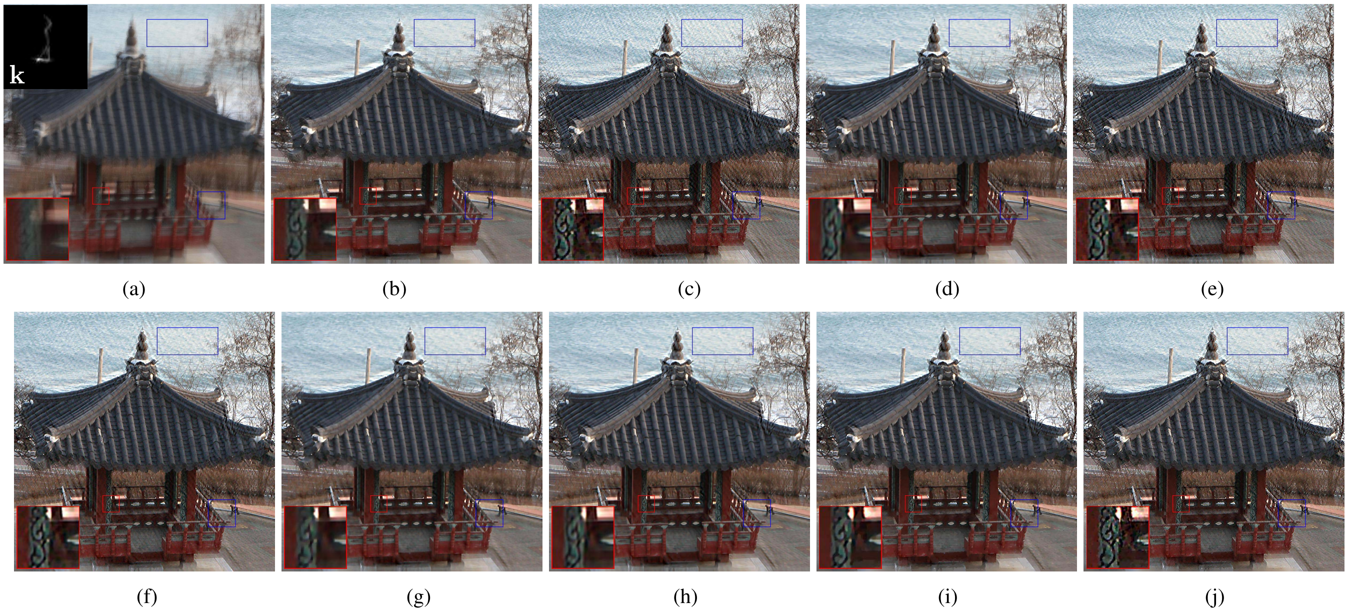


Fig. 12. Experimental results on a real-world image. (a) Input blurred image **y** and blur kernel **k**. (b) Deblurring result of the software in [29] is used as a baseline. (c)-(j) Deconvolution results of different methods. The results of MPTV with different values for parameter λ are shown in (i) and (j). The results of MPTV suffer from fewer ringing artifacts (see blue boxes).

1) Sensitivity Study on Blur Kernel Errors: In practice, the blur kernel for image deconvolution is usually an estimate containing errors, which lead to ringing artifacts [20] and incur degraded deconvolution performance. We thus here describe an experiment to study the sensitivity of different methods to errors in the kernel estimate. Seven 256 × 256 sharp natural images (among the 24 sharp images above) and two blur kernels (Gaussian blur kernel #1 and motion blur kernel #6) are used to generate the test data. The ground truth known blur kernel is applied to each image, and each kernel then has noise added to form an erroneous blur kernel estimate which is passed to the various deblurring methods. The added noise is sampled from Gaussian distribution with a noise level increasing from 0.2% to 0.5% (with an interval of 0.05%).

As shown in Fig. 9, the proposed method is more robust than other algorithms since the cutting-plane of the proposed MPTV estimates the sharp image by gradually activating the significant gradients which helps to suppress the ringing artifacts.

E. Experiments on Real-World Images

We evaluate the performance of MPTV on two kinds of real-world blurred images, i.e. blurred text images and natural images. Here, the blur kernels are unknown. For real-world image deconvolution, the kernel estimation error is one main reason causing ringing artifacts [20]. For fair comparison, we use blur kernels estimated by the method in [60] for all compared deconvolution methods.

We first report the results on a blurred text image in Fig. 10. From the figure, the deblurred results of most compared methods contain significant ringing artifacts and/or over-smooth strokes. In fact, since these methods usually use a universal criterion to suppress undesired components, such as thresholding on gradients, they may cause over-smoothness (see Fig. 10 (h)), or fail to suppress artifacts (see Fig. 10 (b)). On the contrary, the proposed MPTV algorithm can recover sharper and clearer results by gradually activating significant strokes to fit the blurred observation (Fig. 10 (k) and (l)).

In Fig. 11, we show the deblurring results on a natural image containing both subtle textures and flat background. Although the ringing artifacts caused by the kernel estimation error are unavoidable, MPTV performs favorably against the state-of-the-art competitors by alleviating ringing artifacts and preserving shape details simultaneously. Fig. 12 shows that, on an image with complex contents, the proposed MPTV can achieve favorable results than other methods. With a smaller regularization weight ($\lambda = 0.00005$, Fig. 12 (j)), MPTV recovers much sharper and clearer details but still suppresses the ringing artifacts to quite a low level.

VI. Conclusion

We have proposed a matching pursuit based total variation minimization method for image deconvolution. By introducing a binary vector to indicate the significant components in the image gradients, we formulate the TV minimization problem as a QCLP problem and solve it via a cutting-plane method. The proposed algorithm with early stopping helps to reduce the solution bias, to alleviate over-smoothness, and suppress the ringing artifacts in the deconvolution results. Comprehensive empirical studies on many different datasets show the superior performance of the proposed MPTV over the state-of-the-art methods.

In the future, we plan to apply the proposed method to a variety of related image processing tasks, such image super-resolution. We are also interested to extend the matching pursuit framework to the wavelet frame based on the models introduced in [61].

References

[1] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.

[2] J.-F. Cai, B. Dong, S. Osher, and Z. Shen, "Image restoration: Total variation, wavelet frames, and beyond," *J. Amer. Math. Soc.*, vol. 25, no. 4, pp. 1033–1089, 2012.

[3] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. ICCV*, Nov. 2011, pp. 479–486.

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

[5] D. Gong, M. Tan, Y. Zhang, A. van den Hengel, and Q. Shi, "Blind image deconvolution by automatic gradient activation," in *Proc. CVPR*, Jun. 2016, pp. 1827–1836.

[6] W. H. Richardson, "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Amer.*, vol. 62, no. 1, pp. 55–59, Jan. 1972.

[7] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-Laplacian priors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1033–1041.

[8] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1838–1857, Jul. 2011.

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

[10] W. Zuo and Z. Lin, "A generalized accelerated proximal gradient approach for total-variation-based image restoration," *IEEE Trans. Image Process.*, vol. 20, no. 10, pp. 2748–2759, Oct. 2011.

[11] Y. Lou, T. Zeng, S. Osher, and J. Xin, "A weighted difference of anisotropic and isotropic total variation model for image processing," *SIAM J. Imag. Sci.*, vol. 8, no. 3, pp. 1798–1823, 2015.

[12] G. Yuan and B. Ghanem, "$l_0$tv: A new method for image restoration in the presence of impulse noise," in *Proc. CVPR*, Jun. 2015, pp. 5369–5377.

[13] T. F. Chan and C.-K. Wong, "Total variation blind deconvolution," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 370–375, Mar. 1998.

[14] D. Perrone and P. Favaro, "Total variation blind deconvolution: The devil is in the details," in *Proc. CVPR*, Jun. 2014, pp. 2909–2916.

[15] A. Marquina and S. J. Osher, "Image super-resolution by TV-regularization and Bregman iteration," *J. Sci. Comput.*, vol. 37, no. 3, pp. 367–382, Dec. 2008.

[16] J. Shen and T. F. Chan, "Mathematical models for local nontexture inpaintings," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2002.

[17] C.-A. Deledalle, N. Papadakis, and J. Salmon, "On debiasing restoration algorithms: Applications to total-variation and nonlocal-means," in *Scale Space and Variational Methods in Computer Vision*. Cham, Switzerland: Springer, 2015, pp. 129–141.

[18] C.-A. Deledalle, N. Papadakis, J. Salmon, and S. Vaiter, "CLEAR: Covariant least-square refitting with applications to image restoration," *SIAM J. Imag. Sci.*, vol. 10, no. 1, pp. 243–284, 2017.

[19] E.-M. Brinkmann, M. Burger, J. Rasch, and C. Sutour, "Bias reduction in variational regularization," *J. Math. Imag. Vis.*, vol. 59, no. 3, pp. 534–566, 2017.

[20] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 15–19, 2008.

[21] A. Mosleh, J. P. Langlois, and P. Green, "Image deconvolution ringing artifact detection and removal via PSF frequency analysis," in *Proc. ECCV*. Cham, Switzerland: Springer, 2014, pp. 247–262.

[22] D. Gong, M. Tan, Y. Zhang, A. van den Hengel, and Q. Shi, "MPGL: An efficient matching pursuit method for generalized lasso," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1934–1940.

[23] J. Wang, Q. Li, S. Yang, W. Fan, P. Wonka, and J. Ye, "A highly scalable parallel algorithm for isotropic total variation models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 235–243.

[24] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 20, no. 6, pp. 1964–1977, 1999.

[25] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 937–966, 2005.

[26] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via L_0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, 2011, Art. no. 174.

[27] L. Xu, S. Zheng, and J. Jia, "Unnatural L_0 sparse representation for natural image deblurring," in *Proc. CVPR*, Jun. 2013, pp. 1107–1114.

[28] J. Yang, Y. Zhang, and W. Yin, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM J. Sci. Comput.*, vol. 31, no. 4, pp. 2842–2865, 2009.

[29] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2010, pp. 157–170.

[30] C. Clason, "L^∞ fitting for inverse problems with uniform noise," *Inverse Problems*, vol. 28, no. 10, p. 104007, 2012.

[31] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 460–489, 2005.

[32] M. Burger, "Bregman distances in inverse problems and partial differential equations," in *Advances in Mathematical Modeling, Optimization and Optimal Control*. Cham, Switzerland: Springer, 2016, pp. 3–33.

[33] G. Gilboa, N. Sochen, and Y. Y. Zeevi, "Variational denoising of partly textured images by spatially varying constraints," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2281–2289, Aug. 2006.

[34] K. Bredies, Y. Dong, and M. Hintermüller, "Spatially dependent regularization parameter selection in total generalized variation models for image restoration," *Int. J. Comput. Math.*, vol. 90, no. 1, pp. 109–123, 2013.

[35] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[36] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, p. 70, 2007.

[37] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image restoration by sparse 3D transform-domain collaborative filtering," *Proc. SPIE*, vol. 6812, p. 681207, Mar. 2008.

[38] A. Kheradmand and P. Milanfar, "A general framework for regularized, similarity-based image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5136–5151, Dec. 2014.

[39] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Progressive inter-scale and intra-scale non-blind image deconvolution," *ACM Trans. Graph.*, vol. 27, no. 3, 2008, Art. no. 74.

[40] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.

[41] L. Xiao, J. Gregson, F. Heide, and W. Heidrich, "Stochastic blind motion deblurring," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3071–3085, Oct. 2015.

[42] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Schölkopf, "A machine learning approach for non-blind image deconvolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1067–1074.

[43] L. Xu, J. S. J. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1790–1798.

[44] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 2774–2781.

[45] D. Gong, Z. Zhang, Q. Shi, A. van den Hengel, C. Shen, and Y. Zhang. (2018). "Learning an optimizer for image deconvolution." [Online]. Available: https://arxiv.org/abs/1804.03368

[46] U. Schmidt, Q. Gao, and S. Roth, "A generative perspective on MRFs in low-level vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1751–1758.

[47] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Trans. Graph*, vol. 26, no. 3, p. 1, Jul. 2007.

[48] S. Cho, J. Wang, and S. Lee, "Handling outliers in non-blind image deconvolution," in *Proc. ICCV*, Nov. 2011, pp. 495–502.

[49] O. Whyte, J. Sivic, and A. Zisserman, "Deblurring shaken and partially saturated images," *Int. J. Comput. Vis.*, vol. 110, no. 2, pp. 185–201, 2014.

[50] M. Tan, I. W. Tsang, and L. Wang, "Matching pursuit LASSO Part I: Sparse recovery over big dictionary," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 727–741, Feb. 2015.

[51] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[52] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[53] E. Y. Pee and J. O. Royset, "On solving large-scale finite minimax problems using exponential smoothing," *J. Optim. Theory Appl.*, vol. 148, no. 2, pp. 390–421, 2011.

[54] K. O. Kortanek and H. No, "A central cutting plane algorithm for convex semi-infinite programming problems," *SIAM J. Optim.*, vol. 3, no. 4, pp. 901–918, 1993.

[55] M. Tan, I. W. Tsang, and L. Wang, "Towards ultrahigh dimensional feature selection for big data," *J. Mach. Learn. Res.*, vol. 15, pp. 1371–1429, Apr. 2014.

[56] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. CVPR*, Jun. 2009, pp. 1964–1971.

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

[58] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[59] J. Portilla, "Image restoration through l0 analysis-based sparse optimization in tight frames," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 3909–3912.

[60] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via L_0-regularized intensity and gradient prior," in *Proc. CVPR*, Jun. 2014, pp. 2901–2908.

[61] L. He and Y. Wang, "Iterative support detection-based split Bregman method for wavelet frame-based image inpainting," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5470–5485, Dec. 2014.

Dong Gong received the B.S. degree in computer science from Northwestern Polytechnical University, Xi'an, China. He is currently pursuing the Ph.D. degree with the School of Computer Science, Northwestern Polytechnical University. He was the Joint-Training Ph.D. Student with The University of Adelaide from 2015 to 2016. His current research interests include machine learning and optimization techniques and their applications in image processing and computer vision.

Mingkui Tan received the bachelor's degree in environmental science and engineering and the master's degree in control science and engineering from Hunan University, China, in 2006 and 2009, respectively, and the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2014. From 2014 to 2016, he was a Senior Research Associate in computer vision with the School of Computer Science, University of Adelaide, Australia. Since 2016, he has been with the School of Software Engineering, South China University of Technology, China, where he is currently a Professor. His research interests include machine learning, sparse analysis, deep learning, and large-scale optimization.

Qinfeng Shi received the bachelor's and master's degrees in computer science and technology from Northwestern Polytechnical University, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science in machine learning from Australian National University, Canberra, ACT, Australia, in 2011. He is currently an Associate Professor with the School of Computer Science, The University of Adelaide, Adelaide, SA, Australia.

Anton van den Hengel received the B.Sc. degree in mathematics, the B.L. degree, the M.Sc. degree in computer science, and the Ph.D. degree in computer vision from The University of Adelaide in 1991, 1993, 1994, and 2000, respectively. He is currently a Professor with the University of Adelaide and the Founding Director of The Australian Centre for Visual Technologies.

Yanning Zhang received the B.S. degree from the Dalian University of Technology in 1988 and the M.S. and Ph.D. degrees from Northwestern Polytechnical University in 1993 and 1996, respectively. She is currently a Professor with the School of Computer Science and Technology, Northwestern Polytechnical University. She has published over 200 papers in his research fields including ICCV 2011 Best Student Papers. Her research works focus on signal and image processing, computer vision, and pattern recognition. She was the Organization Chair of ACCV 2009 and the Publicity Chair of ICME 2012.