Blind Image Deconvolution by Automatic Gradient Activation

Dong Gong[†‡], Mingkui Tan[‡], Yanning Zhang[†], Anton van den Hengel[‡], Qinfeng Shi[‡*]
[†]School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[‡]School of Computer Science, The University of Adelaide, Australia
[‡]{dong.gong,mingkui.tan,anton.vandenhengel,javen.shi}@adelaide.edu.au [†]ynzhang@nwpu.edu.cn

Abstract

Blind image deconvolution is an ill-posed inverse problem which is often addressed through the application of appropriate prior. Although some priors are informative in general, many images do not strictly conform to this, leading to degraded performance in the kernel estimation. More critically, real images may be contaminated by nonuniform noise such as saturation and outliers. Methods for removing specific image areas based on some priors have been proposed, but they operate either manually or by defining fixed criteria.

We show here that a subset of the image gradients are adequate to estimate the blur kernel robustly, no matter the gradient image is sparse or not. We thus introduce a gradient activation method to automatically select a subset of gradients of the latent image in a cutting-plane-based optimization scheme for kernel estimation. No extra assumption is used in our model, which greatly improves the accuracy and flexibility. More importantly, the proposed method affords great convenience for handling noise and outliers. Experiments on both synthetic data and real-world images demonstrate the effectiveness and robustness of the proposed method in comparison with the state-of-the-art methods.

1. Introduction

Image blur can be caused by a variety of factors, from camera movement [9, 29], to depth-of-field effects [21]. Blur removal has thus become one of the standard functions within image processing packages, and anyone who has used them will understand some of these methods' deficiencies. Assuming that the blur is spatially invariant, the observation process of a blurred image $\mathbf{y}$, can be modeled as a convolution between a latent sharp image $\mathbf{x}$ and some unknown blur kernel (or point spread function, PSF) $\mathbf{k}$:

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

where $*$ denotes the convolution operator, and $\mathbf{n}$ usually refers to additive noise (sampled from an i.i.d. zero-mean Gaussian distribution [9]). Here $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{k} \in \mathbb{R}^m$ and $\mathbf{n} \in \mathbb{R}^n$. The blur model in (1) can also be represented in matrix-vector form [19, 36]:

$$\mathbf{y} = \mathbf{H}(\mathbf{k})\mathbf{x} + \mathbf{n} = \mathbf{A}(\mathbf{x})\mathbf{k} + \mathbf{n}\,, \tag{2}$$

where $\mathbf{H}(\mathbf{k}) \in \mathbb{R}^{n\times n}$ and $\mathbf{A}(\mathbf{x}) \in \mathbb{R}^{n\times m}$ are convolution matrices associated with $\mathbf{k}$ and $\mathbf{x}$, respectively. In either formulation the blur kernel and latent (unblurred) image are unseen, and intrinsically linked. Errors in the estimate of either inevitably impact upon the quality of the other.

Blind deconvolution, also known as *blind deblurring*, seeks to recover the latent sharp image $\mathbf{x}$ from the observed blurry image $\mathbf{y}$. Blind deblurring is a highly ill-posed inverse problem since one has to estimate $\mathbf{x}$ and $\mathbf{k}$ simultaneously [22, 27, 36]. Solving the problem thus requires additional assumptions or priors on $\mathbf{x}$ and $\mathbf{k}$ [9, 29, 5, 23, 19, 38, 39, 36]. For example, ℓ_2-norm [5, 17, 39] or ℓ_1-norm [29, 18] regularization is often imposed on $\mathbf{k}$ to encourage the smoothness or sparsity. Most methods similarly assume that $\mathbf{x}$ conforms to a prior such as Total-Variation (TV) over the latent sharp image [4, 29]. However, such priors may not be adequate for blind image deblurring as they may favor the trivial solution $\mathbf{x} = \mathbf{y}$ with $\mathbf{k} = \boldsymbol{\delta}$ (i.e., the identity kernel) [22].

A range of methods have been developed to identify and ignore problematic areas of the image [5, 38, 13, 36]. Whether defined in terms of the image areas to include, or exclude, these methods develop heuristics, such as manually defined edge selection criterion [5, 38] and edge reweighting [18, 19]. Many researchers have also applied a sparsity prior on image gradients for kernel estimation [39, 17, 36, 28], or approximate ℓ_0-norm regularization [39, 26].

These methods have achieved state-of-the-art performance on several benchmark data sets [22, 15]. However,

*This work is supported by the National Natural Science Foundation of China (No.61231016, 61301192, 61303123, 61301193), Australian Research Council grants (DP140102270, DP160100703), and the Data 2 Decisions CRC.

1063-6919/16 $31.00 © 2016 IEEE
DOI 10.1109/CVPR.2016.202

they exhibit some important limitations in practice. First, for many real-world images (particularly images with rich textures), the sparse prior may not hold, thus a simple sparse regularizer may not be adequate. Similarly, methods like bilateral filtering [5] may not find the strong edges they require as cues. Second, these methods may be sensitive to large noise [10, 31, 44]. Even significant image gradients become less identifiable as noise increases, and the probability of being distracted mistakenly increases also.

1.1. Contributions

- Unlike existing methods relying on sharp edge prediction or sparse regularizers, we introduce a gradient activation scheme to directly detect significant gradients that are beneficial for kernel estimation. This idea is motivated by an important observation that the blur kernel can be estimated based on a subset of the gradients, no matter the gradients are sparse or not (See Proposition 1).
- Relying on the above observation, we formulate the gradient activation problem as a convex optimization problem and address it via a cutting-plane method. After that, the blur kernel estimation will be conducted based on the activated gradient only.
- The proposed cutting-plane scheme allows us to easily drop the problematic components, such as saturation, and hence it affords great convenience for handling noise and outliers.

1.2. Notation

Throughout the paper, let $\|\mathbf{v}\|_p$ be the ℓ_p-norm and $\odot$ be the element-wise product. Given a positive integer n, let $[n]$ be the set $\{1, ..., n\}$. Given any set $\mathcal{S} \subseteq [n]$, $|\mathcal{S}|$ denotes the cardinality of $\mathcal{S}$, and $\mathcal{S}^c$ is the complementary set, namely $\mathcal{S}^c = [n] \setminus \mathcal{S}$. For any vector $\mathbf{x}$, let the calligraphic letter $\mathcal{X} = \text{support}(\mathbf{x}) = \{i | x_i \neq 0\} \in [n]$ be its support. For a vector $\mathbf{v} \in \mathbb{R}^n$ and any $\mathcal{S} \subseteq [n]$, let $\mathbf{v}_\mathcal{S}$ denote the subvector indexed by $\mathcal{S}$, and v_i be the i-th component of $\mathbf{v}$. $\nabla\mathbf{x}$ denotes the gradients of $\mathbf{x}$.

2. Related Work

Existing methods can be mainly categorized into two groups, maximum a posterior (MAP) methods [29, 5, 39] and variational Bayesian (VB) methods [23, 9, 36].

MAP based methods. MAP methods seek to minimize the negative log-posterior $-\log p(\mathbf{y}|\mathbf{x}, \mathbf{k})p(\mathbf{x})p(\mathbf{k})$ where $p(\mathbf{y}|\mathbf{x}, \mathbf{k})$ is the likelihood related to equation (1), $p(\mathbf{x})$ and $p(\mathbf{k})$ denote the priors on $\mathbf{x}$ and $\mathbf{k}$, respectively. The minimization can be achieved by addressing the problem:

$$\min_{\mathbf{k},\mathbf{x}} \frac{1}{2}\|\mathbf{y} - \mathbf{k} * \mathbf{x}\|_2^2 + \lambda\Omega(\mathbf{x}) + \gamma\Upsilon(\mathbf{k}), \tag{3}$$

where λ and γ are trade-off parameters, and $\Omega(\mathbf{x})$ and $\Upsilon(\mathbf{k})$ are regularizers that reflect the priors. Choosing appropriate priors on $\mathbf{x}$ and $\mathbf{k}$, or equivalently regularizers $\Omega(\mathbf{x})$ and $\Upsilon(\mathbf{k})$, is critical for the performance. Note that many existing works operate in the gradient domain rather than the image domain. In this case, $\mathbf{x}$ and $\mathbf{y}$ in model (3) are simply replaced with $\nabla\mathbf{x}$ and $\nabla\mathbf{y}$, respectively [9, 19].

Researchers have shown that promoting sparsity on image gradients is helpful for kernel estimation [9, 39, 36]. To achieve this, some methods explicitly promote sparsity to prune detrimental structures using carefully designed regularizers. For example, Chan and Wong use an ℓ_1-norm on image gradients, called the TV norm [4], to choose salient edges. However, this prior may favor a blurry image in certain conditions [22, 27]. Shan *et.al.* [29] thus combine the ℓ_1-norm and a ring suppressing term. Other methods to suppress small structures include the ℓ_1/ℓ_2-norm regularization [19] and other reweighted norms [17]. Xu *et.al.* [39] propose an approximate ℓ_0-norm regularizer on gradients, but the resultant problem is intractable or can be only solved approximately. Zuo *et.al.* [46] learn iteration-wise hyper-Laplacian prior on gradients.

Rather than using sparse regularizers, some methods explicitly extract salient structures for kernel estimation. For example, Joshi *et.al.* [14, 5] detect edges from the blurry image for kernel estimation. Cho and Lee [5] extract edges using a bilateral filter and a shock filter. Sun *et.al.* [30] and Lai *et.al.* [20] predict edges in a similar way to [5], but they further refine edges using data-driven priors. Xu and Jia [38] select informative structures and remove tiny edges based on the relative total variation. Zhou and Komodakis [45] detect edges using a high-level scene-specific prior. However, these edge prediction methods may fail if there are not enough strong structures [25]. Regarding this, some example-based methods [25, 11] are proposed. Other methods to generate priors can be found in [16, 45, 24, 37].

VB based methods. Unlike MAP methods, VB methods estimate the blur kernel by minimizing an upper bound of $-\log p(\mathbf{k}|\mathbf{y})$, which is the negative log-marginal distribution of $\mathbf{k}$, i.e., $p(\mathbf{k}|\mathbf{y}) = \int_\mathbf{x} p(\mathbf{x}, \mathbf{k}|\mathbf{y})d\mathbf{x}$ [9, 22, 23, 1, 36]. Wipf and Zhang [36] show that minimizing the upper bound using VB methods is equivalent to solving an MAP problem with a regularizer coupling $\mathbf{x}$ and $\mathbf{k}$. There are some methods beyond MAP and VB, such as [10, 3, 43].

Handling noise and outliers. Most of the above methods are sensitive to noise and outliers. Recently, some methods focus on handling noise and outliers (e.g. saturation). Tai and Lin [31] propose to estimate the blur kernel and remove noise alternatively. Zhong *et.al.* [44] use a set of directional filters to remove noise. Zhang and Yang [42] estimate the kernel from a set of blurry images smoothed by Gaussian filters in different scales. Hu *et.al.* [12] explicitly detect saturated pixels and light streaks and handle them separately in

kernel estimation.

3. Deblur by Gradient Activation

The proposed blur kernel estimation method applies only a specific, but as-yet-undetermined, subset of the image gradients. During the kernel estimation, a subset of non-zero gradients will be activated and updated for kernel estimation.

3.1. Motivation

The quality of the deblurred image depends critically upon the quality of blur kernel estimate. We here show that, whether the gradient image is sparse or not, the blur kernel can be estimated from a subset of the gradient image. This observation is the corner stone of our framework. To demonstrate this fact, we begin with the common formulation [9, 19, 41, 36],

$$\nabla\mathbf{y} = \mathbf{H}(\mathbf{k})\nabla\mathbf{x} + \nabla\mathbf{n} = \mathbf{A}(\nabla\mathbf{x})\mathbf{k} + \nabla\mathbf{n}\,, \qquad (4)$$

where $\nabla\mathbf{y}$, $\nabla\mathbf{x}$ and $\nabla\mathbf{n}$ are the gradients of $\mathbf{y}$, $\mathbf{x}$ and the noise $\mathbf{n}$, respectively. Given any intermediate $\nabla\mathbf{x}$, based on model (3), we may estimate the kernel by solving

$$\min_{\mathbf{k}} \|\nabla\mathbf{y} - \nabla\mathbf{x} * \mathbf{k}\|_2^2 + \gamma\|\mathbf{k}\|_2^2, \text{s.t.} \|\mathbf{k}\|_1 = 1, \mathbf{k} \geq \mathbf{0}. \quad (5)$$

where $\mathbf{k}$ is constrained in a simplex derived from [4]. Here the ℓ_2-norm regularizer $\|\mathbf{k}\|_2^2$ is often used to avoid the trivial solutions where $\mathbf{x} = \mathbf{y}$.

Let $\nabla\mathbf{x}_0$ be gradient of the ground truth image $\mathbf{x}_0$, and $\mathbf{k}_0$ be the ground truth kernel[1]. Hereafter, we **focus on only a subset of** $\nabla\mathbf{x}_0$ which is indexed by a set $\mathcal{S}$. Let $\varrho = \|\nabla\mathbf{x}_{0\mathcal{S}}\|_2$ and $\mathbf{A}(\nabla\mathbf{x}_{0\mathcal{S}})$ be the corresponding convolution matrix regarding $\nabla\mathbf{x}_{0\mathcal{S}}$. Moreover, based on $\nabla\mathbf{x}$ and $\mathcal{S}$, we define a new vector $\widetilde{\nabla\mathbf{x}}$ such that

$$\widetilde{\nabla\mathbf{x}}_{\mathcal{S}^c} = \mathbf{0} \;\text{ and }\; \widetilde{\nabla\mathbf{x}}_{\mathcal{S}} = \nabla\mathbf{x}_{\mathcal{S}}. \qquad (6)$$

Focusing on the subset $\nabla\mathbf{x}_{0\mathcal{S}}$, problem (5) for kernel estimation is equivalent to the following problem:

$$\min_{\mathbf{k}} \|\nabla\mathbf{y} - \widetilde{\nabla\mathbf{x}} * \mathbf{k}\|_2^2 + \gamma\|\mathbf{k}\|_2^2, \;\; \text{s.t.} \;\; \mathbf{k} \in \mathcal{K}, \qquad (7)$$

where $\mathcal{K} = \{\mathbf{k} : \|\mathbf{k}\|_1 = 1, k_i \geq 0, \forall i\}$ be the domain of $\mathbf{k}$. By neglecting the boundary effect of convolution, we then have the following proposition.

Proposition 1. *For any intermediate $\nabla\mathbf{x}$, let $\mathbf{e} = \nabla\mathbf{x} - \nabla\mathbf{x}_0$, $\epsilon^2 = \|\mathbf{e}_{\mathcal{S}}\|_2^2$ and $\Delta = (\epsilon^2 + 3\epsilon\varrho + \gamma)$. If $\nabla\mathbf{x}_{0\mathcal{S}}$ satisfies $\inf_{\|\mathbf{v}\|_2=1} \|\mathbf{A}(\nabla\mathbf{x}_{0\mathcal{S}})\mathbf{v}\|_2 = \omega > \Delta$, the solution to (7), denoted by $\widehat{\mathbf{k}}$, follows*

$$\|\widehat{\mathbf{k}} - \mathbf{k}_0\|_2 \leq \frac{\Delta}{\omega - \Delta}\|\mathbf{k}_0\|_2 + \frac{\epsilon + \varrho}{\omega - \Delta}\|\nabla\mathbf{n}_{\mathcal{S}}\|_2.$$

[1]We assume that the image is much larger than the latent blur kernel, e.g. $n \gg m$, and hence the boundary effect of convolution can be neglected.

The proof can be found in the long version of this paper.

Remark 1. *From the definition of Δ, the accuracy of kernel estimation is not dependent on the sparsity of image gradient. Conversely, it is mainly affected by three factors, i.e., the accuracy of $\nabla\mathbf{x}$ to $\nabla\mathbf{x}_0$ w.r.t. $\mathcal{S}$ (i.e., $\epsilon = \|\mathbf{e}_{\mathcal{S}}\|$), the chosen $\nabla\mathbf{x}_{0\mathcal{S}}$ which determines ϱ, and the parameter γ.*

From Remark 1, to estimate $\mathbf{k}$ accurately, we need to obtain an accurate $\nabla\mathbf{x}$. We address this by a two-step iterative approach, i.e., compute the intermediate $\nabla\mathbf{x}$ with $\mathbf{k}$ and then estimate the kernel $\mathbf{k}$ with $\nabla\mathbf{x}$ (See Algorithm 2) which is motivated by the following corollary and remarks.

Corollary 1. *In Proposition 1, $\|\mathbf{A}(\nabla\mathbf{x}_{0\mathcal{S}})\mathbf{v}\|_2 = \|\nabla\mathbf{z}\|_2 = \left(\sum_{i\in\mathcal{S}'}(\nabla z_i)^2\right)^{1/2}$ where $\nabla\mathbf{z} = \widetilde{\nabla\mathbf{x}} * \mathbf{v}$ and $\mathcal{S}'$ is the support of $\nabla\mathbf{z}$. It implies that, given size of the kernel $\mathbf{v}$, $\inf_{\|\mathbf{v}\|_2=1} \|\mathbf{A}(\nabla\mathbf{x}_{0\mathcal{S}})\mathbf{v}\|_2$ is large only if $\|\widetilde{\nabla\mathbf{x}}\|_2$ (also $\|\nabla\mathbf{x}_{0\mathcal{S}}\|_2$) is large.*

Remark 2. *From Corollary 1, to ensure the condition $\inf_{\|\mathbf{v}\|_2=1} \|\mathbf{A}(\nabla\mathbf{x}_{0\mathcal{S}})\mathbf{v}\|_2 = \omega > \Delta > 0$, **1)** the size of $\mathcal{S}$ (i.e. $|\mathcal{S}|$) should be large enough, and **2)** the magnitude of elements in $\nabla\mathbf{x}_{0\mathcal{S}}$ should be as large as possible. Conversely, insignificant gradients may lead to poor performance in kernel estimation and could be even suppressed by noise.*

Remark 3. *It seems that a small γ would be favorable when estimating $\mathbf{k}$. However, if γ is too small, it may incur trivial solution. On the simple noiseless case, setting $\gamma = 0$ in (7) at the beginning where $\nabla\mathbf{x} = \nabla\mathbf{y}$ will lead to $\mathbf{k} = \boldsymbol{\delta}$ regarding any $\mathcal{S}$. This suggests applying a homotopy strategy for setting γ, that is, monotonically decreasing γ from a large initial value to a small target value.*

3.2. Proposed Model

From Remark 1 we see that the accuracy of the kernel estimate is not dependent on the sparsity of the gradient image. Motivated by this, rather than enforcing sparsity over the whole latent $\nabla\mathbf{x}$, we introduce a binary activation vector $\boldsymbol{\tau} \in \{0, 1\}^n$ which indicates the areas where we believe the prior applies. Having specified $\boldsymbol{\tau}$ allows us to ignore other latent image gradients by considering only $(\nabla\mathbf{x} \odot \boldsymbol{\tau})$. We constrain $\|\boldsymbol{\tau}\|_0 \leq \kappa$, where the integer κ controls the the number of latent image gradients not set to zero. To avoid the degenerate solution, one may impose additional constraints on $\boldsymbol{\tau}$. For example, considering that the candidates of activated gradients should not be isolated, we can constrain $\boldsymbol{\tau}$ to exclude distracting gradients or outliers (more detail is provided in Section 4.2). For convenience, let $\Lambda = \{\boldsymbol{\tau} : \|\boldsymbol{\tau}\|_0 \leq \kappa, \boldsymbol{\tau} \in \{0, 1\}^n, \boldsymbol{\tau} \in \Pi\}$ be the domain of $\boldsymbol{\tau}$, where Π represent some additional constraint set.

Algorithm 1: Kernel Estimation by Gradient Activation

Input: Blurry image $\mathbf{y}$, $\eta < 1$, γ^{tar}, initial $\mathbf{k}^1$ and γ^1.

for $t = 1 : T$ **do**

  Given $\mathbf{k}^t$, find set $\mathcal{S}^t = \text{support}(\boldsymbol{\tau}^t)$ by solving (9), and obtain $\nabla\mathbf{x}^t_{\mathcal{S}}$.

  Given $\mathcal{S}^t$, $\nabla\mathbf{x}^t_{\mathcal{S}}$ and γ^t, estimate $\mathbf{k}^t$ by solving (7).

  Set $\gamma^{t+1} = \max(\eta\gamma^t, \gamma^{tar})$.

Return $\mathbf{k}^t$.

With the introduction of $\boldsymbol{\tau}$, our image deblurring model can be formulated into the following optimization problem:

$$\min_{\boldsymbol{\tau}\in\Lambda,\mathbf{k},\nabla\mathbf{x}} \frac{1}{2}\|\nabla\mathbf{y}-\mathbf{k}*(\nabla\mathbf{x}\odot\boldsymbol{\tau})\|_2^2+\lambda\Omega(\nabla\mathbf{x}\odot\boldsymbol{\tau})+\gamma||\mathbf{k}||_2^2\,, \tag{8}$$

where $\Omega(\nabla\mathbf{x}\odot\boldsymbol{\tau})$ denotes any regularizer that might be applied to $(\nabla\mathbf{x}\odot\boldsymbol{\tau})$. Note that $\boldsymbol{\tau}$ is directly coupled with $\nabla\mathbf{x}$. Once obtaining $\boldsymbol{\tau}$, we can immediately obtain $\widetilde{\nabla\mathbf{x}}$ according to equation (6).

Note that there are $\sum_{i=0}^{\kappa}\binom{n}{i}$ feasible $\boldsymbol{\tau}$'s in Λ. The problem of how to choose the best $\boldsymbol{\tau}$ still remains. From Remark 2, the activated gradients should reflect the observed gradients in $\nabla\mathbf{y}$ (so that $\|\nabla\mathbf{x}_{0\mathcal{S}}\|$ will be large). Given fixed $\mathbf{k}$, based on (8), we propose to find the best $\boldsymbol{\tau}$ from the feasible set to minimize the following objective:

$$\min_{\boldsymbol{\tau}\in\Lambda}\left(\min_{\nabla\mathbf{x}}\frac{1}{2}\|\nabla\mathbf{y}-\mathbf{k}*(\nabla\mathbf{x}\odot\boldsymbol{\tau})\|_2^2+\lambda\Omega(\nabla\mathbf{x}\odot\boldsymbol{\tau})\right). \tag{9}$$

This problem is non-convex. To address it efficiently, we introduce a tight convex relation of this problem, and then propose a cutting plane algorithm to solve the relaxed problem instead. For convenience, we leave details in Section 4. After obtaining $\mathcal{S}$ coupled with $\nabla\mathbf{x}$, we now address problem (7) to update the kernel $\mathbf{k}$.

3.3. General Optimization Scheme

Based on the gradient activation model, the general scheme of the proposed kernel estimation, which is shown in Algorithm 1[2]. In Step 2, with fixed $\mathbf{k}$, we solve problem (9) to select a subset of non-zero gradients indicated by set $\mathcal{S}$ and obtain $\nabla\mathbf{x}_{\mathcal{S}}$. In Step 3, with fixed $\mathcal{S}$, $\nabla\mathbf{x}_{\mathcal{S}}$, we construct $\widetilde{\nabla\mathbf{x}}$ by equation 6, and then estimate $\mathbf{k}^t$ by solving problem (7) which is addressed by a proximal gradient method [8]. Note that in Step 4, motivated by Remark 3, we apply a homotopy strategy to set the regularization parameter γ, which monotonically decreases from a initial value γ^1 to a target value γ^{tar}.

[2]To improve the robustness, Algorithm 1 is implemented in a coarse-to-fine scheme following the image pyramid.

After obtaining the final kernel, we will recover the sharp image by non-blind deconvolution algorithms, for instance, the hyper-Laplacian based method [18].

Setting κ. In this paper, we set κ adaptively. Let $\boldsymbol{\beta} = \mathbf{H}^\mathsf{T}\nabla\mathbf{y}$. We set κ as the number of β_j that satisfies $\beta_j \geqslant \eta\|\boldsymbol{\beta}\|_\infty$, where $\eta \in (0, 1]$. In general, a bigger η will lead to a smaller κ. In this paper, we empirically set η to 0.5.

Now we highlight some differences and advantages of our model over existing methods. (1) Unlike existing methods which apply various sparse regularizers to induce a sparse solution, we use $\boldsymbol{\tau}$ to explicitly and automatically detect a subset of non-zero gradients. Even $\nabla\mathbf{x}$ may not be sparse, due to the constraint $\|\boldsymbol{\tau}\|_0 \leq \kappa$, our proposed method (See Algorithm 2) can still find a sparse subset of non-zero gradients for kernel estimation. (2) It is worth mentioning that, if the gradients are not sparse, using sparse regularizer may be suboptimal as one should sacrifice fitting accuracy to achieve sparsity. In fact, the sparsity of the solution is often controlled by the parameter λ: a large λ is necessary for inducing sparsity but often causes solution bias [7]; while a small λ will enforce non-sparse solutions. In our model, relying on the constraint $\|\boldsymbol{\tau}\|_0 \leq \kappa$ and the optimization scheme (Algorithm 2), a small λ can be used to achieve high fitting accuracy on the required image areas (See Figure 4(a) for the importance of a small λ).

4. Activate Gradients Automatically

We are now ready to solve problem (9). Let $\mathbf{H}(\mathbf{k})$ be the convolution matrix of $\mathbf{k}$, and hence we have $\nabla\mathbf{y}-\mathbf{k}*(\nabla\mathbf{x}\odot\boldsymbol{\tau}) = \nabla\mathbf{y} - \mathbf{H}(\mathbf{k})(\nabla\mathbf{x}\odot\boldsymbol{\tau})$. Since $\mathbf{k}$ is fixed at this stage, for simplicity, we use $\mathbf{H}$ to denote the convolution matrix $\mathbf{H}(\mathbf{k})$. Here we use $\Omega(\nabla\mathbf{x}\odot\boldsymbol{\tau}) = \frac{1}{2}\|\nabla\mathbf{x}\odot\boldsymbol{\tau}\|_2^2$.

By introducing an auxiliary variable $\boldsymbol{\xi} = \nabla\mathbf{y}-\mathbf{H}(\nabla\mathbf{x}\odot\boldsymbol{\tau})$, problem (9) can be rewritten as

$$\begin{aligned}\min_{\boldsymbol{\tau}\in\Lambda}\min_{\nabla\mathbf{x},\,\boldsymbol{\xi}} \quad & \frac{\lambda}{2}\|\nabla\mathbf{x}\odot\boldsymbol{\tau}\|_2^2+\frac{1}{2}\|\boldsymbol{\xi}\|_2^2,\\ \text{s.t.} \quad & \boldsymbol{\xi} = \nabla\mathbf{y}-\mathbf{H}(\nabla\mathbf{x}\odot\boldsymbol{\tau}).\end{aligned} \tag{10}$$

This problem is a non-convex problem, so it is difficult to solve. Motivated by [32], we solve a convex relaxation of problem (10). To achieve this, we introduce a dual variable $\boldsymbol{\alpha} \in \mathbb{R}^n$ to the constraint $\boldsymbol{\xi} = \nabla\mathbf{y} - \mathbf{H}(\nabla\mathbf{x}\odot\boldsymbol{\tau})$ w.r.t. any fixed $\boldsymbol{\tau}$, and transform (10) into a minimax problem by introducing the dual form of the inner problem of (10):

$$\min_{\boldsymbol{\tau}\in\Lambda}\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \boldsymbol{\alpha}^\mathsf{T}\nabla\mathbf{y}-\frac{\boldsymbol{\alpha}^\mathsf{T}\mathbf{H}\text{diag}(\boldsymbol{\tau})\mathbf{H}^\mathsf{T}\boldsymbol{\alpha}}{2\lambda}-\frac{\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\alpha}}{2}.$$

Define $\psi(\boldsymbol{\alpha},\boldsymbol{\tau})=\frac{1}{2\lambda}\boldsymbol{\alpha}^\mathsf{T}\mathbf{H}\text{diag}(\boldsymbol{\tau})\mathbf{H}^\mathsf{T}\boldsymbol{\alpha}+\frac{1}{2}\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\alpha}-\boldsymbol{\alpha}^\mathsf{T}\nabla\mathbf{y}$. By applying a convex relaxation $\min_{\boldsymbol{\tau}\in\Lambda}\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\psi(\boldsymbol{\alpha},\boldsymbol{\tau}) \geq \max_{\boldsymbol{\alpha}\in\mathbb{R}^n}\min_{\boldsymbol{\tau}\in\Lambda} -\psi(\boldsymbol{\alpha},\boldsymbol{\tau})$ [2], we achieve the following quadratically constrained quadratic program (QCQP) problem:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n,\theta\in\mathbb{R}} \theta,\ \text{s.t.}\ \psi(\boldsymbol{\alpha},\boldsymbol{\tau})\leq\theta,\ \forall\,\boldsymbol{\tau}\in\Lambda\,, \tag{11}$$

where $\theta \in \mathbb{R}$ is an auxiliary variable.

Note that there are many elements in Λ (i.e., $|\Lambda| = \sum_{i=0}^{\kappa} \binom{n}{i}$). Therefore, problem (11) involves many constraints. We then adopt a cutting-plane algorithm, which iteratively detects the most-active constraint from Λ. The overall cutting-plane algorithm is shown in Algorithm 2. In Step 4, the most-active constraint selection reflects the activation of the appropriate gradients. Let

$$\mathbf{g} = \mathbf{H}^{\mathsf{T}} \boldsymbol{\xi}^{i-1}. \tag{12}$$

The most-active constraint can be selected by choosing the κ largest $|g_j|$ and recording their indices into a set $\mathcal{C}^i$. After that, we update $\mathcal{S}^i$ by $\mathcal{S}^i = \mathcal{S}^{i-1} \cup \mathcal{C}^i$, and update $\nabla \mathbf{x}_{\mathcal{S}^i}$ by solving a group lasso problem which is the primal form of problem (11) with chosen $\boldsymbol{\tau}$'s [32]:

$$\min_{\nabla \mathbf{x}_{\mathcal{S}^i}} \frac{1}{2} \|\nabla \mathbf{y} - \mathbf{H} \nabla \mathbf{x}_{\mathcal{S}^i}\|_2^2 + \frac{\lambda}{2} \left(\sum_{o=1}^{i} \|\nabla \mathbf{x}_{\mathcal{C}^o}\|_2 \right)^2, \tag{13}$$

where $\nabla \mathbf{x}_{\mathcal{S}^i} = [\nabla \mathbf{x}_{\mathcal{C}^1}^{\mathsf{T}}, ..., \nabla \mathbf{x}_{\mathcal{C}^i}^{\mathsf{T}}]^{\mathsf{T}}$.

In Algorithm 2, each iteration includes κ informative elements. Note that due to the relaxation, there may be more than one active constraints $\boldsymbol{\tau}$'s, and $|\mathcal{S}|$ can be larger than κ. This is reasonable since κ is just an estimation of the active gradients. Nevertheless, due to the introduction of $\boldsymbol{\tau}$ and an elegant early stopping strategy below, a subset of gradients will be obtained.

Algorithm 2: Solving (10) by a cutting-plane method

Input: Blurry image $\mathbf{y}$ and $\mathbf{H}$

1 Initialize $\nabla \mathbf{x}^0 = \mathbf{0}$, $\boldsymbol{\xi}^0 = \nabla \mathbf{y}$, $\mathcal{S}^0 = \emptyset$, $i = 0$;

2 **while** *Stopping conditions are not achieved* **do**

3 Let $i = i + 1$;

4 **Gradient activation:**
Compute $\mathbf{g} = \mathbf{H}^{\top} \boldsymbol{\xi}^{i-1}$, and record the indices of κ largest $|g_j|$ into $\mathcal{C}^u$, where $j \in [n] \setminus \mathcal{S}^{i-1}$;

5 Let $\mathcal{S}^i = \mathcal{S}^{i-1} \cup \mathcal{C}^i$;

6 Update $\nabla \mathbf{x}_{\mathcal{S}^i}$ by solving (13);

7 Update $\boldsymbol{\xi}$ by $\boldsymbol{\xi}^i = \nabla \mathbf{y} - \mathbf{H} \nabla \mathbf{x}^i$;

8 Compute $\mathcal{S} = \text{support}(\nabla \mathbf{x})$. Output $\nabla \mathbf{x}^i$ and $\mathcal{S}$.

4.1. Stopping Conditions

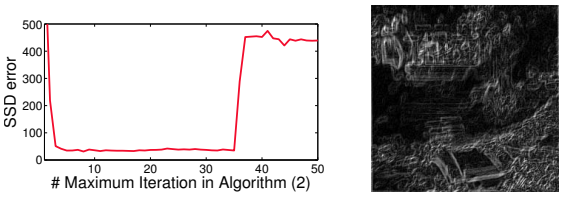
Early stopping in Algorithm 2 plays a very critical role in the proposed method. Here, we apply two early stopping conditions. First, we stop Algorithm 2 if

$$|f(\nabla \mathbf{x}^i) - f(\nabla \mathbf{x}^{i-1})| / |\kappa f(\nabla \mathbf{x}^0)| \leq \varepsilon, \tag{14}$$

where ε is a small tolerance and $f(\nabla \mathbf{x}^0)$ denotes the initial objective value. In this paper, we set $\varepsilon = 10^{-4}$ for all cases.

Early stopping when gradients are not exactly sparse. In this case, condition in (14) may be hard to achieve. We then stop the algorithm after a maximum number iterations $I_{\max}$ is achieved. In this paper, we set $I_{\max} = 15$.

The importance of the above condition is shown in Figure 1 on a toy image whose ground-truth is known and gradients are not exactly sparse (Figure 1. (b)). We report the sum-of-squared-difference (SSD) errors using different $I_{\max}$. From Figure 1. (a), the SSD error keeps stable when $5 < I_{\max} < 30$. However, when $I_{\max} > 35$, the SSD error becomes very large, where we choose a very large number of gradients and the algorithm fails. This verifies our Remark 2 that more gradients may even degrade the performance.



(a) (b)

Figure 1. (a) SSD error with increasing number of iterations on non-sparse gradients; (b) Gradients of the input image.

4.2. Handling Noise and Outliers

The performance of most existing kernel estimation methods may degrade significantly when there are non-negligible noise and/or outliers [44, 42] that cannot be handled by the priors [31] (See Figure 2. (c)-(d)).

The gradient activation scheme in Algorithm 2 provides great convenience for handling noise and outliers. Specifically, for uniform noise, relying on the early stopping strategies, noise contaminated weak gradients will not be included for the kernel estimation. As a result, the influence of uniform noise can be avoided (Figure 3 (e)).

Outliers are referred to non-Gaussian noise, which are unavoidable in some special scenarios such as low-light conditions, in which outliers usually appear as saturated pixels, strong light streaks, impulse noise and so on [6, 12]. In the gradient domain, those pixels often have very high values, but should not be activated. However, the sparse regularization methods will tend to choose them mistakenly.

To handle the **saturations and light streaks**, in Step 4 of Algorithm 2, we can simply discard those gradients corresponding to high-intensity pixels by forcing $\mathcal{C}^i \cap [\mathbf{y} > h] = \emptyset$, where $[\mathbf{y} > h]$ denotes the index set of pixels with intensity higher than a threshold h. Similarly, for **impulse noise**, they can be easily avoided by discarding isolated gradients (discarding any activated gradients with less than 4 elements), since impulse noise is often disconnected with other gradients in gradient domain.

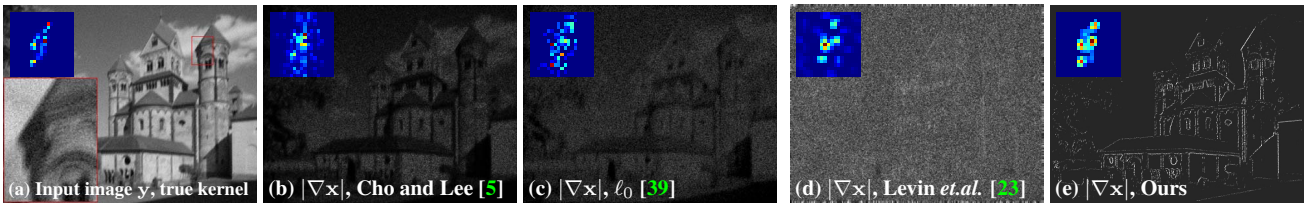


Figure 2. Intermediate results from a noisy input. For visualization, we define $|\nabla \mathbf{x}| = |\nabla_h \mathbf{x}| + |\nabla_v \mathbf{x}|$ as the summation of the absolute of gradients on horizontal and vertical directions. All kernels are zoomed and visualized using pseudo-color. (a) Synthetic input image with 5% noise and ground truth kernel. A local part is zoomed to show noise. (b)-(e) Intermediate gradients by different methods.

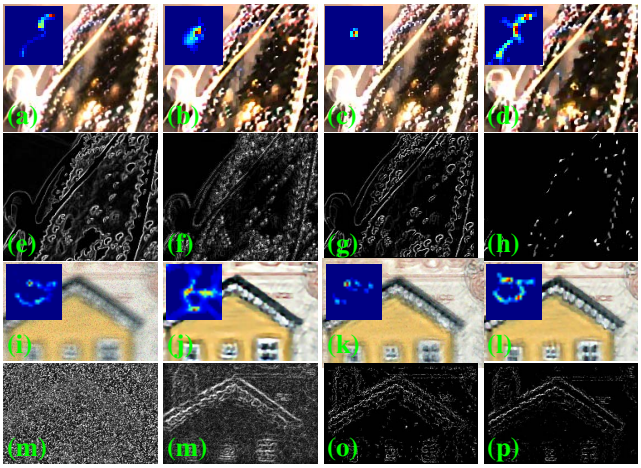


Figure 3. Examples of handling saturation and impulse noise. (a)-(h) An example with saturation. (i)-(p) An example with impulse noise. Columns from left to right: Blurry image and ground truth kernel, results of Pan *et.al.* [26] ((b) and (f)) or Zhong *et.al.* [44] ((j) and (m)), Our results without outlier discarding, our results with outlier discarding. (f)-(h) and (m)-(p) are estimated intermediate gradients. (e) and (m) are gradients of blurry image.

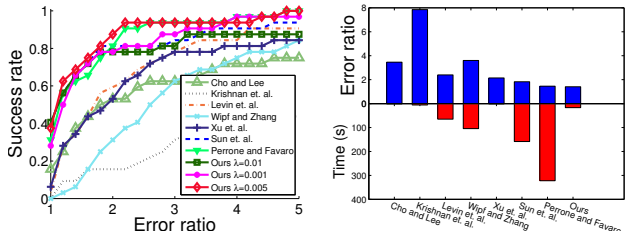
The effectiveness of the above strategy can be found in Figure 3, which shows that the pruning of the high-intensity pixels is essential for handling outliers.

5. Experiments

We evaluate the performance of our method on both synthetic data and real-world images, and compare with several state-of-the-art image deblurring methods. We implement our method in Matlab and evaluate the efficiency on an Intel Core i7 CPU with 8GB of RAM.

5.1. Evaluation on Synthetic Data

Noise-free images. We investigate a benchmark dataset of Levin *et.al.* [22] which contains 32 blurry images of size 255×255 generated by blurring 4 images using 8 ground-truth blur kernels. To evaluate the accuracy of estimated blur kernels, we recover the sharp image with the same non-blind deconvolution method in [21], and then measure the sum-of-squared-difference (SSD) error between the recovered image and the ground truth image.



(a) (b)

Figure 4. Comparison on the noise free synthetic data in [22]. (a) cumulative curves of SSD error ratio. (b) Mean SSD error ratio and mean running time (seconds).

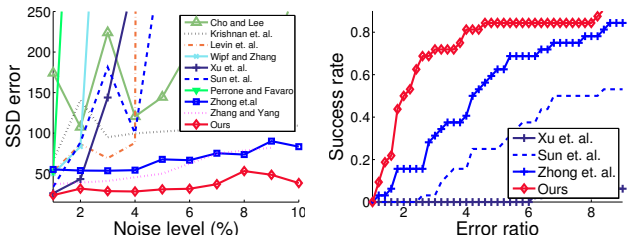
We evaluate our method and compare with other methods from Cho and Lee [5], Krishnan *et.al.* [19], Levin *et.al.* [23], Wipf and Zhang [36], Xu *et.al.* [39], Sun *et.al.* [30], Perrone and Favaro [28]. Figure 4(a) records the cumulative curves of SSD error ratio [22] which show that our method is comparable with the state-of-art log-prior based method [28], and outperforms the other methods. In this experiment, we study three values to λ (i.e. $\lambda = 0.001, 0.005, 0.01$) for our method. We can see that the proposed method with $\lambda = 0.005$ consistently performs better than others. A proper λ can help the proposed method handle the noise and estimation error when solving (13). Even so, the performances of the other two parameter settings are still on par with other state-of-the-art methods, which suggests that our method is relative insensitive to the regularization parameter, and a small λ is enough for generating high quality blur kernels.

Figure 4(b) shows the mean SSD error rate and the running time of competing methods, which suggests that our method can estimate an accurate kernel in a little time. The methods of Cho and Lee [5] and Xu *et.al.* [39] are implemented in C/C++, thus they are much faster than others.

Noisy images. To validate the noise robustness of the proposed method, we add i.i.d. Gaussian noise with zero mean to Levin *et.al.* 's data [22], and then perform kernel estimation on the noisy images. The estimated kernels are used to recover the noise-free blurry images via the non-blind deconvolution [21]. In this way, the evaluation for the kernel

Table 1. Comparison on Köhler *et.al.* 's dataset [15].

	Image 1		Image 2		Image 3		Image 4		Total Avg.	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Whyte *et.al.* [35]	27.5475	0.7359	22.8696	0.6796	28.6112	0.7484	24.7065	0.6982	25.9337	0.7155
Krishnan *et.al.* [10]	26.8654	0.7632	21.7551	0.7044	26.6443	0.7768	22.8701	0.6820	24.5337	0.7337
Cho and Lee [5]	28.9093	0.8156	24.2727	0.8008	29.1973	0.8067	26.6064	0.8117	27.2464	0.8087
Xu and Jia [38]	29.4054	0.8207	25.4793	0.8045	29.3040	0.8162	26.7601	0.7967	27.7372	0.8095
Yue *et.al.* [40]	30.1340	**0.8819**	25.4749	0.8439	30.1777	0.8740	26.7661	0.8117	28.1158	0.8484
Ours	**30.3572**	0.8611	**25.5210**	**0.8483**	**31.6577**	**0.9268**	**27.4804**	**0.8807**	**28.7541**	**0.8768**



(a) (b)

Figure 5. Comparison on noisy synthetic data. (a) SSD error with increasing noise level. (b) Cumulative curves of SSD error ratio.

estimation ability will not be influenced by the performance of the non-blind deconvolution on noisy data [5, 42]. Two methods with specific noise handling strategies [44, 42] are also evaluated in this experiment.

We first evaluate the different methods with increasing noise levels from 1% to 10%, and record the SSD error values of different methods in Figure 5(a) [3]. From this figure, all compared methods perform well with low noise levels, but the performances of most methods degrades dramatically when noise becomes relatively severe. However, the method of Zhong *et.al.* [44], Zhang and Yang [42] and our method still work robustly against the increase of noise. Note that due to the absence of source code, the results of Zhang and Yang is drawn from their paper [42]. Since their experimental setting may be different from ours, the comparison with this method is only for reference. We also show the cumulative curves of SSD error ratio on the 32 blurry image with 5% Gaussian noise in Figure 5(b). From the above comparison, only the results of several top methods are recorded in Figure 5(b) for visualization.

5.2. Evaluation on Real-world Images

Dataset from Köhler *et.al.* [15]. We also test our algorithm on a real-world dataset [15] that contains 48 (800×800) real blurry images from 4 sharp images and 12 blur kernels. The ground truth images are recorded in this dataset. Table 1 shows the PSNR and SSIM [33] based qualitative comparison, which suggests that the proposed method achieves the better performance than other state-of-art methods.

We further evaluate the deblurring performance on several real-world images. As the ground truth images of real-world blurry images are unavailable, we only evaluate the visual performances. For fairness, the recovered images on each testing image are generated using the same non-blind deconvolution methods [21, 44, 34].

Images with mild noise. Blurry images photographed in normal condition usually contain mild noise, which can be handled by many methods [31]. For the images with mild noise, we compare the proposed method with the methods studied in 5.1. In Figure 6, an example [38] with a large blur kernel is shown. The estimated blur kernels of most previous methods are damaged or noisy, thus the recovered images suffer from server ring artifacts, partially because the blur kernel is large. The results of Xu *et.al.* [39] and Sun *et.al.* [30] suffer less from ringing than others, although our method preforms even better, benefiting from the intact and noiseless kernel.

Images with high levels of noise. Most state-of-art methods are less effective in processing blurry images with high levels of noise [44]. As is shown on synthetic data, the proposed method can handle noise well. Figure 7 (a) shows a real blurry image containing severe noise from [31]. We compare our method with several state-of-art methods [31, 44, 42] which seek to handle noise in deblurring. As is shown in Figure 7, our deblurred image contains more details but no strong ringing artifacts due to the accurate kernel estimation.

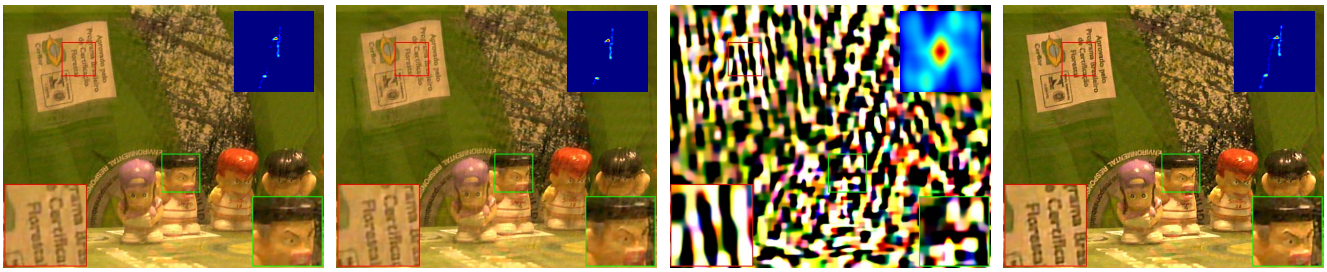
Images with saturation. As discussed in 4.2, saturated regions interfere with the kernel estimation. We compare with the previous state-of-art methods which achieve good performances on images with saturation [26, 12]. As shown in Figure 8, the method of Hu *et.al.* [12] and ours reveal more details benefiting from the accurate blur kernel. Our method is slightly better than that of Hu *et.al.* [12] (See the zoomed area.).

6. Conclusion

In this paper, we propose an effective and robust method for blind image deconvolution. The proposed method is based on an observation that only a subset of the gradients of latent image is sufficient for kernel estimation. We have

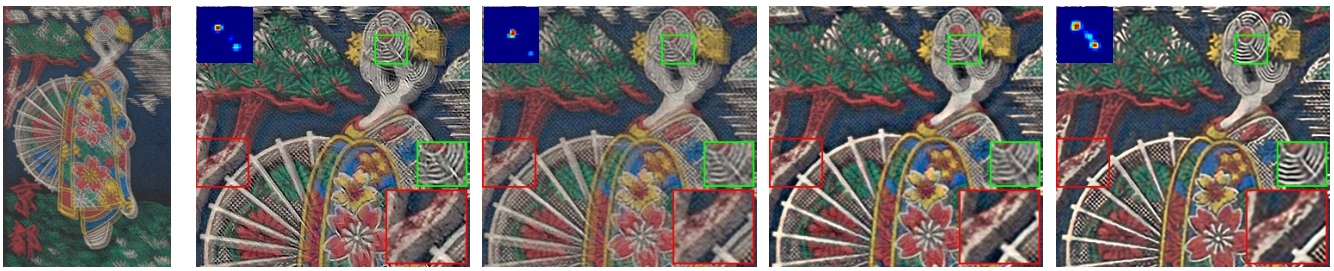
[3] Images with $1\% - 10\%$ noise are generated by adding Gaussian noise with zero mean and standard deviations from 0.01 to 0.1 respectively for images with $[0, 1]$ intensity range [44].

(a) Input (b) Cho and Lee [5] (c) Krishnan *et.al.* [19] (d) Levin *et.al.* [23]



(e) Xu *et.al.* [39] (f) Sun *et.al.* [30] (g) Perrone and Favaro [28] (h) Ours

Figure 6. Deblurring results on real-world image `toy` where he blur kernel size is 51×51. Our results are sharper and have less ringing artifacts.



(a) Input (b) Tai and Lin [31] (c) Zhong *et.al.* [44] (d) Zhang and Yang [42] (e) Ours

Figure 7. Deblurring results on a noise contaminated real-world image `kyoto` where the blur kernel size is 35×35. Our method can recover sharp image with more details and less artifacts than others. Note that the kernel of [42] is unavailable.

(a) Input (b) Pan *et.al.* [26] (c) Hu *et.al.* [12] (d) Ours

Figure 8. Deblurring results on a low-light image with saturation where the blur kernel size is 45×45. Ours has more details.

introduced a gradient activation vector, and proposed a new model for kernel estimation based only on activated gradients. Thanks to the gradient activation vector, we developed a cutting-plane method which incrementally activate the appropriate gradients for the blur kernel estimation. Our method is robust to noise and outliers, and significantly improves the quality of blur kernel estimation. Experimental results on both synthetic and real-world data show the excellence of the proposed method.

References

[1] S. D. Babacan, R. Molina, M. N. Do, and A. K. Katsaggelos. Bayesian blind deconvolution with general sparse image priors. In *ECCV*, pages 341–355. 2012. 2

[2] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 4

[3] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Blind motion deblurring from a single image using sparse approximation. In *CVPR*, pages 104–111, 2009. 2

[4] T. F. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Trans. Image Processing*, pages 370–375, 1998. 1, 2, 3

[5] S. Cho and S. Lee. Fast motion deblurring. *SIGGRAPH ASIA*, 2009. 1, 2, 6, 7, 8

[6] S. Cho, J. Wang, and S. Lee. Handling outliers in non-blind image deconvolution. In *ICCV*, pages 495–502, 2011. 5

[7] 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*. 2015. 4

[8] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008. 4

[9] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *ACM Transactions on Graphics*, 2006. 1, 2, 3

[10] A. Goldstein and R. Fattal. Blur-kernel estimation from spectral irregularities. In *ECCV*, pages 622–635, 2012. 2, 7

[11] Y. Hacohen, E. Shechtman, and D. Lischinski. Deblurring by example using dense correspondence. In *ICCV*, pages 2384–2391, 2013. 2

[12] Z. Hu, S. Cho, J. Wang, and M.-H. Yang. Deblurring low-light images with light streaks. In *CVPR*, pages 3382–3389, 2014. 2, 5, 7, 8

[13] Z. Hu and M.-H. Yang. Good regions to deblur. In *ECCV*, pages 59–72, 2012. 1

[14] N. Joshi, R. Szeliski, and D. J. Kriegman. Psf estimation using sharp edge prediction. In *CVPR*, pages 1–8, 2008. 2

[15] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *ECCV*. 2012. 1, 7

[16] N. Komodakis and N. Paragios. Mrf-based blind image deconvolution. In *ACCV*, pages 361–374. 2013. 2

[17] D. Krishnan, J. Bruna, and R. Fergus. Blind deconvolution with re-weighted sparsity promotion. *CoRR*, 2013. 1, 2

[18] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, pages 1033–1041, 2009. 1, 4

[19] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, pages 233–240, 2011. 1, 2, 3, 6, 8

[20] W.-S. Lai, J.-J. Ding, Y.-Y. Lin, and Y.-Y. Chuang. Blur kernel estimation using normalized color-line priors. In *CVPR*, pages 64–72, 2015. 2

[21] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM Transactions on Graphics*, page 70, 2007. 1, 6, 7

[22] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, pages 1964–1971, 2009. 1, 2, 6

[23] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, pages 2657–2664, 2011. 1, 2, 6, 8

[24] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *ECCV*, pages 783–798. 2014. 2

[25] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring face images with exemplars. In *ECCV*, pages 47–62. 2014. 2

[26] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via l_0-regularized intensity and gradient prior. In *CVPR*, pages 2901–2908, 2014. 1, 6, 7, 8

[27] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *CVPR*, 2014. 1, 2

[28] D. Perrone and P. Favaro. A logarithmic image prior for blind deconvolution. *IJCV*, pages 1–14, 2015. 1, 6, 8

[29] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM Transactions on Graphics*, page 73, 2008. 1, 2

[30] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *ICCP*, pages 1–8, 2013. 2, 6, 7, 8

[31] Y.-W. Tai and S. Lin. Motion-aware noise filtering for deblurring of noisy and blurry images. In *CVPR*, pages 17–24, 2012. 2, 5, 7, 8

[32] M. Tan, I. W. Tsang, L. Wang, and X. Zhang. Convex matching pursuit for large-scale sparse coding and subset selection. In *AAAI*, 2012. 4, 5

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Processing*, 2004. 7

[34] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. *IJCV*, 2014. 7

[35] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International journal of computer vision*, pages 168–186, 2012. 7

[36] D. Wipf and H. Zhang. Revisiting bayesian blind deconvolution. *The Journal of Machine Learning Research*, pages 3595–3634, 2014. 1, 2, 3, 6

[37] L. Xiao, J. Gregson, F. Heide, and W. Heidrich. Stochastic blind motion deblurring. *IEEE Trans. Image Processing*, 2015. 2

[38] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, pages 157–170, 2010. 1, 2, 7

[39] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, pages 1107–1114, 2013. 1, 2, 6, 7, 8

[40] T. Yue, S. Cho, J. Wang, and Q. Dai. Hybrid image deblurring by fusing edge and power spectrum information. In *ECCV*. 2014. 7

[41] H. Zhang, D. Wipf, and Y. Zhang. Multi-image blind deblurring using a coupled adaptive sparse prior. In *CVPR*, 2013. 3

[42] H. Zhang and J. Yang. Scale adaptive blind deblurring. In *NIPS*, pages 3005–3013, 2014. 2, 5, 7, 8

[43] H. Zhang, J. Yang, Y. Zhang, N. M. Nasrabadi, and T. S. Huang. Close the loop: Joint blind image restoration and recognition with sparse representation prior. In *ICCV*, pages 770–777, 2011. 2

[44] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling noise in single image deblurring using directional filters. In *CVPR*, 2013. 2, 5, 6, 7, 8

[45] Y. Zhou and N. Komodakis. A map-estimation framework for blind deblurring using high-level edge priors. In *ECCV*, pages 142–157. 2014. 2

[46] W. Zuo, D. Ren, S. Gu, L. Lin, and L. Zhang. Discriminative learning of iteration-wise priors for blind deconvolution. In *CVPR*, pages 3232–3240, 2015. 2