IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 20, 2019, accepted December 11, 2019, date of publication December 23, 2019,
date of current version March 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961597

Robust Motion Blur Kernel Estimation by Kernel Continuity Prior

XUELING CHEN[1], YU ZHU[2], JINQIU SUN[1], AND YANNING ZHANG[1], (Member, IEEE)
[1]School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
[2]School of Astronomy, Northwestern Polytechnical University, Xi'an 710072, China

Corresponding author: Yu Zhu (yuzhu@nwpu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61901384 and Grant 61871328.

ABSTRACT The accurate kernel estimation is key to the blind motion deblurring. Many previous methods depend on the image regularization to recover strong edges in the observed image for kernel estimation. However, the estimated kernel will be degraded when recovered strong edges are less accurate, especially in images full of small-scale edges. Different from previous methods, we focus on the kernel regularization. Inspired by the fact that the blur kernel is highly related to the continuous camera motion trajectory during the image capturing, we propose to encourage the continuity of the kernel through a kernel prior. The proposed prior measures the continuity of each element in the kernel and generates a continuity map. By encouraging the sparsity of the map using L_0 norm, discontinuous kernel elements are suppressed. Since the model with the proposed prior is non-convex and non-linear, an approximation method is proposed to minimize the cost function efficiently. Numerous experimental results show that our method outperforms state-of-the-art methods on both the normal and challenging cases. Moreover, the proposed prior is able to further improve the performance of existing MAP-based methods.

INDEX TERMS Image deblurring, robust blur Kernel estimation, small-scale edges, Kernel continuity.

I. INTRODUCTION

Motion blur is an image degradation caused by the motion between the camera and the scene during the exposure. Blind deblurring aims to recover the latent image and the blur kernel based on the observed blurred image, which is ill-posed. Consequently, extra information is required to alleviate this ill-posed problem. The maximum a posterior (MAP) framework [1]–[5] is commonly used in conventional deblurring methods. This framework introduces extra information by the way of priors:

$$\min_{\mathbf{x},\mathbf{k}} \|\mathbf{y} - \mathbf{k} * \mathbf{x}\|_2^2 + \lambda\phi(\mathbf{x}) + \rho(\mathbf{k}) \tag{1}$$

where $\mathbf{y}$, $\mathbf{x}$ and $\mathbf{k}$ denote the blurred image, the latent sharp image and the blur kernel, respectively. $\|\mathbf{y} - \mathbf{k} * \mathbf{x}\|_2^2$ is the likelihood term that enforces the similarity between the blurred image $\mathbf{y}$ and the latent image degradation $\mathbf{k} * \mathbf{x}$. $\phi(\mathbf{x})$ is the image prior and $\rho(\mathbf{k})$ is the blur kernel prior.

Previous methods seek help from image statistics for image priors $\phi(\mathbf{x})$, such as the mixture of Gaussian [6], the normalized sparsity [7], the L_0 sparse representation [2] and the Hyper-Laplacian [8]. Their success relies on the recovery of strong edges during optimization [9], [10], since strong edges provide most of the accurate blur information. In addition to image priors, heuristical image filters are also effectively used for recovering strong edges [11]–[13]. Meanwhile, Xu et al. [1] figure out strong edges are not always good for kernel estimation because some strong but small-scale edges may introduce ambiguities. Moreover, the image prior $\phi(\mathbf{x})$ can only affect the kernel accuracy indirectly by the likelihood term.

On the other hand, the accurate description of the kernel prior is more direct and important [5], [21], [22]. In the view of that the motion blur kernel is generated by the continuous camera shake trajectory during the exposure, the kernel highly resembles the connected and continuous trajectory. Therefore, the continuity is a very important property of the kernel. To encourage the continuity, the most popular kernel priors focus on the kernel gradient domain $\nabla\mathbf{k}$, such as $\|\nabla\mathbf{k}\|_2^2$ [15], $\|\nabla\mathbf{k}\|_{0.5}$ [17] and the new spatial term [16]. The model is easily optimized with the gradient term, but it emphasizes more on the local smoothness rather than the connectivity of the support area which is the key property of the continuity. Other methods use the post-processing step to

The associate editor coordinating the review of this manuscript and approving it for publication was Hongjun Su.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

TABLE 1. Different methods to constrain the kernel estimation.

Related Work	Method	Category	Continuity Preservation
[14], [15]	$\Vert\mathbf{k}\Vert_0$	prior	No
[16]	$C(k) = \#\{\mathbf{x} \mid \lvert\nabla_x \mathbf{k}\rvert + \lvert\nabla_y \mathbf{k}\rvert \neq 0\}$	prior	Yes
[17]	$\Vert\mathbf{k}\Vert_{0.9} + \Vert\nabla\mathbf{k}\Vert_{0.5}$	prior	Yes
[7], [18]	$\Vert\mathbf{k}\Vert_1$	prior	No
[19]	$\Vert\mathbf{k}\Vert_{0.5}$	prior	No
[15]	$\Vert\mathbf{k}\Vert_0 + \Vert\nabla\mathbf{k}\Vert_2^2$	prior	Yes
[2]–[4], [20]	$\Vert\mathbf{k}\Vert_2^2$	prior	No
[3], [4]	noise pruning	post-process	Yes

enforce the kernel continuity, i.e.the adaptive threshold [1] and the noise pruning [3], [4]. But these two-step methods lack of a unified cost function, leading to the difficulties on the convergency and the global optimization. Therefore, a unified model with a proper continuity preservation is required in the kernel estimation problem.

Inspired by this, we propose a new continuity measurement for the kernel prior in a unified model. First, we define the continuity on each element within a window to form a map. Second, by enforcing the map to be sparse using the L_0 norm, less connected elements, e.g.isolated noise, are penalized. Thus the problem of the encouraging the kernel continuity becomes the problem of encouraging the map sparsity. Finally a unified model with the kernel continuity prior is proposed. As the kernel prior is non-linear and non-convex, an approximate solver is proposed to optimize the unified model efficiently.

The contributions of the paper are summarized as follows:

- We are the first to propose a kernel prior that measures the continuity of each element in a window and preserves the kernel continuity by encouraging the sparsity in a unified model.
- An approximate solver is proposed to minimize the non-convex and non-linear cost function and achieves fast convergence.
- Solid experimental results on large datasets and challenging cases show our method outperforms previous methods in both accuracy and efficiency. Moreover, the proposed prior can further improve the performance of the state-of-the-art method.

The rest of the paper is organized as follows: Section II reviews the related work. Section III describes the proposed continuity kernel prior. In section IV, the unified model and the optimization are presented. Section V analyzes the effectiveness of the proposed kernel prior. Section VI shows the experimental results. In sectionVII, we draw the conclusion of our method.

II. RELATED WORK

Recent years have witnessed the fast development of the kernel estimation in the blind image deblurring. We review the related work in three aspects: the image priors, the kernel priors and the post-processing methods.

A. IMAGE PRIORS

Most previous methods focus on the image priors [1]–[5], [7], [8], [11], [16], [23]–[29]. The success of these methods relies on the recovery of strong edges for the kernel estimation. However, strong edges do not always benefit the kernel estimation, for example, strong but small-scale edges even introduce ambiguities [1]. Here small-scale edges indicate the closely adjacent strong edges whose gap is smaller than the kernel size. Under the blur degradation, the high frequency details are easily confused. Xu et al. [1] point out that the gradient magnitude of the small-scale edge in the blurred image is much lower than the original sharp one, making it hard to recover the original sharp edges. They propose a criterion to select larger-scale edges in the image for kernel estimation to avoid the negative effect of small-scale edges. The criterion is effective and also used by method [16]. However, these edge selection methods are based on the assumption that large-scale edges are available in the blurred image, which may be violated in some cases. Moreover, the image prior can only affect the kernel accuracy indirectly by the likelihood term.

B. KERNEL PRIORS

Other methods study the characteristics of the blur kernel and constrain the estimated kernel using priors, as shown in Table 1. Among them, L_2 norm [2]–[4], [20], L_0 norm [14], [15], L_1 norm [7], [18], $L_{0.5}$ norm [19] and $L_{0.9}$ norm [17] are widely used as blur kernel priors $p(\mathbf{k})$. However, most of them do not take the kernel continuity into consideration. To encourage the continuity, some researchers propose priors with kernel gradients. Reference [17] proposes a prior that combines the kernel intensity and gradient: $\|\mathbf{k}\|_{0.9} + \|\nabla\mathbf{k}\|_{0.5}$. Similarly, [15] proposes another combination: $\|\mathbf{k}\|_0 + \|\nabla\mathbf{k}\|_2^2$. Reference [16] borrows the idea of L_0 image smoothing method [30] to constrain the continuity in kernel gradient domain. The idea of restricting the kernel gradient has a weakness that it emphasize more on the local smoothness rather than the kernel continuity.

C. KERNEL POST-PROCESSING METHODS

In addition to the kernel prior, post-processing methods [3], [4], [31], [32] are also widely used to refine the estimated kernel, as shown in Table 1. A threshold is the simplest way to denoise the kernel after the kernel estimation. However, it is not efficient to remove noise that has higher intensity than the main trajectory. Some methods also remove noise using space consistency [31] or spareness evaluation [32], but these methods are for certain cases. [3], [4] use the noise pruning method to preserve and remove small non-zero regions after the kernel estimation. However, when those regions are connected to the main trajectory of the kernel, they can not be identified and removed by the noise pruning method. In a word, the post-process method can only refine the kernel, but the accuracy of the kernel mainly depends on the kernel estimation performance. Moreover, these two-step methods lack of a unified cost function, leading to the difficulties on the convergence and the global optimization.

In this paper, we propose a new kernel prior to preserve the kernel continuity. Different from the previous priors that preserve the kernel continuity in the gradient domain, we define the continuity of each element within a window and form a continuity map. By encouraging the map to be sparse, in a unified model, the unconnected kernel elements are suppressed.

III. THE KERNEL CONTINUITY PRIOR

In this section, we give details of the kernel continuity prior. A blur kernel $\mathbf{k} \in \mathbb{R}^{f \times f}$ describes the motion trajectory of a camera during the exposure time, in which non-zero elements denote positions that the camera goes through. Usually, the kernel element with higher intensity denotes a longer integration at the corresponding position. In a continuous period of exposure time, the camera trajectory is continuous. Consequently, non-zero elements of the kernel should be connected, which form the main structure related to the camera trajectory, as shown in Figure 1 (a). Therefore, we get two requirements of a possible kernel trajectory element:

- The element intensity should be relatively high, indicating the photon integration during the exposure.
- The element should have connected neighbors with similar or higher intensities.

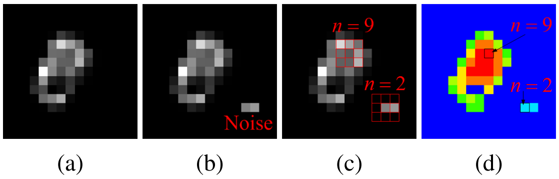


FIGURE 1. The kernel continuity map generation steps. (a): a ground truth kernel in Levin dataset [33]. (b): the kernel with noise. (c): measuring the continuity of each kernel element by counting elements that meet the two requirements presented in Section III. (d): the final continuity map in pseudo color, in which the more red color denotes the higher value.

Considering the above requirements, we measure the possibility of each kernel element and form a kernel continuity map.

First, we judge whether the intensity of each kernel element is high enough using a global threshold T_g. We set $T_g = a \cdot \mathbf{k}_{max}$, where $0 < a < 1$ and $\mathbf{k}_{max}$ is the highest intensity of the kernel $\mathbf{k}$. If the intensity of the current element is higher than T_g, the element meets the first requirement. Here, a lower T_g is necessary for involving all the possible trajectory elements for evaluation. As a result, we set $a = 0.05$ which is used as an empirical threshold for removing smaller values in an estimated kernel in previous methods [3]–[5].

Algorithm 1 The Kernel Continuity Map Generation

```
Input: the blur kernel $\mathbf{k} \in \mathbb{R}^{f \times f}$, the global threshold $T_g$,
    the local threshold $T_l$, the window size $r$
Output: continuity map $M(\mathbf{k})$
Initialize: the map $M$
for $h = 1{:}f$ do
    for $v = 1{:}f$ do
        if $\mathbf{k}(h, v) > T_g$ then
            select the $r \times r$ patch centering at $(h, v)$.
            counting the number $n$ of elements that are
            larger than $T_l$ in the patch.
            $M(h, v) = n$
        end if
    end for
end for
```

Second, we use a sliding window whose size is $r \times r$ to extract connected neighbors of each kernel element. A threshold $T_l = b \cdot \mathbf{k}_c$ is used to evaluate all the neighbors in the window, where $0 < b < 1$ and $\mathbf{k}_c$ is the intensity of the center element in the window. Only neighbors with higher intensities than T_l will be counted. The value of T_l directly affects the continuity measurement of the kernel element, a relatively higher T_l is necessary to leave out noise elements. So we set $b = 0.1$. The window size is set as $r = 3$ to ensure all the elements in the window are next to the center element. We denote the number n of satisfying elements within the window of each kernel element as the continuity metric. Larger n means the more connected neighbours and the higher possibility of the element belonging to the trajectory. In this way, we judge whether each kernel element meets the second requirement.

After all the elements of the kernel is measured, a map $M(\mathbf{k}) \in \mathbb{R}^{f \times f}$ will be formed. The steps of the map generation are shown in Figure 1 (c)-(d) and the detailed implementation is shown in Algorithm 1. Figure 1 (b) shows a kernel with noise. Noise is inevitable during the optimization and it degrades the continuity of the kernel. In the proposed continuity map in (d), kernel noise is imposed with lower values, even though the noise has higher intensity. It is the main difference from threshold-based methods.

Figure 2 shows comparisons between the intensity maps and the continuity maps of kernels with and without noise. Obviously, all noise elements with higher or lower intensities are imposed with lower values in the continuity map. Thus, using the continuity map, we can easily distinguish noise.

To encourage the kernel continuity during the optimization, we encourage the continuity map to be sparse using L_0 norm. The proposed kernel prior is $\|M(\mathbf{k})\|_0$. However,

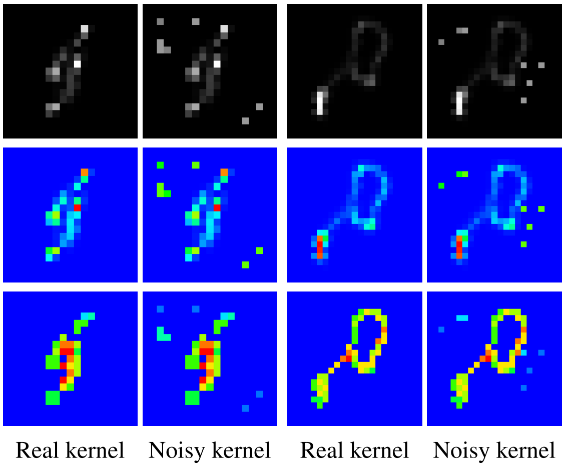


FIGURE 2. Comparison between kernel continuity map and intensity map. First row: real kernels and corresponding kernels with manually added noise. Second row: corresponding intensity maps in pseudo color. Third row: corresponding continuity maps in pseudo color. This figure shows the main trajectories are more separable from noise in the continuity maps than that in the intensity maps.

the non-convex and non-linear of the prior makes it hard to optimize a cost function with the prior. Next section will give detailed solver of it.

IV. FRAMEWORK AND OPTIMIZATION

In this section, the details of the kernel estimation framework is presented and we also give the approximate solver of the non-convex and non-linear cost function.

As mentioned in Section III, we use the L_0 norm to encourage the sparsity of the continuity map $M(\mathbf{k})$. The cost function with the proposed prior is:

$$\min_{\mathbf{x},\mathbf{k}} \|\mathbf{y}-\mathbf{k}*\mathbf{x}\|_2^2+\lambda\phi(\mathbf{x})+\gamma\|\mathbf{k}\|_2+\alpha\|M(\mathbf{k})\|_0 \tag{2}$$

where λ, γ and α are the weights to balance the terms. To minimize (2), we alternatively update the blur kernel $\mathbf{k}$ while fixing $\mathbf{x}$:

$$\min_{\mathbf{k}} \|\mathbf{y}-\mathbf{k}*\mathbf{x}\|_2^2+\gamma\|\mathbf{k}\|_2+\alpha\|M(\mathbf{k})\|_0 \tag{3}$$

and the image $\mathbf{x}$ while fixing $\mathbf{k}$:

$$\min_{\mathbf{x}} \|\mathbf{y}-\mathbf{k}*\mathbf{x}\|_2^2+\lambda\phi(\mathbf{x}) \tag{4}$$

During the alternative optimization, the updated $\mathbf{x}$ and the updated $\mathbf{k}$ in each iteration are denoted as intermediate image and intermediate kernel, respectively. The flowchart of our framework is presented in Figure 3. After the output kernel is estimated, the final deblurred image is obtained by the deconvolution of the input image with the output kernel.

A. ESTIMATING THE INTERMEDIATE BLUR KERNEL

Estimating $\mathbf{k}$ using (3) is not easy due to the combination of the non-linear function $M(\cdot)$ and the non-convex function L_0. We use the half quadratic splitting [1] to solve the L_0 regularized term by introducing an auxiliary variables $\mathbf{u}$ with respect to $M(\mathbf{k})$. We rewrite the cost function as:

Algorithm 2 Kernel Estimation

```
1: Input:blurred image $\mathbf{y}$, latent image $\mathbf{x}$
2: Initilize: initial kernel $\mathbf{k}_{prev}$
3: Output: estimated kernel $\mathbf{k}$
4: $M^{\mathbf{T}}\mathbf{u}=\mathbf{k}_{prev}$
5: for $j$ = 1:$n_{\mathbf{k}}$ do
6:     solve for $\mathbf{k}$ by Equation (9)
7:     solve for $\mathbf{u}$ by Equation (8)
8:     $\beta=\beta\times 2$
9: end for
```

$$\min_{\mathbf{k}} \|\mathbf{y}-\mathbf{k}*\mathbf{x}\|_2^2+\gamma\|\mathbf{k}\|_2+\beta\|M(\mathbf{k})-\mathbf{u}\|_2+\alpha\|\mathbf{u}\|_0 \tag{5}$$

where β is the penalty parameter. According to [1], when β is close to infinity, the solution of (5) is close to the solution of (3). Similar to [1]–[3], we update the value of β by $\beta=\beta\times 2$ in each iteration as shown in Algorithm 2. We minimize (5) by alternatively solving the following two sub-problems: estimating $\mathbf{k}$ while fixing $\mathbf{u}$

$$\min_{\mathbf{k}} \|\mathbf{y}-\mathbf{k}*\mathbf{x}\|_2^2+\gamma\|\mathbf{k}\|_2+\beta\|M(\mathbf{k})-\mathbf{u}\|_2 \tag{6}$$

and estimating $\mathbf{u}$ while fixing $\mathbf{k}$

$$\min_{\mathbf{u}} \beta\|M(\mathbf{k})-\mathbf{u}\|_2+\alpha\|\mathbf{u}\|_0 \tag{7}$$

The details of the intermediate kernel estimation sub-model is shown in Algorithm 2. Solving (7) is relatively easy, because when $\mathbf{k}$ is fixed we can directly compute $M(\mathbf{k})$, and then estimating $\mathbf{u}$ becomes an element-wise minimization problem [1] that can be solved as:

$$\mathbf{u}=\begin{cases} M(\mathbf{k}), & |M(\mathbf{k})|^2\geq\frac{\alpha}{\beta}\\ 0, & \textbf{otherwise}\end{cases} \tag{8}$$

The detailed implementation is shown in Figure 4. First, we transfer a noisy kernel to a continuity map $\mathbf{u}$ using $M(\cdot)$. Second, we use (8) to update $\mathbf{u}$, consequently, noise is identified in the map $\mathbf{u}$. Third, we transfer $\mathbf{u}$ back to a blur kernel $\mathbf{k}=M^{\mathbf{T}}(\mathbf{u})$ by setting noise to be zero in $\mathbf{k}$.

Equation (6) is a standard quadratic function that can be solved using FFT method [3]:

$$\mathbf{k}=\mathcal{F}^{-1}\left(\frac{\bar{\mathcal{F}}(\mathbf{x})\cdot\mathcal{F}(\mathbf{y})+\beta\mathcal{F}(M^{\mathbf{T}}\mathbf{u})}{\bar{\mathcal{F}}(\mathbf{x})\cdot\mathcal{F}(\mathbf{x})+\alpha+\beta}\right) \tag{9}$$

where $\bar{\mathcal{F}}(\mathbf{x})$ is the conjugate of $\mathcal{F}(\mathbf{x})$.

B. ESTIMATING THE INTERMEDIATE IMAGE

Given $\mathbf{k}$, we estimate $\mathbf{x}$ by minimizing (4). The chosen of the image prior $\phi(\mathbf{x})$ heavily affects the performance of the intermediate image estimation. Previous methods have used the image gradient prior by representing the image gradient distribution using L_0 norm [2], $L_{0.8}$ norm [5], L_1 norm, L_1/L_2 norm [7] and also image intensity prior with L_0

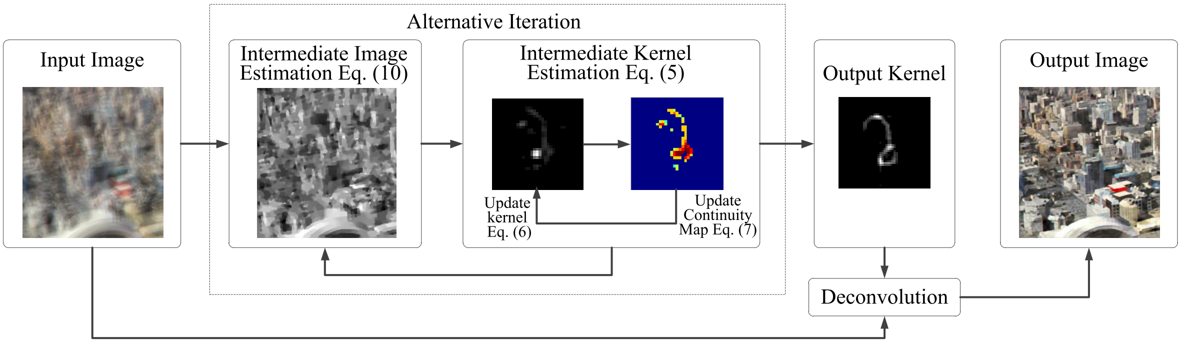

Alternative Iteration
Input Image
Intermediate Image Estimation Eq. (10)
Intermediate Kernel Estimation Eq. (5)
Update kernel Eq. (6)
Update Continuity Map Eq. (7)
Output Kernel
Output Image
Deconvolution

FIGURE 3. Our framework. Estimating a blur kernel includes two parts: the intermediate image estimation and the intermediate kernel estimation. The two parts are implemented alternatively. The output deblurred image is the deconvolution of the output kernel and the input image.

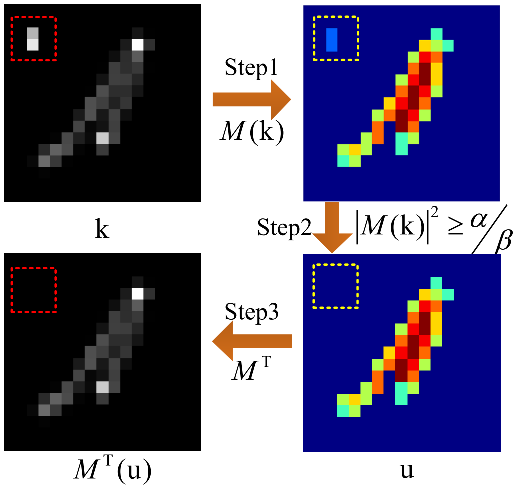

Step1
$M(\mathrm{k})$
k
Step2
$|M(\mathrm{k})|^2 \geq \alpha/\beta$
Step3
M^{T}
$M^{\mathrm{T}}(\mathrm{u})$
u

FIGURE 4. Steps to encourage the sparsity of the continuity map u. Step 1: transfer the kernel to the continuity map. Step 2: update u in Equation 8. Step 3: transfer the continuity map back to the kernel.

norm [3]. With rapid progress of the deep neural networks, some methods [24], [34] replace the above handcrafted priors with data-driven priors that are learnt from collected training datasets using neural networks. Data-driven priors do not make assumptions of the distribution of the latent image as what handcrafted priors do. But training dataset may introduce bias to the data-driven prior.

For simplicity and efficiency, we choose $\phi(\mathbf{x}) = \|\nabla \mathbf{x}\|_0$ [2], [3], [4] for our framework. The cost function of the intermediate image is:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{k} * \mathbf{x}\|_2^2 + \lambda \|\nabla \mathbf{x}\|_0 \tag{10}$$

Here, we use half quadratic splitting (HQS) [1] to solve (10).

C. ESTIMATING THE FINAL LATENT IMAGE

After estimating the blur kernel using Algorithm 2, we recover the final latent image using non-blind deconvolution method. Different from the intermediate image in Section IV-B that is used for kernel estimation, the latent image is the final deblurred image with fine textures. As this paper focuses on kernel estimation, we use existing non-blind deconvolution method in [4] to recover our deblurred image.

Algorithm 3 Framework of the Multi-Scale Implementation

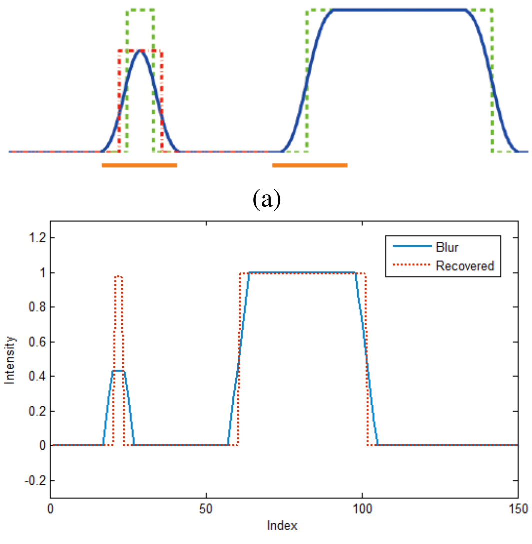
```
Input:blurred image $\mathbf{y}$, kernel size $f$, iteration number $m$
Initialize: kernel $\mathbf{k_0}$ in the coarsest scale
Output: kernel $\mathbf{k}$
Compute the scale number $s$ according to [11]
$\mathbf{k} = \mathbf{k_0}$
for $i = 1{:}s$ do
    downsample $\mathbf{y}$ to the current image pyramid to get $\mathbf{y}_i$
    for $g = 1{:}m$ do
        solve for $\mathbf{x}$ by Equation (10)
        solve for $\mathbf{k}$ by Equation (5)
    end for
    upsample $\mathbf{k}$ to fit the next scale
end for
```

D. MULTI-SCALE STRATEGY

In order to ensure the fast convergence, many previous methods use the multi-scale strategy [3], [4], [11], [16]. Because the blur could be reduced when the image is downsampled to coarse scales, moreover, the reduced blur makes it easier to estimate an accurate kernel. We also adopt the multi-scale strategy [11] to estimate the blur kernel by a coarse-to-fine pyramid of image resolutions. To get the blurred image $\mathbf{y}_i$ in each scale of the pyramid, we downsample $\mathbf{y}$ using a factor $\frac{\sqrt{2}}{2}$, which is similar to [16]. The implementation of the multi-scale strategy in our framework is shown in the Algorithm 3.

V. ANALYSIS OF THE PROPOSED KERNEL PRIOR

In this section, we analyze the effectiveness of the proposed kernel prior. First, we show the potential of an intermediate kernel to recover accurate strong edges in the intermediate image, which is key to conventional blind deblurring methods [2], [16], [20]. Second, the effect of the proposed kernel prior in improving the accuracy of the intermediate kernel



(a)

(b)

FIGURE 5. The potential of the accurate kernel. (a) The ambiguity of the blurred small-scale edge presented in [1]. According to [1], the magnitude of the latent small-scale signal (green dashed line) is lowered after being blurred (blue line). The red dashed line is more likely to be the latent sharp signal rather than the green one. (b) Our experimental result. The recovered sharp signal of small-scale can be recovered precisely if an accurate kernel is given.

is presented. Third, we compare the proposed prior with other methods preserving the kernel continuity. Last but not the least, we evaluate the convergence property of our method.

A. THE POTENTIAL OF AN INTERMEDIATE KERNEL IN RECOVERING STRONG EDGES

We revisit some previous edge selection methods [1], [16] and show that an accurate intermediate kernel is crucial for the strong edges recovery.

Similar to [1], we use an 1D signal with strong edges for illustration. Figure 5 (a) is presented in [1] to show the ambiguity of the recovered strong edges in the intermediate image. It has a small-scale edge on the left and a large-scale edge on the right. The blue curve denotes the blurred signal and the green curve denotes the ground truth signal. The magnitude of blurred small-scale edge is much lower than the ground truth. As a result, without other information it tends to recover the signal to the red line instead of the ground truth green one. Edge selection methods [1], [16] will exclude the small-scale edge in the kernel estimation. However, if the blurred image does not has large-scale edges, these methods will fail.

In this case, an accurate intermediate kernel can provide information to recover the ground truth edge. If the intermediate kernel is as accurate as the ground truth kernel $\mathbf{k}_{GT}$, we can recover a sharp signal $\hat{\mathbf{x}}$ by (10). Figure 5 (b) shows the potential of the accurate kernel. The recovered signal (red dashed curve) has much higher magnitude than the blurred one (blue curve) and is much close to the ground truth one, which means the ambiguity of the edge magnitude is well suppressed.

Even though the intermediate kernel may not be as accurate as the ground truth in practice, by improving the kernel accuracy in each iteration, small-scale edges can be accurately recovered in the intermediate image. Figure 6 (e) gives an example that our method can recover the small-scale edge better than the other methods (b)-(d) as shown in the red box.

B. ABLATION STUDY

We verify the effectiveness of the proposed kernel prior by experiments in this section.

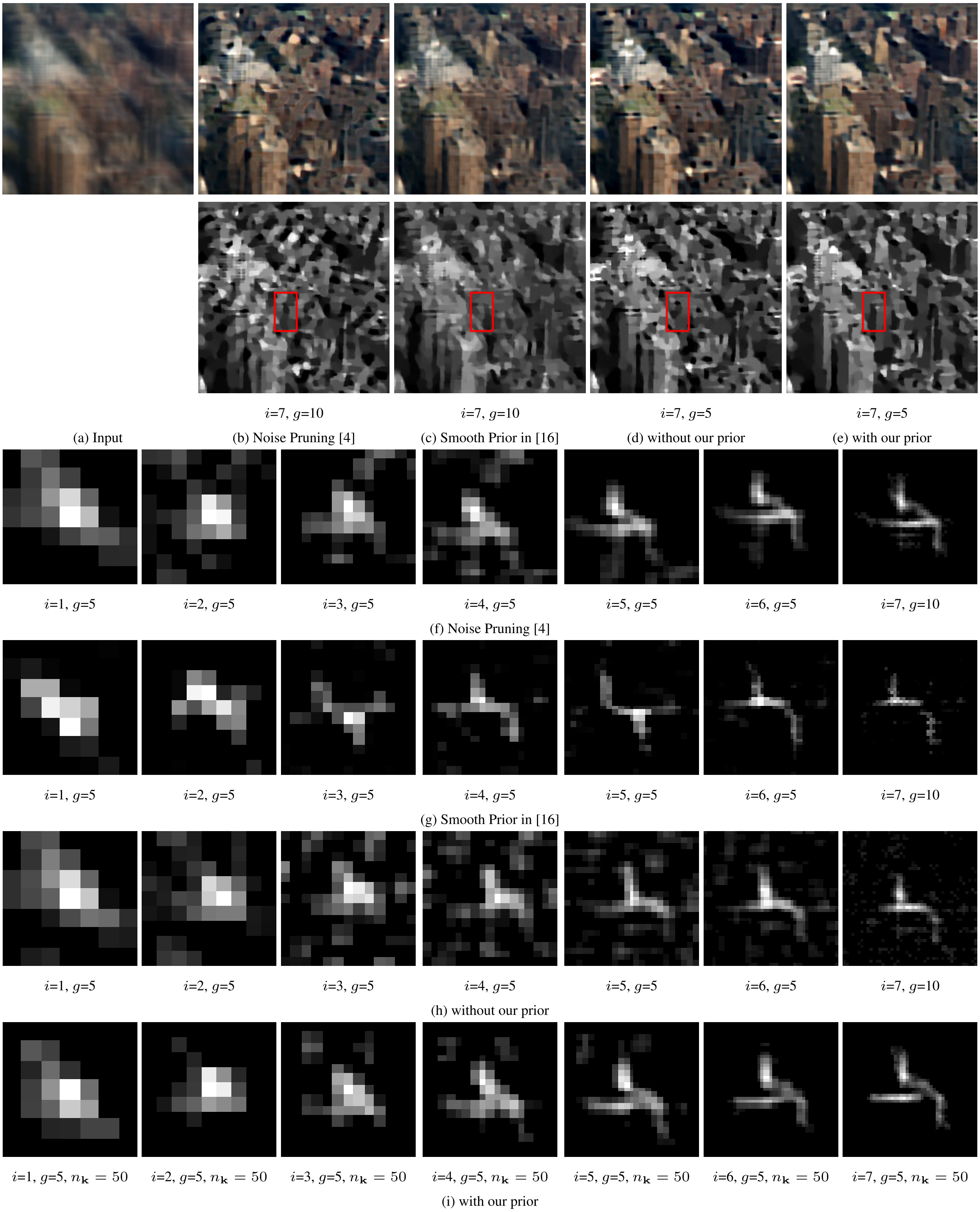
In our framework, we set $\lambda = 0.004$, $\gamma = 0.1$, $\alpha = 16$, $\beta = 1$, $n_{\mathbf{k}} = 50$, $m = 5$, $a = 0.05$, $b = 0.1$, $r = 3$ and n is computed using the multi-scale strategy in Section IV-D. We use the same non-blind deconvolution method [4] to recover the final sharp image with estimated kernels.

To compare the deblur performance with and without the proposed prior, we set $\alpha = 16$ and $\alpha = 0$ in our framework, respectively. Figure 6 gives an example: an image full of small-scale edges, in which the sizes of most edges are smaller than the blur size. Figure 6 shows the final deblurred images, intermediate images and intermediate kernels in different iterations. To ensure each algorithm achieve their best results, we use different iteration numbers in the finest scale. (d) and (e) show the deblurred images and the intermediate images with and without the proposed prior, respectively. In (d), obvious ringing artifacts spread over the whole image, while (e) recovers faithful image edges. It is because without the proposed prior, noise of the kernel is not suppressed in each iteration (as shown in (h)) and the final estimated kernel contains severe noise beyond the main trajectory of the blur kernel ($i = 1, g = 5$). Moreover, the main trajectory is less accurate than the estimated kernel using the proposed prior as shown in the last image of (i). Using the proposed prior, noise of the kernel is progressively suppressed during iterations in (i). Consequently, the quality of the deblurred image improves a lot in (e). The comparison indicates that, the proposed prior is able to assist to estimate an accurate blur kernel in the challenging case that image is full of small-scale edges.

We also conduct a quantitative evaluation with the commonly used dataset [33]. The reason for choosing this dataset is that all the blur kernels in it are from real camera shake, which is more convincing. Table 2 shows the average PSNR and SSIM of the deblurred image using various methods. Apparently, our method with the proposed prior outperforms the method without the proposed prior in both PSNR and SSIM, which means more accurate blur kernels are estimated with the proposed prior.

C. COMPARISONS WITH OTHER CONTINUITY PRESERVATION METHODS

As shown in Table 1, previous methods that preserve the continuity of the blur kernel can be divided in two categories: 1) proposing a kernel prior. 2) removing noise as a post-process method. From each category, we choose a typical method for comparison. The first method is a similar



(a) Input (b) Noise Pruning [4] (c) Smooth Prior in [16] (d) without our prior (e) with our prior

(f) Noise Pruning [4]

(g) Smooth Prior in [16]

(h) without our prior

(i) with our prior

FIGURE 6. Comparisons among different continuity preservation methods. (a) a blurred image full of small-scale edges. (b)-(e) the deblurred images (up) and the intermediate images (down) of the noise pruning [4], the smooth prior [16], ours without and with the proposed prior, respectively. All the intermediate images shown are from the last iteration of the finest scale. The intermediate image in (e) successfully recovers the small-scale edge in the red box while others do not. And the deblurred image in (e) outperforms other results. (f), (g), (h) and (i) show the intermediate kernels of each method in different iterations. i, g, $n_{\mathbf{k}}$ indicate the ith, gth and $n_{\mathbf{k}}$th iteration in Algorithm 3 and Algorithm 2. Apparently, the proposed prior suppresses noise in each iteration and recovers the main trajectory of the kernel in (i) successfully.

TABLE 2. Quantitative evaluations on levin dataset [33].

	Framework without our prior (baseline)	Noise pruning method [3], [4]	Our prior + Noise pruning [3], [4]	Smooth prior [16]	Framework with our prior
PSNR	26.41	26.49	26.56	26.24	**26.96**
SSIM	0.84	0.85	0.84	0.83	**0.86**

L_0 gradient kernel prior [16] that preserves the smoothness of the kernel to encourage the continuity. We denote it as the smooth prior in our paper. The second one is the noise pruning method [3], [4] that remove noise elements after the kernel estimation. Here our framework without the proposed prior act as the baseline model. For the L_0 gradient kernel prior, we replace the kernel regularization term in our framework with the L_0 gradient kernel prior for comparison method. For the noise pruning method, we use it as the post-process step after kernel estimation in the baseline model.

Figure 6 (b) and (c) show the deblurred images and the intermediate images of the two methods. (f) and (g) show the iterations of kernel estimates of the two methods. Obviously, the noise pruning method removes the isolated noise in the kernel background (shown in the last image of (f)), but it does not remove the noise connected to the main trajectory in each iteration. As a result, the deblurred image in (b) degrades. The smooth prior does not remove noise in the kernel (shown in the last image of (e)) and the main trajectory is thinner than others due to its smoothing effect. So the deblurred edges are still blur in (c).

For the qualitative evaluation in Table 2, the noise pruning method [4] outperforms the baseline by improving the average PSNR value from 26.41 to 26.49, but its value are still lower than the proposed prior (26.96). The result of the smooth prior is even lower than the baseline model. We also combine both the proposed prior and the noise pruning method in our framework, but the result are worse than the one without using the noise pruning method. It indicates the noise pruning method has a negative effect when used together with the proposed prior.

To further evaluate the quality of the estimated kernel, we use the error ratio [33]:

$$e = \frac{\|\hat{\mathbf{x}} - \mathbf{x}_{GT}\|_2^2}{\|\hat{\mathbf{x}}_t - \mathbf{x}_{GT}\|_2^2} \tag{11}$$

where $\mathbf{x}_{GT}$ is the ground truth sharp image and $\hat{\mathbf{x}}_t$ is the estimated image using the ground truth kernel. The error ratio describes the similarity of the deblurred image using the estimated kernel and the deblurred image using the ground truth kernel. The smaller the value is, the higher similarity is. Figure 7 plots success rates of different error ratios in the same way as [33]. For the non-blind deconvolution method, we use a simple yet effective non-blind deconvolution method [33] to recover the final deblurred image. Each number in the y-axis shows the percentage of test images whose error ratio are below a certain threshold and x-axis lists all the thresholds. It indicates that our method provides more reliable results than other methods.

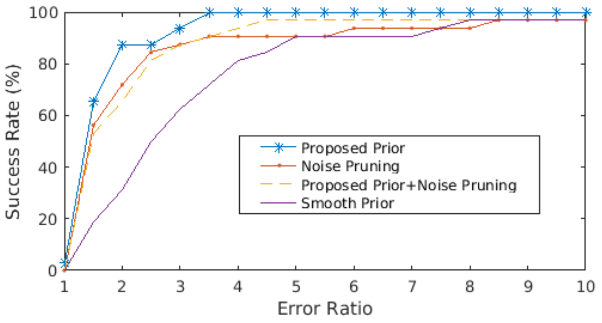


FIGURE 7. Success rates of different error ratios on Levin Dataset [33].

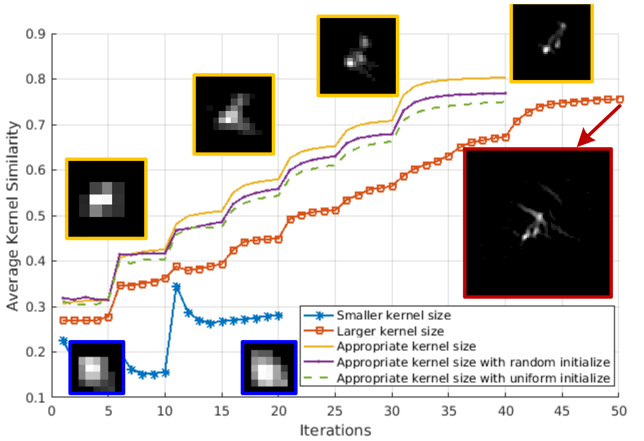


FIGURE 8. The convergence property of our method evaluated on the Levin Dataset [33]. Visual examples of intermediate kernels are shown. The ground truth kernel is shown in Figure 13 (f). A smaller kernel size needs fewer scales in the multi-scale strategy [11], leading to fewer iterations. Using the appropriate kernel size, our method converges to the good estimation. The larger kernel size leads to a biased solution, while the smaller kernel size does not converge. The convergence of our method using another two kernel initialization methods are also evaluated.

D. CONVERGENCE PROPERTY

As our energy function is non-linear and non-convex, we quantitatively evaluate convergence properties of our method on the Levin dataset [33]. We evaluate the similarity [35] between the intermediate kernel of each iteration and the ground truth kernel in Figure 8. As our method is implemented in the multi-scale manner (details in Section IV-D), we unroll iterations in all the scales and list them in sequence. Our method converges in iterations of each scale, which results in the periodic increase in curves. The convergence of different kernel sizes (appropriate, larger and smaller) are evaluated. A smaller kernel size needs fewer scales in the multi-scale strategy [11], leading to fewer iterations. The visualized examples show that a larger kernel size may converges to a biased solution, because it increases the degree of freedom of kernel estimation [20]. But a smaller size makes the method difficult to converge, as shown in the blue line of Figure 8. Consequently, choosing an appropriate kernel size is

TABLE 3. Comparison among three kernel initialization methods on Levin Dataset [33].

	Random	Uniform	Method in [7]
PSNR	24.07	26.02	**26.96**
SSIM	0.73	0.82	**0.86**

essential and there are several attempts to automatically find it [20], [36], [37].

The effect of the first estimation on the convergence property is also evaluated. Our framework initialize the kernel by [7]. We also evaluate another two initialization methods: random and uniform. Different initialization methods lead to different first estimation of both images and kernels according to (4) and (3). Our model converges using any of the three initializations as shown in Figure 8. Table 3 shows average PSNR and SSIM on the Levin Dataset [33] of the three initializations. It indicates that using [7] leads to the best deblur performance.

Besides the kernel size and the first estimation, the image type also has an effect on the convergence. For example, as recovering strong edges in each iteration is key to the success of MAP-based methods, image types lacking of strong edges make the method difficult to converge. Some failure cases are shown in Section VI-D.

VI. EXPERIMENTAL RESULTS

We compare the proposed method with state-of-the-art methods [2], [4], [7], [16], [38]. Recent years has witnessed the rapid progress of deep learning methods in the field of low-level image processing, especially in image deblurring [38], [40], [41]. However, most of them directly map the blurred image to the deblurred one without estimating a kernel, which is one of the big differences between conventional optimization-based methods and deep learning methods. We choose the recent deep learning method SRN [38] for comparison.

We test the proposed prior on some challenging cases for visual comparison. The parameter settings of our framework are the same as Section V-B. Moreover, the large deblurring dataset [27] with 640 images is used for evaluating how the proposed prior improves the previous method [4]. We also evaluate the efficiency of our method by comparing average runtime and PSNR on Köhler Dataset [42].

A. SOME CHALLENGING CASES

We conduct the experiment on three challenging cases: images full of small-scales, the noisy-blurred image and the large blur image.

1) SMALL-SCALE EDGE IMAGE

We evaluate our method on two small-scale edge images. Figure 9 shows a synthetic blurred image with small-scale grasses. Our estimated kernel in (f) is the most similar one to the ground truth kernel in (a), while other kernels are not as thin as the ground truth one. As a result, their deblurred image are degraded. The deblurred image of Xu et al. [2] in (b) contains obvious ringing artifacts around the boundary. The zoomed-in flower region of [16] in (c) is still blurred compared with ours in (f). Using the dark channel method [4] in (d), the deblurred image is over smooth as shown in the zoomed-in grass region and also there are black dots spreading over the flower region. The result of SRN [38] does not remove the blur. Unfortunately, the deep learning method does not show its priority in recovering small-scale edge image. Without the guidance of the blur kernel, SRN [38] can not accurately find the original position of image edges, which leads to the blurred result. Our method not only removes the blur accurately but also recovers natural sharp image with fine textures.

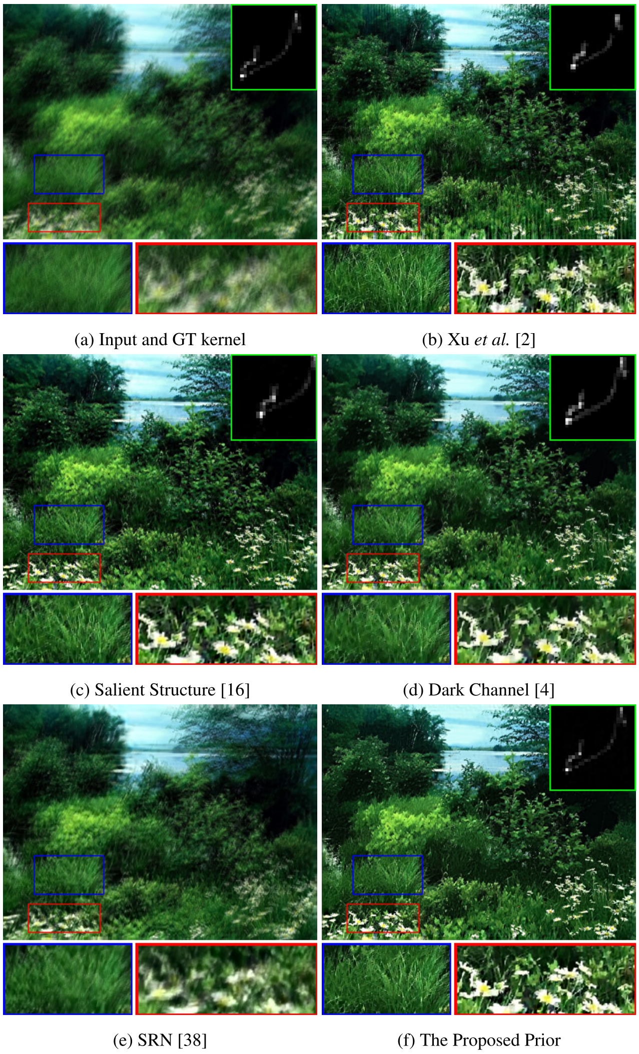


FIGURE 9. Deblurring results comparison on an image with small-scale textures. The ground truth kernel is shown in (a). (b) the blur kernel is less accurate and the deblurred image contains obvious ringing artifacts. The kernel of (c) contains isolated noise around the main trajectory, so the results contains ringing artifacts too. The kernel in (d) is not as thin as the ground truth one. The deep learning method SRN [38] does not remove blur thoroughly.

Figure 10 shows a real blurred image of building. Different from the grass in Figure 9, the edges of Figure 10 are stronger.

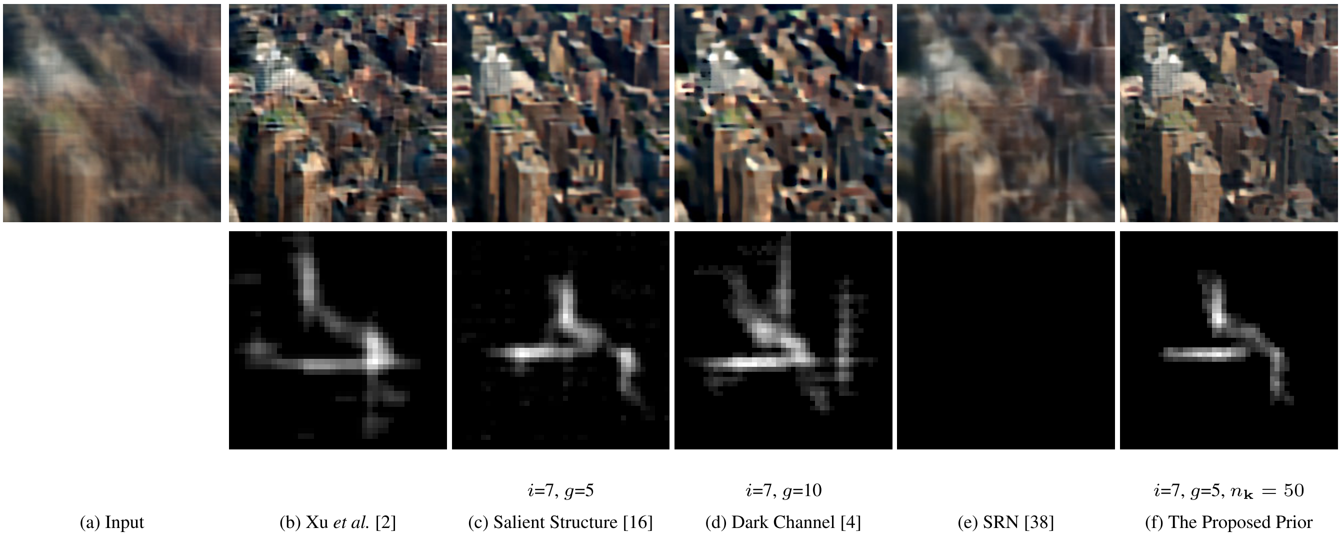


FIGURE 10. Deblurring results comparison on an image full of small-scale edges. The first row shows deblurred images and the second row shows the corresponding estimated kernels. The iteration numbers of all the three MAP-based methods are shown. For convenience, we use the same notation *i* and *g* as our framework, because the three methods share the same optimization framework. In (b) and (d), kernels are less accurate and deblurred images contains obvious ringing artifacts. The kernel of (c) contains isolated noise around the main trajectory, so the results contains ringing artifacts too. Ours achieves the best performance in both the kernel estimation and the deblurred image.

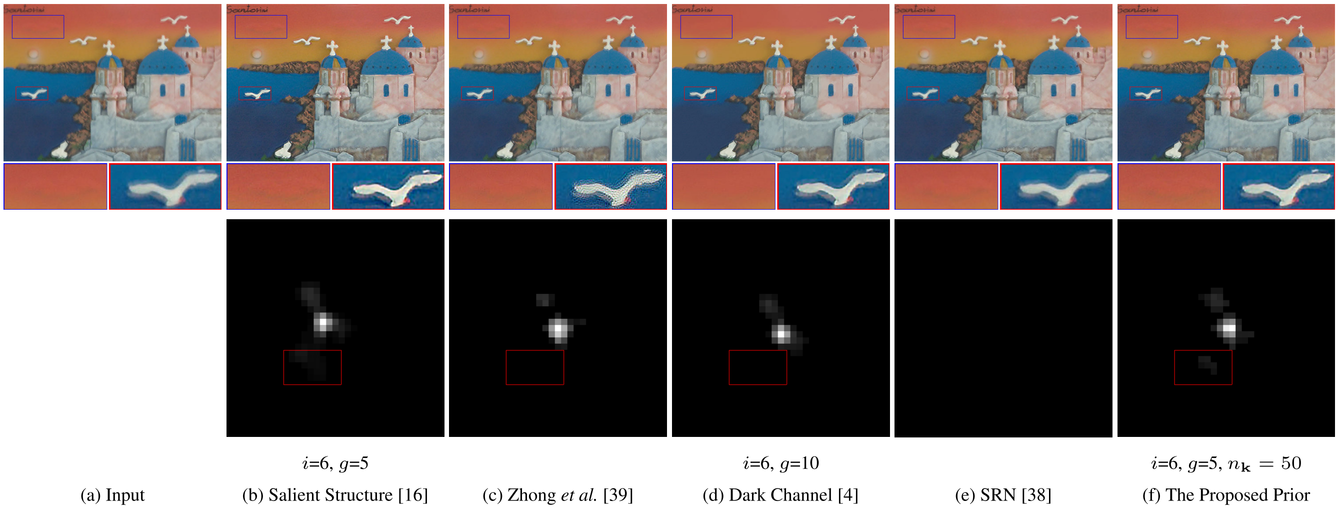
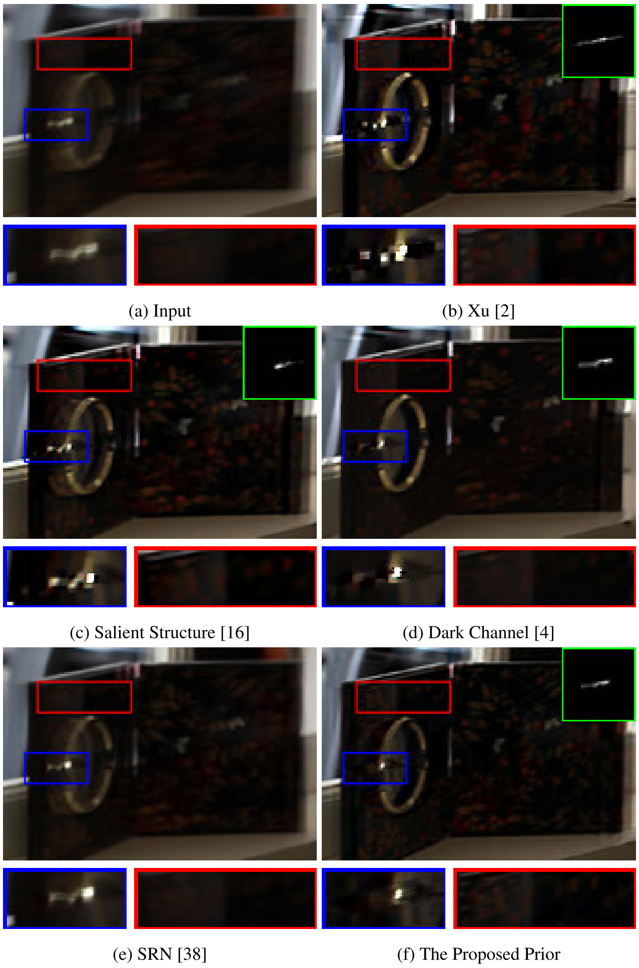


FIGURE 11. Deblurring results comparison on a blurred image with some noise. The first row shows deblurred images and the second row shows the corresponding estimated kernels of all the methods. The iteration numbers of each methods are shown as in Figure 10. Only (b) and (f) recover the main trajectory of the kernel successfully. As a result, both (b) and (f) deblur images with the least ringing artifacts and recover the cloud in the blue box.

For better comparison, we show the iteration numbers of Dark Channel [4] and Salient Structure [16]. We refer the iteration numbers in their published codes. We do not show iteration numbers of other conventional methods due to the lack of published codes. As most MAP-based methods share the same optimization framework, we use the same notation i and g to denote their iteration numbers. Coincidentally, all the competitors use the same iteration numbers. Kernel noise are obvious in (b), (c) and (d). Moreover, (b) and (d) fails to estimate the main trajectory of the kernel, which leads to ambiguous edges of the building. Without estimating a kernel, SRN [38] does not remove the blur accurately. Our method can estimate both the clear kernel trajectory and the natural sharp image.

2) NOISY-BLURRED IMAGEZ

Noisy-blurred image is another challenge for most deblurring method. It is because noise in the image can be easily regarded as strong edges, which leads to errors in the kernel estimation. Figure 11 shows deblurred results of an noisy-blurred image, in which Zhong et al. [39] is a kernel estimation method with the noise handling strategy. We also show the iteration numbers of each methods


(a) Input

(b) Xu [2]

(c) Salient Structure [16]

(d) Dark Channel [4]

(e) SRN [38]

(f) The Proposed Prior

FIGURE 12. Deblurring results comparison on an image with large blur. Deblurred results in (b), (c) and (f) contain obvious ringing artifacts. Both (d) and (g) achieve good performance, but our method in (f) recovers more details than (d).

as Figure 10. For the estimated kernels in the second row, neither Zhong et al. [39] nor Dark Channel Prior method [4] estimate the main trajectory of the kernel, as a branch of the trajectory is lost in their red boxes. Consequently, the blur is not well removed in (c) as shown in the zoomed-in red box and the cloud in the blue box is absent in (d). Both our method and the salient structure [16] estimate accurate kernels and outperform other methods. However, different from [16] that involves both the kernel continuity preservation and the edge selection strategy, our method only use the kernel continuity preservation to estimate the kernel. The performance shows that even though our method is not designed for noisy-blurred image, the continuity preservation makes the kernel estimation more robust to image noise.

3) LARGE BLUR IMAGE

The large blur is difficult to remove due to the large kernel size involving more unknown elements to be estimated. The image in Figure 12 is a real captured photo from [16]. The blur size is around 64 pixels, which is larger than those of Levin dataset [33] whose blur sizes are all under 30 pixels. (b) shows the deblurred result of Xu et al. [2], in which the blur is not well removed as in the red box and there are obvious ringing artifacts in the blue box. The result in (c) also shows obvious ringing artifacts in both the two zoomed-in regions. In (d), the zoomed-in region of the red box is over-smooth and most fine textures are lost. The result of SRN [38] are still blur. Our result outperform others by recovering a sharp image with fine textures. We also evaluate our method on Köhler Dataset [42] in Section VI-C, in which more large blur images are quantitatively evaluated.

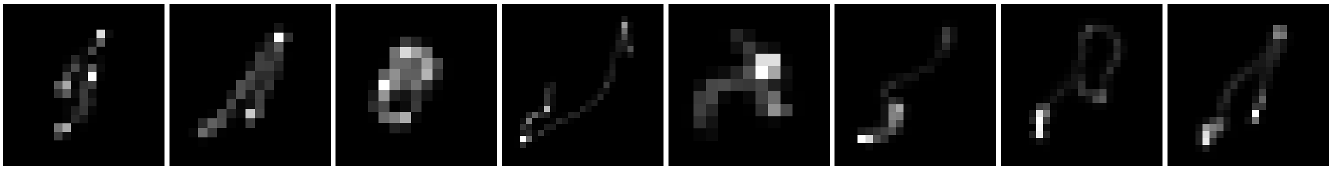
B. PERFORMANCE IMPROVEMENT WITH OUR PRIOR

The proposed kernel prior can also be used separately in any previous methods that use the same MAP-based framework. By replacing their kernel estimation sub-model with ours in Section IV-A, the proposed kernel prior can improve the performance of kernel estimation of previous methods. In this experiment, we choose the Dark Channel Prior method [4] as the baseline as it has the published code and achieves best performance in conventional optimization-based methods. The method uses the noise pruning as post-process method to remove noise in the estimated kernel. By using our kernel prior, we disable the noise pruning method in [4] for fair comparison. We test its performance on Sun Dataset [27] with 640 blurred images. All the 640 images can be divided into 8 groups, in which images in each group share the same blur kernel. The 8 blur kernels are shown in Figure 13. All the images in Sun Dataset [27] are nature images with forests, lakes or buildings.

We show the average PSNR values of each group in Table 4. The PSNR of Krishnan et al. [7] is reported in benchmark [27]. We evaluate Pan et al. [4] and Tao et al. [38] on the dataset using their published code. We replace the kernel estimation sub-model of Pan et al. [4] with ours and evaluate the performance on the same dataset. Using our prior, the average PSNR values are all higher than Pan et al. [4], especially for the larger kernels: $\mathbf{k}4$, $\mathbf{k}6$, $\mathbf{k}7$ and $\mathbf{k}8$. Using the proposed kernel prior, the average PSNR values improve in all the 8 kernels, especially in kernels with larger size: $\mathbf{k}4$, $\mathbf{k}6$, $\mathbf{k}7$ and $\mathbf{k}8$. It indicates that for images not belonging to challenging cases, our proposed prior can still benefit the kernel estimation and improve the performance of previous method.

C. EFFICIENCY

The proposed prior can be implemented more efficiently than other conventional methods. Our experimental environments are MATLAB2016b, Intel Core i7 and CPU @ 4.2GHZ*8. The complexity of the intermediate kernel estimation model is $O(N)$. Our model converges after less than 50 iterations, which only takes 3.2 seconds for kernel size 51 × 51. The intermediate kernel estimation takes more time as the kernel size increases, for example, 2.3 seconds (7 × 7), 2.4 seconds (11 × 11), 2.5 seconds (19 × 19), 2.7 seconds (27 × 27) and 2.9 seconds (37 × 37).



(a) **k** 1 (b) **k** 2 (c) **k** 3 (d) **k** 4 (e) **k** 5 (f) **k** 6 (f) **k** 7 (f) **k** 8

FIGURE 13. Blur kernels in Sun Dataset [27].

TABLE 4. Average PSNR of each 80 images for 8 blur kernels in Sun Dataset [27].

	k 1	**k** 2	**k** 3	**k** 4	**k** 5	**k** 6	**k** 7	**k** 8	Average
kernel size	19 × 19	17 × 17	15 × 15	27× 27	13× 13	21×21	23× 23	23× 23	
Krishnan *et al.* [7]	22.96	21.28	24.76	20.6705	24.85	22.51	23.46	23.21	22.96
Tao *et al.* [38]	24.45	25.77	24.86	20.08	23.80	19.44	20.48	21.10	22.50
Pan *et al.* [4] w/o our prior	27.75	28.46	26.18	22.13	29.20	20.16	22.40	23.46	24.97
Pan *et al.* [4] w/ our prior	**27.94**	**28.48**	**26.83**	**28.14**	**29.51**	**26.92**	**27.68**	**28.53**	**28.00**

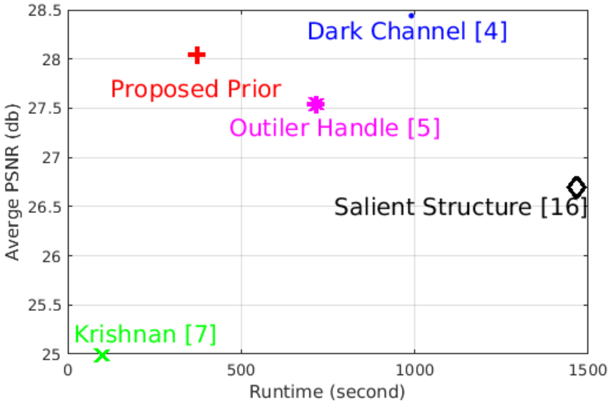
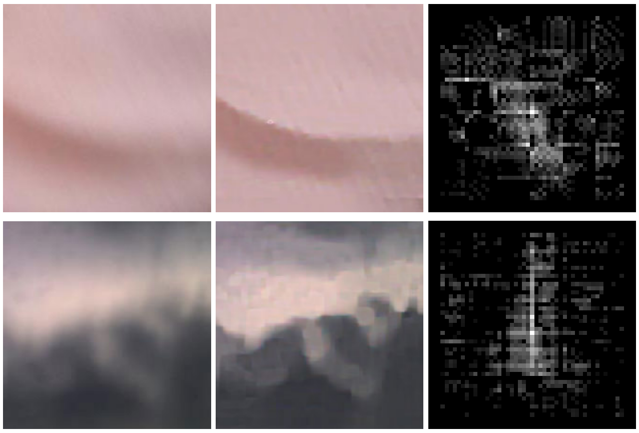


FIGURE 14. The PSNR vs. runtime of recent conventional deblurring methods [4], [5], [7], [16] and the proposed prior on the Kohler Dataset [42]. Even though the average PSNR of ours is a little lower than the Dark Channel method [4] (28.05 vs. 28.44), the proposed prior is almost 3 times faster (370s vs. 990s).



(a) Input (b) Deblur (c) Kernel

FIGURE 15. Two failure cases. Without strong edges in the input image, the proposed method fails to estimate accurate blur kernel. As a result, the deblurred images contain obvious artifacts.

To further evaluate the efficiency of the proposed prior, we compare the average runtime and PSNR of the kernel estimation model of several conventional methods [4], [5], [7], [16] on the same environment. For fair comparison, we use the same non-blind deconvolution method [4] to deblur images with their estimated kernels. We run the published code of each competitor on the Köhler Dataset [42]. The dataset contains 48 images whose sizes are 800 × 800 and their kernel sizes are from 31 to 145. We compute the average PSNR values and average runtime of each method as shown in Figure 14. The average PSNR value of our method is a little lower than the first rank, but ours is almost 3 times faster than it.

D. FAILURE CASES

As mentioned in Section V-A, the proposed kernel prior assists to recover strong and small-scale edges in the intermediate image, which improves the accuracy of the estimated kernel. Thus, the proposed prior will fail if the blurred image only contains fine textures without strong edges, because the fine textures are removed during the intermediate image estimation and few information can be used to estimate the kernel. Figure 15 shows two failure cases. As the two input images only contain weak and fine textures, the proposed method fails to estimate the blur kernel, which leads to degraded deblurred results.

VII. CONCLUSION

In the view of the significance of the kernel continuity description and preservation, we propose a kernel prior to preserve kernel continuity. The prior forms a continuity map of the kernel and encourages the sparsity of the map to preserve the kernel continuity. To deal with the non-convex and non-linear prior, we also propose an approximation to solve the cost function, which makes our method more efficient than many other conventional methods. Solid experimental results show that the proposed prior is able to help estimate a more accurate kernel even in some challenging cases:

small-scale blurred images, noisy-blurred images and large blur images. Also, the proposed prior can further improve the performance of previous method by leveraging our kernel continuity prior.

REFERENCES

[1] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. Eur. Conf. Comput. Vis.*, 2010.

[2] L. Xu, S. Zheng, and J. Jia, "Unnatural L_0 sparse representation for natural image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1107–1114.

[3] J. Pan, Z. Hu, Z. Su, and M.-H. Yang, "Deblurring text images via l_0-regularized intensity and gradient prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2901–2908.

[4] J. Pan, D. Sun, H. Pfister, and M. Yang, "Blind image deblurring using dark channel prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1628–1636.

[5] J. Pan, Z. Lin, Z. Su, and M. Yang, "Robust kernel estimation with outliers handling for image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2800–2808.

[6] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, 2006.

[7] D. Krishnan, T. Tay, and R. Fergus, "Blind deconvolution using a normalized sparsity measure," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 233–240.

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

[9] Z. Dou, K. Gao, X. Zhang, and H. Wang, "Fast blind image deblurring using smoothing-enhancing regularizer," *IEEE Access*, vol. 7, pp. 90904–90915, 2019.

[10] M. Zhao, X. Zhang, Z. Shi, P. Li, and B. Li, "Restoration of motion blurred images based on rich edge region extraction using a gray-level co-occurrence matrix," *IEEE Access*, vol. 6, pp. 15532–15540, 2018.

[11] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graph.*, vol. 28, no. 5, p. 145, 2009.

[12] N. Joshi, R. Szeliski, and D. J. Kriegman, "PSF estimation using sharp edge prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[13] C. Cai, H. Meng, and Q. Zhu, "Blind deconvolution for image deblurring based on edge enhancement and noise suppression," *IEEE Access*, vol. 6, pp. 58710–58718, 2018.

[14] X. Fang, Q. Zhou, J. Shen, C. Jacquemin, and L. Shao, "Text image deblurring using kernel sparsity prior," *IEEE Trans. Cybern.*, to be published.

[15] S. Tang, W. Zheng, X. Xie, T. He, P. Yang, L. Luo, Z. Li, Y. Hu, and H. Zhao, "Multi-regularization-constrained blur kernel estimation method for blind motion deblurring," *IEEE Access*, vol. 7, pp. 5296–5311, 2019.

[16] J. Pan, R. Liu, Z. Su, and X. Gu, "Kernel estimation from salient structure for robust motion deblurring," *Signal Process., Image Commun.*, vol. 28, no. 9, pp. 1156–1170, 2013.

[17] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman, "Blur kernel estimation using the radon transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 241–248.

[18] T. A. Javaran, H. Hassanpour, and V. Abolghasemi, "Blind motion image deblurring using an effective blur kernel prior," *Multimedia Tools Appl.*, pp. 1–20, 2019.

[19] D. Krishnan, J. Bruna, and R. Fergus, "Blind deconvolution with re-weighted sparsity promotion," *CoRR*, vol. abs/1311.4029, 2013.

[20] S. Cho and S. Lee, "Convergence analysis of MAP based blur kernel estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4808–4816.

[21] J. Dong, J. Pan, Z. Su, and M. Yang, "Blind image deblurring with outlier handling," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2497–2505.

[22] D. Gong, M. Tan, Y. Zhang, A. van den Hengel, and Q. Shi, "Self-paced kernel estimation for robust blind image deblurring," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1670–1679.

[23] J. Pan, Z. Hu, Z. Su, and M. Yang, "L_0-regularized intensity and gradient prior for deblurring text images and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 342–355, Feb. 2017.

[24] L. Sun, S. Cho, J. Wang, and J. Hays, "Good image priors for non-blind deconvolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 231–246.

[25] H. Zhang, D. Wipf, and Y. Zhang, "Multi-image blind deblurring using a coupled adaptive sparse prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1051–1058.

[26] W. Lai, J. Ding, Y. Lin, and Y. Chuang, "Blur kernel estimation using normalized color-line priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 64–72.

[27] L. Sun, S. Cho, J. Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *Proc. IEEE Int. Conf. Comput. Photography*, Apr. 2013, pp. 1–8.

[28] W. Ren, X. Cao, J. Pan, X. Guo, W. Zuo, and M.-H. Yang, "Image deblurring via enhanced low-rank prior," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3426–3437, Jul. 2016.

[29] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao, "Image deblurring via extreme channels prior," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6978–6986.

[30] 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.

[31] Z. Shen, T. Xu, J. Pan, and J. Guo, "Non-uniform motion deblurring with kernel grid regularization," *Signal Process., Image Commun.*, vol. 62, pp. 1–15, 2018.

[32] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Blurred/non-blurred image alignment using sparseness prior," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[33] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1964–1971.

[34] L. Li, J. Pan, W.-S. Lai, C. Gao, N. Sang, and M.-H. Yang, "Blind image deblurring via deep discriminative priors," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1025–1043, Jan. 2019.

[35] Z. Hu and M.-H. Yang, "Learning good regions to deblur images," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 345–362, 2015.

[36] S. Liu, H. Wang, J. Wang, S. Cho, and C. Pan, "Automatic blur-kernel-size estimation for motion deblurring," *Vis. Comput.*, vol. 31, no. 5, pp. 733–746, 2015.

[37] Y. Liu, J. Wang, S. Cho, A. Finkelstein, and S. Rusinkiewicz, "A no-reference metric for evaluating the quality of motion deblurring," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 175:1–175:12, 2013.

[38] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8174–8182.

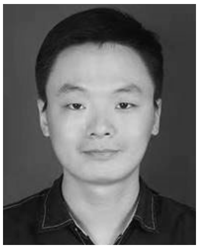
[39] L. Zhong, S. Cho, D. N. Metaxas, S. Paris, and J. Wang, "Handling noise in single image deblurring using directional filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 612–619.

[40] C. Min, G. Wen, B. Li, and F. Fan, "Blind deblurring via a novel recursive deep CNN improved by wavelet transform," *IEEE Access*, vol. 6, pp. 69242–69252, 2018.

[41] S. Zhang, P. Li, Y. Meng, L. Li, Q. Zhou, and X. Fu, "A video deblurring algorithm based on motion vector and an encorder-decoder network," *IEEE Access*, vol. 7, pp. 86778–86788, 2019.

[42] R. Köhler, M. Hirsch, B. J. Mohler, B. Schölkopf, and S. Harmeling, "Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 27–40.

XUELING CHEN received the B.S. degree from Northwestern Polytechnical University, in 2014, where she is currently pursuing the Ph.D. degree. She was sponsored by the China Scholarship Council as a Visiting Ph.D. Student at The University of Adelaide, from the Spring of 2017 to the early 2019. Her research interests include image deblurring, low-level image processing, and deep learning.



YU ZHU received the Ph.D. degree from Northwestern Polytechnical University (NPU), Xi'an, China, 2017. He is currently with NPU as an Assistant Researcher. His current research interests include computer vision, image super-resolution, and other image fusion topics.

JINQIU SUN received the B.S. degree from Northwestern Polytechnical University (NPU), in 1999, and the M.S. and Ph.D. degrees from NPU, in 2004 and 2005, respectively. She is currently a Professor with the School of Astronomy, NPU. Her research works focus on signal and image processing, computer vision, and pattern recognition.

YANNING ZHANG received the B.S. degree from the Dalian University of Science and Engineering, in 1988, and the M.S. and Ph.D. degrees from Northwestern Polytechnical University (NPU), in 1993 and 1996, respectively. She is currently a Professor with the School of Computer Science, Northwestern Polytechnical University. She is also the Organization Chair of the Ninth Asian Conference on Computer Vision (ACCV2009). Her research works focus on signal and image processing, computer vision, and pattern recognition. She has published over 200 articles in international journals, conferences, and Chinese key journals.