Robust Kernel Estimation with Outliers Handling for Image Deblurring

Jinshan Pan[1,2], Zhouchen Lin[3,4,*], Zhixun Su[1,5], and Ming-Hsuan Yang[2]

[1]School of Mathematical Sciences, Dalian University of Technology
[2]Electrical Engineering and Computer Science, University of California at Merced
[3]Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
[4]Cooperative Medianet Innovation Center, Shanghai Jiaotong University
[5]National Engineering Research Center of Digital Life

http://vllab.ucmerced.edu/~jinshan/projects/outlier-deblur/

Abstract

Estimating blur kernels from real world images is a challenging problem as the linear image formation assumption does not hold when significant outliers, such as saturated pixels and non-Gaussian noise, are present. While some existing non-blind deblurring algorithms can deal with outliers to a certain extent, few blind deblurring methods are developed to well estimate the blur kernels from the blurred images with outliers. In this paper, we present an algorithm to address this problem by exploiting reliable edges and removing outliers in the intermediate latent images, thereby estimating blur kernels robustly. We analyze the effects of outliers on kernel estimation and show that most state-of-the-art blind deblurring methods may recover delta kernels when blurred images contain significant outliers. We propose a robust energy function which describes the properties of outliers for the final latent image restoration. Furthermore, we show that the proposed algorithm can be applied to improve existing methods to deblur images with outliers. Extensive experiments on different kinds of challenging blurry images with significant amount of outliers demonstrate the proposed algorithm performs favorably against the state-of-the-art methods.

1. Introduction

Recent years have witnessed significant advances in single-image deblurring [13, 15, 18]. The success of state-of-the-art methods mainly stems from accurate restoration of sharp edges [3, 5, 14, 30, 36] or strong priors of natural images and blur kernels [2, 7, 17, 23, 29, 32, 38, 39]. While these methods are able to deblur natural images well, they are less effective for blurred inputs with significant amount of outliers (*e.g.*, saturated areas and non-Gaussian noise) as illustrated by examples in Figure 1.
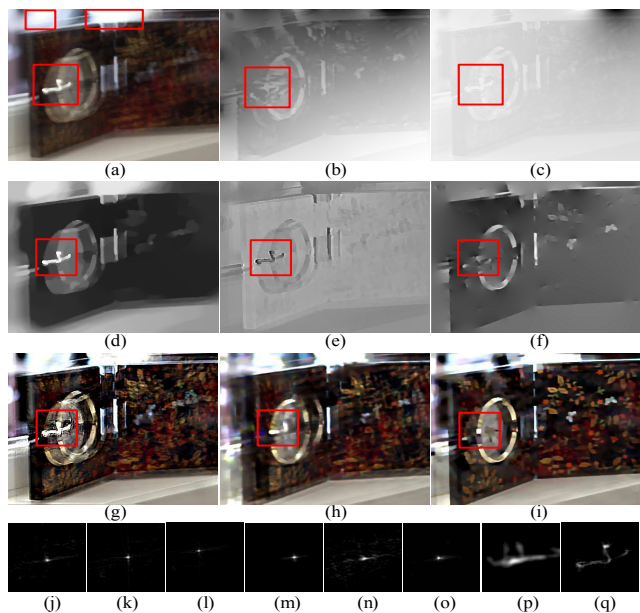
---

*Corresponding author.

Figure 1. Effects of outliers on blind deblurring. (a) A real captured blurred image with large saturated regions, *e.g.*, light streaks and blobs in red boxes. (b)-(e) Intermediate results generated by Cho and Lee [3], Xu and Jia [36], Xu *et al.* [38], and Pan *et al.* [27]. (f) Salient edges extracted by the proposed algorithm (shown by Poisson reconstruction). (g)-(h) Deblurred results of Xu and Jia [36] and Hu *et al.* [12]. (i) Deblurred result by the proposed algorithm. (j)-(o) Estimated kernels by Xu and Jia [36], Krishnan *et al.* [17], Levin *et al.* [21], Zhong *et al.* [40], Xu *et al.* [38], Pan *et al.* [27], and Pan *et al.* [26]. (p) Estimated kernel by the proposed algorithm. The red boxes in (a)-(e) and (g)-(h) contain saturated regions (best viewed on high-resolution displays).

The main reason that most algorithms do not perform well is that the gray levels of the pixels in the bright or specular regions are clipped to the maximum value (*e.g.*, 255) during the image formation process due to the limited quantization range of camera sensors. Thus, the linear blur model,

$$B = I * k + e, \tag{1}$$

that most deblurring methods assume does not hold for blurred images containing saturated areas, where B, I, k, e, and $*$ denote the blurred image, latent image, blur kernel, noise, and the convolution operator, respectively. In addition, some dead or hot pixels of camera sensors also affect the quality of captured images.

The problem with image outliers, especially saturated pixels, has not received much attention in blind deblurring, and existing methods mainly address its effect on non-blind deblurring. Harmeling *et al.* [9] propose a multi-frame blind deblurring method by thresholding the blurred image to detect saturated pixels. In [4], Cho *et al.* present detailed analysis on image outliers that affect restoration of latent images, and propose an EM-based non-blind deconvolution method. Since the pixels of saturated areas cannot be well described by the linear convolution model (1), Whyte *et al.* [34] extend the Richardson-Lucy algorithm [22] by specific functions. Recently, deep convolutional neural networks have been applied to restore latent images [37]. For blind deblurring, recent work [12] focuses on handling specific images with low light streaks. This method is effective when the light streaks can be detected but ineffective if the saturated regions are large (*e.g.*, bright blobs in Figure 1). In addition, a text image prior [26] is developed, which is able to deblur saturated images, but less effective for images with non-Gaussian noise.

The examples illustrated in Figure 1 reveal one critical problem with generic and specific blind deblurring methods (*e.g.*, [12, 26]). Namely, it is difficult to recover the blur kernel when the blurred image contains large number of saturated pixels. Furthermore, it is not a coincidence that the estimated kernels by [17, 21, 27, 36, 38, 40] are close to delta functions. Analysis and explanations of the estimated kernels by these methods are presented in Section 4.1.

We note that one specific outlier modeled by the nonlinear camera response function (CRF) has been addressed [31]. Although the nonlinear CRF affects kernel estimation, it can be avoided by using raw camera output, or alleviated by first applying an inverse response curve obtained from camera calibration [8]. Furthermore, the CRF estimation process can be estimated before kernel estimation as evidenced in [31], whereas the aforementioned outliers, *e.g.*, saturated pixels, cannot be effectively removed in advance (See Section 4.3). Although some aforementioned outliers, *e.g.*, impulse noise, can be removed from blurred images using filters (*e.g.*, median filter), kernels cannot be accurately estimated from the filtered results [40].

We address the blind deblurring problem for images with significant outliers such as saturated and clipped areas [4] and non-Gaussian noise [1] in this paper. In contrast to existing edge selection methods [3, 27, 36], we propose a novel scheme to extract reliable edges for kernel estima-

tion, thereby facilitating the subsequent optimization process without the effects of outliers. As shown in Figure 1(f), the selected salient edges contain fewer saturated pixels, which accordingly lead to a better estimated blur kernel (Figure 1(q)) and deblurred result with fine textures (Figure 1(i)).

The contributions of this work are summarized as follows. First, we propose an algorithm to remove outliers from selected edges for robust kernel estimation (See Figure 1(f)). We show that the proposed method is generic and applicable to existing blind deblurring methods (*e.g.*, [3, 17, 36, 38]) for better performance. In addition, we analyze the effects of outliers on kernel estimation and show that existing methods favor delta kernels for such blurred inputs. Second, we propose a robust deconvolution model according to the properties of outliers for latent image restoration. Third, the proposed method is extended to handle the non-uniform deblurring problem. Numerous experimental evaluations show that the proposed algorithm performs favorably against the state-of-the-art blind deblurring methods.

2. Robust Kernel Estimation

In this section, we present a robust kernel estimation method within the maximum a posteriori (MAP) framework to deal with outliers. We note that saturated and clipped pixels cannot be well described by the linear convolution model (1), and the non-Gaussian noise can be described by the noise term e in (1). To establish a proper blur model including the aforementioned outliers, we have the following formulation:

$$B = f(I * k) + e, \tag{2}$$

where $f(\cdot)$ can be either a piecewise function describing the saturated and clipped pixels, *i.e.*, if $I * k$ is within the dynamic range, $f(I * k) = I * k$ and otherwise $f(I * k)$ is a non-linear function (*e.g.*, a truncated function which returns the maximum or minimum intensity of the dynamic range), or an index function that indicates missing pixels. With this model, a blur kernel can be estimated by solving

$$\min_{I,k} \|B - f(I * k)\|_1 + \lambda_c E_I(I) + \gamma E_k(k), \tag{3}$$

where $E_I(I)$ and $E_k(k)$ are regularization terms on the latent image and blur kernel; λ_c and γ are positive scalar weights. The ℓ_1 norm is used in the first term to handle non-Gaussian noise [1]. Due to non-linearity of f, it is difficult to minimize (3) within the conventional MAP framework. In the following, we propose an efficient algorithm that does not estimate f directly. Central to our method is to select reliable salient edges (See the part in the blue box in Figure 2) that satisfy the linear convolution model (1) for kernel estimation. Figure 2 shows the main steps of the proposed deblurring algorithm. We first describe the proposed algorithm and then analyze the components in Section 4.
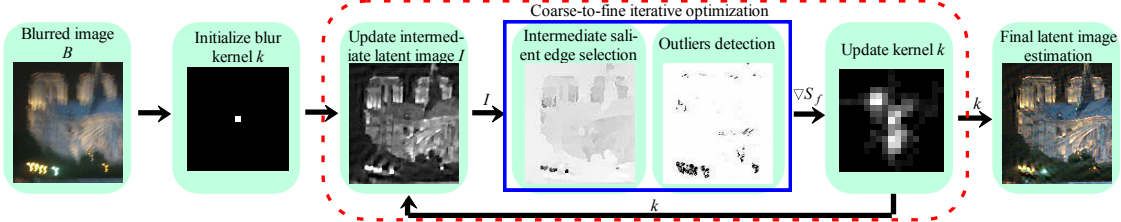
Figure 2. Main steps of the proposed algorithm. The dotted red line box encloses the process of kernel estimation, and the solid blue line box encloses our core contribution.

2.1. Update Intermediate Latent Image I

Within the MAP framework, the intermediate image I can be obtained when the blur kernel k is known,

$$\min_I \|f(I * k) - B\|_1 + \lambda_c E_I(I). \quad (4)$$

It is difficult to minimize (4) as f is non-linear and its concrete form is assumed to be unknown in this work. To generate an intermediate image for kernel estimation, we approximate the data fitting term with $\|I * k - B\|_1$. This substitution makes (4) tractable but may lead to the results with ringing artifacts (See the intermediate images in Figures 2 and 3). However, we do not need accurate intermediate results at this stage. The proposed edge selection method (illustrated in Section 2.2) is able to select reliable edges from this approximated intermediate result for kernel estimation. For the regularization term $E_I(I)$, we use the hyper-Laplacian prior [19] to estimate the intermediate image. Thus, the intermediate latent image is estimated by

$$\min_I \sum_{\mathbf{x}} |(I * k - B)_{\mathbf{x}}| + \lambda_c(|(\partial_x I)_{\mathbf{x}}|^{0.8} + |(\partial_y I)_{\mathbf{x}}|^{0.8}), \quad (5)$$

where the subscript $\mathbf{x}$ denotes the spatial location of a pixel. We use the iteratively re-weighted least squares (IRLS) [19] method to solve (5). At each iteration, we need to solve the quadratic problem:

$$\min_{I^{\{t\}}} \sum_{\mathbf{x}} w_d(\mathbf{x}) |(I^{\{t\}} * k - B)_{\mathbf{x}}|^2 + \\ \lambda_c(w_r^h(\mathbf{x}) |(\partial_x I^{\{t\}})_{\mathbf{x}}|^2 + w_r^v(\mathbf{x}) |(\partial_y I^{\{t\}})_{\mathbf{x}}|^2), \quad (6)$$

where $w_d(\mathbf{x}) = |(I^{\{t-1\}} * k - B)_{\mathbf{x}}|^{-1}$, $w_r^h(\mathbf{x}) = |(\partial_x I^{\{t-1\}})_{\mathbf{x}}|^{-1.2}$, $w_r^v(\mathbf{x}) = |(\partial_y I^{\{t-1\}})_{\mathbf{x}}|^{-1.2}$, and t denotes the iteration index.

2.2. Update Blur Kernel k

Most state-of-the-art methods either use the gradient properties [17, 21, 29] or select sharp edges [3, 27, 36] from the estimated intermediate image I for kernel estimation. However, those aforementioned methods are less effective for kernel estimation when the blurred image B contains numerous saturated or clipped pixels, as these outliers are salient and considered as important salient edges for kernel estimation (See Figure 1). To address these issues, we introduce a model to select intermediate salient edges (Section 2.2.1) such that tiny details corresponding to small image gradients are removed. We then propose a method to

detect and remove outliers from intermediate salient edges (Section 2.2.2).

2.2.1 Intermediate salient edge selection

The goal of edge selection in most blind deblurring methods [3, 36] is to retain large image gradients and remove tiny details. In this work, we propose a model to select salient edges from an intermediate image I,

$$\min_{\nabla S} \sum_{\mathbf{x}} \frac{1}{2} \|\nabla S(\mathbf{x}) - \nabla \Phi(I(\mathbf{x}))\|_2^2 + \theta \omega(\mathbf{x}) \|\nabla S(\mathbf{x})\|_1, \quad (7)$$

where $\Phi(\cdot)$ denotes the operation of shock filter [24]; $\nabla S(\mathbf{x}) = (\partial_x S(\mathbf{x}), \partial_y S(\mathbf{x}))^\top$ corresponds to the gradients $\nabla \Phi(I(\mathbf{x})) = (\partial_x \Phi(I(\mathbf{x})), \partial_y \Phi(I(\mathbf{x})))^\top$; $\|\nabla S(\mathbf{x})\|_1 = |\partial_x S(\mathbf{x})| + |\partial_y S(\mathbf{x})|$ is an anisotropic total variation (TV) regularizer to preserve the sharp edges; θ is a weight parameter; and $\omega(\mathbf{x}) = \exp(-(r(\mathbf{x}))^{0.8})$ with

$$r(\mathbf{x}) = \frac{\|\sum_{\mathbf{y} \in N_h(\mathbf{x})} \nabla B(\mathbf{y})\|_2^2}{\sum_{\mathbf{y} \in N_h(\mathbf{x})} \|\nabla B(\mathbf{y})\|_2^2 + 0.5}. \quad (8)$$

In the above equation based on [36], $B(\mathbf{y})$ is the blurred image and $N_h(\mathbf{x})$ is an $h \times h$ window centered at pixel $\mathbf{x}$. A small value of $r(\mathbf{x})$ indicates that the area is flat, whereas a large value of $r(\mathbf{x})$ suggests that there exist strong image structures in the local window.

The model in (7) is in spirit similar to edge-preserving methods [6, 27] that are developed to keep large gradients. Since our goal is to select gradients from the intermediate image I, we use gradients as the data fitting term, which accordingly leads to a closed-form solution of (7) based on the shrinkage formula:

$$\begin{cases} \partial_x S(\mathbf{x}) = \text{sign}(\partial_x \Phi(I(\mathbf{x}))) \max(|\partial_x \Phi(I(\mathbf{x}))| - \theta \omega(\mathbf{x}), 0), \\ \partial_y S(\mathbf{x}) = \text{sign}(\partial_y \Phi(I(\mathbf{x}))) \max(|\partial_y \Phi(I(\mathbf{x}))| - \theta \omega(\mathbf{x}), 0). \end{cases} \quad (9)$$

Thus, a solution of (7) enforces ∇S that only contains gradient values larger than $\theta \omega(\mathbf{x})$.

2.2.2 Outliers removal from intermediate salient edges

Although the proposed model in (7) can remove tiny details and retain salient edges for kernel estimation, it is based on large gradients and thus only effective for the blurred images without significant amount of outliers. Figure 3(c) shows one example that saturated areas are not removed based on the edges extracted by (9).
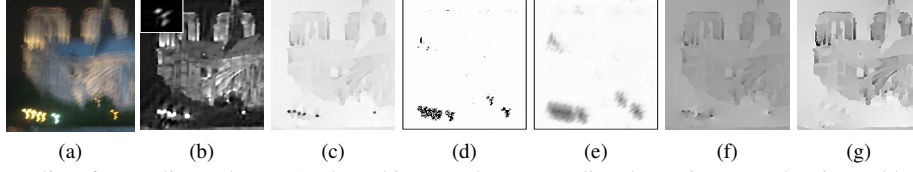
| | | | | | | |
| (a) | (b) | (c) | (d) | (e) | (f) | (g) |

Figure 3. Removing outliers from salient edges. (a) Blurred image. (b) Intermediate latent image and estimated kernel. (c) Edge map of salient edges ∇S extracted by (9). (d) Visualization of $\mathcal{M}$. (e) Visualization of $\mathcal{M} * \mathcal{R}(k)$. (f) Result after applying $\mathcal{M}$ on the edges in (c) (*i.e.*, $\mathcal{M} \circ \nabla S$). (g) Result after applying $(\mathcal{M} * \mathcal{R}(k))$ on the edges in (c) (*i.e.*, $(\mathcal{M} * \mathcal{R}(k)) \circ \nabla S$).

We present a method to remove outliers from the intermediate salient edges ∇S. As the linear model in (1) does not hold for blurred images with outliers, a pixel $\mathbf{x}$ is likely to be an outlier if the value of $I * k - B$ is large. An intuitive approach is to apply the thresholding strategy to $I * k - B$. That is, we set the value of $I * k - B$ at $\mathbf{x}$ to be close to 0 if the value of $I * k - B$ is large at $\mathbf{x}$, and 1 otherwise. As the aforementioned analysis is similar to the binary classification problem, we employ the sigmoid function which is widely used in neural network [10] and logistic regression for such tasks,

$$O(x) = \frac{1}{1 + ae^{cx}}, \tag{10}$$

where a and c are positive constants. Figure 4(a) shows an example of $O(x)$. As the sigmoid function monotonically decreases to 0 when x goes to infinity, we use it to detect outliers as follows,

$$\mathcal{M} = \frac{1}{1 + ae^{c(I*k-B)^2}}. \tag{11}$$

Figure 3(d) shows one example where outliers can be detected using (11) from Figure 3(b).

Since $\mathcal{M}$ is computed from the blurred input, it cannot be directly applied to the extracted salient edges ∇S. We note that the positions of dark regions in $\mathcal{M}$ do not always correspond to the positions of outliers in an intermediate edge extracted by (9) due to the influences of blur. Directly applying $\mathcal{M}$ to ∇S is not able to remove outliers effectively (See Figure 3(f)). To remove outliers from ∇S, we use the following formula:

$$\nabla S_f = (\mathcal{M} * \mathcal{R}(k)) \circ \nabla S, \tag{12}$$

where $\circ$ denotes a pixel-wise multiplication operator and $\mathcal{R}(k)$ is a mirrored function, which rotates k counterclockwise by 180 degrees. The use of $\mathcal{M} * \mathcal{R}(k)$ makes the positions of dark regions in $\mathcal{M}$ cover the positions of outliers in an intermediate edges, and thus remove outliers from ∇S (See Figure 3(g)). After removing outliers in the edges of an intermediate image, we approximate $f(\nabla S_f * k)$ by $\nabla S_f * k$. To further increase the robustness to outliers, we apply $\mathcal{M}$ to the blurred image (outliers are not used for kernel estimation). A blur kernel can thus be estimated by

$$\min_k \|\mathcal{M} \circ (\nabla S_f * k - \nabla B)\|_1 + \gamma \|k\|_1. \tag{13}$$

In (13), the ℓ_1 norm is used to preserve the sparsity of blur kernel k. Similar to (5), we also use the IRLS method to solve (13).
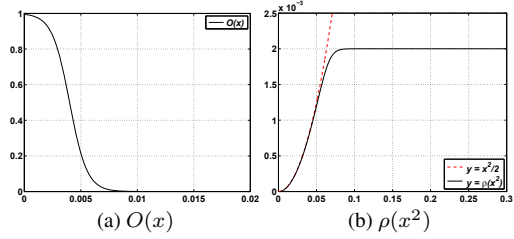

| | |
| (a) $O(x)$ | (b) $\rho(x^2)$ |

Figure 4. A sigmoid function $O(x)$ and the proposed outlier-aware function $\rho(x^2)$ used in final latent image restoration. The values of a and c in (10) are set to be 0.0055 and 1300.5, respectively.

2.3. Final Latent Image Estimation

Existing methods [1, 36] show that the use of $\|I * k - B\|_1$ in deblurring (once the blur kernel is determined) is more robust to outliers. While such approaches that use this data fitting cost are able to deal with large Gaussian or impulse noise [1], they are less effective if blurred images contain saturated pixels [4]. We note that if the image does not contain outliers, the use of $\|I * k - B\|_2^2$ in (5) is able to generate high-quality results [16, 19]. We also note that if the image contains outliers, *e.g.*, saturated areas, useful information in the regions containing outliers is missing (*e.g.*, the saturated regions are usually smooth in clear images (Figure 5(f))). A natural way is to use the regularization term (*e.g.*, the second term in (5)) to smooth the regions containing outliers in the image restoration since the data fidelity is not reliable. Based on above discussions, our goal is to find a function $\rho(x)$ that considers the presence of outliers and satisfies: a) $\rho(|(I * k - B)_\mathbf{x}|^2) \approx |(I * k - B)_\mathbf{x}|^2$, if $\mathbf{x}$ is not an outlier; b) $\rho(|(I * k - B)_\mathbf{x}|^2)$ is close to be a constant, if $\mathbf{x}$ is an outlier. We note that the function

$$\rho(x^2) = x^2/2 - \log(\sigma_1 \exp(\sigma_2 x^2) + 1)/(2\sigma_2), \tag{14}$$

satisfies the above conditions (See Figure 4(b)) where σ_1 and σ_2 are positive constants. The image restoration model is

$$\min_I \sum_\mathbf{x} \rho(|(I * k - B)_\mathbf{x}|^2) + \lambda(|(\partial_x I)_\mathbf{x}|^{0.8} + |(\partial_y I)_\mathbf{x}|^{0.8}), \tag{15}$$

where $\lambda > 0$ is a regularization weight.

We use the IRLS method to solve (15). At the t-th iteration, we need to solve the following problem:

$$\min_{I^{\{t\}}} \sum_\mathbf{x} w_f(|(I^{\{t-1\}} * k - B)_\mathbf{x}|^2)|(I^{\{t\}} * k - B)_\mathbf{x}|^2 \\ + \lambda(w_r^h(\mathbf{x})|(\partial_x I^{\{t\}})_\mathbf{x}|^2 + w_r^v(\mathbf{x})|(\partial_y I^{\{t\}})_\mathbf{x}|^2), \tag{16}$$

(a) Input and kernel (b) TV-ℓ_1 result [36] (c) Whyte *et al.* [34]
PSNR: 19.45   PSNR: 21.71   PSNR: 20.10

(d) Cho *et al.* [4] (e) Proposed result (f) Ground truth
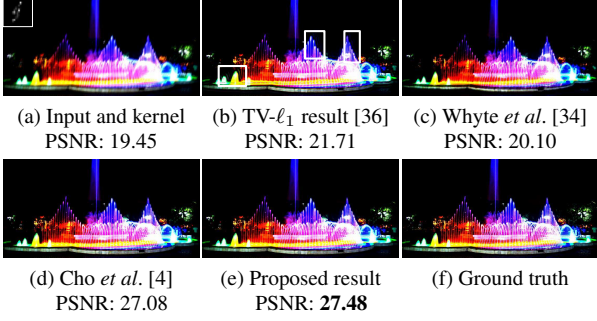PSNR: 27.08   PSNR: **27.48**

Figure 5. Deblurred results on an image containing saturated areas. The white boxes shown in (b) contain ringing artifacts.

where $w_r^h(\mathbf{x})$ and $w_r^v(\mathbf{x})$ are the same as those in (6), and the weight $w_f(|(I^{\{t-1\}} * k - B)_{\mathbf{x}}|^2)$ is the outlier detection function $\mathcal{M}$ if we set $\sigma_1 = a$ and $\sigma_2 = c$, which also shows that (15) is able to deal with outliers.

Figure 5(a) shows an example with several large saturated areas, where the saturated areas are created according to [4]. Although the TV-ℓ_1 method is robust to outliers, *e.g.*, impulse noise, the deblurred result in Figure 5(b) shows that this method is not robust to saturated areas. The deblurred result by the proposed model based on (15) is visually comparable or better than methods in [34] and [4] which are specifically designed to deal with saturated pixels.

Algorithm 1 Proposed blind deblurring algorithm

Input: Blurred image B.
Output: Latent image I and blur kernel k.
 1: Initialize the intermediate image I and kernel k with the results from the coarser level.
 2: **for** $l = 1 \rightarrow 4$ **do**
 3:     Estimate I according to (5);
 4:     Estimate ∇S according to (7);
 5:     Remove outliers from ∇S according to (12);
 6:     Estimate k according to (13);
 7: **end for**
 8: Estimate the final latent image by solving (16).

Similar to the state-of-the-art methods, we use a coarse-to-fine approach with an image pyramid [3]. Algorithm 1 presents the main steps for kernel estimation algorithm.

3. Extension to Non-Uniform Deblurring

Based on the geometric camera model [33, 35], the proposed algorithm can be extended to non-uniform deblurring as follows,

$$\mathbf{B} = f(\sum_i^N \mu_i \mathbf{b}_i \mathbf{I}) + \mathbf{e}, \qquad (17)$$

where $\mathbf{B}$, $\mathbf{I}$, and $\mathbf{e}$ are the vector forms in (2); $\mathbf{b}_i$ is the kernel basis induced by a possible camera shake which can be computed in advance [11]; μ_i is the weight corresponding to $\mathbf{b}_i$; and N is the number of kernel basis. Based on (17), we replace the terms $I * k$ and $\|k\|_1$ with $\sum_i^N \mu_i \mathbf{b}_i \mathbf{I}$
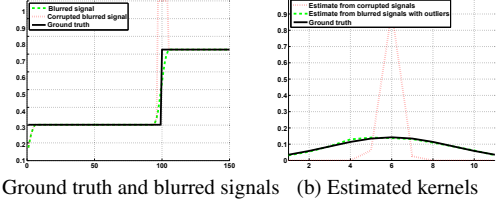

(a) Ground truth and blurred signals (b) Estimated kernels

Figure 6. Effects of outliers on kernel estimation. (b) shows the results of [36] using the signals shown in (a).

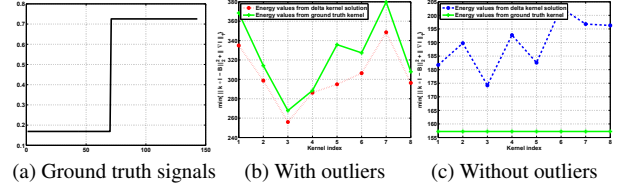(a) Ground truth signals (b) With outliers (c) Without outliers

Figure 7. Energy values when the inputs contain outliers or not. (b)-(c) Energy values of a MAP model [20] when the blurred signals contain outliers or not.

($\mu = \{\mu_i\}_{i=1}^N$) and $\|\mu\|_1$ in the proposed method described in Section 2. We use the efficient filter flow [11] together with the IRLS method described in Section 2.1 to compute latent images and blur kernels.

4. Analysis of Proposed Algorithm

In this section we analyze how outliers affect kernel estimation and explain the steps of the proposed algorithm to remove outliers for kernel estimation.

4.1. Effects of Outliers on Kernel Estimation

The effects of outliers on restoration of latent images have been discussed in [4]. In addition to image restoration, we show and explain why the state-of-the-art blind deblurring methods [3, 7, 17, 21, 36] generate delta functions for kernel estimation when the blurred images contain outliers.

As illustrated in [20], conventional MAP-based kernel estimation methods usually perform well for step edges (See Figure 6(a)) if outliers do not exist. Figure 6(a) shows one example with 1D signals. As the corrupted signals cannot be modeled well by the linear convolution model (1), the state-of-the-art method [36] is less effective in kernel estimation (See Figure 6(b)). Since the solution with the delta kernel has lower energy (*i.e.*, y-axis in Figure 7(b)), methods based on linear convolution without considering outliers favor such estimates. Numerous state-of-the-art methods (*e.g.*, [3, 36]) are effective in estimating kernels when the blurred signals do not contain outliers. This is mainly because the delta kernel solution has higher objective function energy values than those of ground truth kernels when the signals are not corrupted by outliers. The quantitative results in Figure 7(b) and (c) illustrate that existing MAP-based deblurring methods are likely to converge to the delta kernel solutions if the input signals contain outliers.

We propose a sigmoid function (11) to detect the outliers
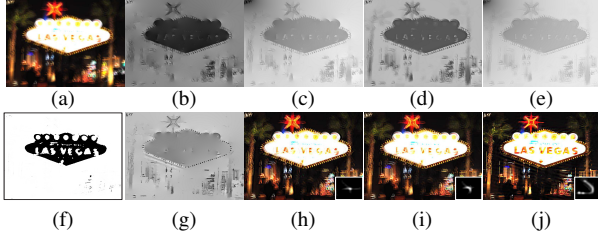
(a)　(b)　(c)　(d)　(e)

(f)　(g)　(h)　(i)　(j)

Figure 8. Outlier removal methods. (a) Blurred image. (b)-(d) Edge maps generated by [3], [36], and [27]. (e) and (g) denote the edge maps by the proposed algorithm without and with $\mathcal{M}$. (f) Our detected $\mathcal{M}$. (h) Results of [36]. (i)-(j) Ours without and with $\mathcal{M}$ (best viewed on high-resolution displays with zoom-in).
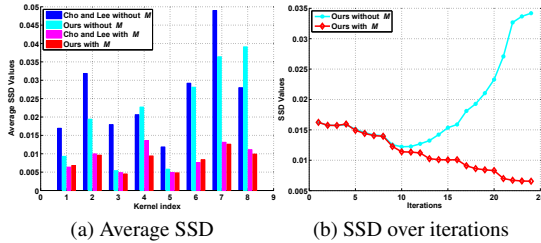

(a) Average SSD　　(b) SSD over iterations

Figure 9. Application and fast convergence property of the proposed algorithm (See supplemental material for the dataset).

from blurred images. As a result, only the salient edges that satisfy the linear convolution model are retained for kernel estimation. We note that large gradients do not always facilitate kernel estimation when a blurred image contains outliers. As such, we use (12) to remove some large gradients whose positions correspond to the outliers from intermediate edges ∇S. We carry out several experiments for validation and one example is shown in Figure 8(a). More results can be found in the supplemental material.

As saturated areas in Figure 8(a) are salient (*e.g.*, the white blobs), edge selection methods [3, 36] based on large gradients are likely to select these areas (Figure 8(b) and (c)) for kernel estimation. Since the linear convolution model (1) does not hold for the pixels in the saturated areas, kernels cannot be estimated well from the salient edges shown in Figure 8(b) and (c). The estimated kernel in Figure 8(h) is similar to a delta kernel as discussed above. Although the algorithm [27] adopts an adaptive TV denoising step to select salient edges, which is similar to the intermediate edge selection step (7) of the proposed algorithm, this method is not able to remove the outliers from salient edges (Figure 8(d)). Thus, the selected salient edges still contain saturated pixels which accordingly affect kernel estimation.

We note that although the use of ℓ_1 metric norm for data fitting in kernel estimation can deal with large Gaussian or impulse noise, this method is less effective (See Figure 8(i)) as it does not remove the saturated areas in the selected salient edges (See Figure 8(e)). Different from existing edge selection methods [3, 27, 36], the proposed algorithm does not necessarily select the salient edges with the largest gra-

dients when blurred image contains saturated areas. As the pixels in the saturated areas (See Figure 8(a)) cannot be described well by the linear convolution model (1), most outliers are detected (Figure 8(f)) and removed in the salient edges by applying (12) although they have large gradients.

4.2. Convergence of Proposed Algorithm

We note that the proposed algorithm without $\mathcal{M}$ degrades to an edge-based deblurring method for images without outliers. Figure 9(b) shows that the sum of squared differences (SSD) of an estimated kernel increases with respect to the iteration due to the effects of outliers. Since the proposed kernel estimation algorithm uses edges where outliers are removed as much as possible, the SSD error converges well as shown in Figure 9(b). We show that existing blind deblurring methods with $\mathcal{M}$ have better convergence properties in Section 4.4.

4.3. Effectiveness of Outlier Detection Method

We note that the intensities of saturated regions (*e.g.*, Figure 1(a) or Figure 8(a)) are usually high. Thus, a straightforward approach is to find the brightest pixels from a blurred input and discard them before kernel estimation. Although this simple strategy is able to deal with some cases [35], it does not remove outliers especially when a blurred image contains large saturated regions as shown in Figure 10(a).

As discussed in Section 1, the outlier handling methods [4, 34] mainly focus on the non-blind image deblurring, and both approaches use the blind deblurring method [3] to estimate blur kernels on the image patches without outliers. However, it is difficult to select a good image patch when the outliers are uniformly distributed in a blurred image (*e.g.*, impulse noise). Without good kernel estimates, clear latent images cannot be recovered well [4, 34]. We note that [4] involves an outlier detection step as an EM approach is developed. To clarify the difference, we use our kernel estimates together with the outlier detection result of [4] for fair comparisons. Figure 10(b) shows that some saturated regions are not detected by [4], which accordingly affect the final deblurred result (Figure 10(e)). In contrast, the proposed method detects the saturated regions and recovers the latent image well (Figure 10(f)). In addition, we show that our method helps improve kernel estimations by [3] in Section 4.4, which accordingly improves the performance of [4, 34].

4.4. Improvement on Existing Deblurring Methods

Since the proposed algorithm described in Section 2.2.2 is effective for removing outliers, we show that it facilitates MAP-based deblurring methods (*e.g.*, [3, 17, 36, 38]) to deblur images. In each deblurring method, we first compute $\mathcal{M}$ from previous estimated intermediate latent images, and then estimate blur kernels and intermediate latent images under the guidance of $\mathcal{M}$. We use one blind deblurring
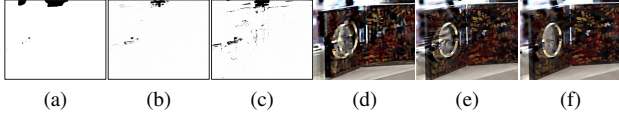
Figure 10. Deblurred results using different masks on the blurred input in Figure 1(a). (a) Intensity mask where dark regions indicate that the pixels with intensity values are larger than 240. (b) Result of [4]. (c) Our detected $\mathcal{M}$. (d)-(f) Results using (a), (b), and (c), respectively.

method [3] as an example in this section, and present more results in the supplemental material. The quantitative evaluations in Figure 9(a) show that the proposed outlier detection algorithm significantly improves the accuracy of [3].

We note that even when a blurred image does not contain saturated and clipped pixels, the proposed algorithm still contributes to kernel estimation. As blurred images may contain ambiguous edges [25, 36], these edges are likely to be selected as they are salient [25] and thus affect kernel estimation. However, we can still use (11) to detect the positions of these ambiguous edges as they cannot be modeled well by the linear convolution model (1), and then remove them from the intermediate edges ∇S such that only the reliable ones are retained for kernel estimation. We present experimental results on challenging examples without outliers in Section 5.

5. Experimental Results

The proposed algorithm is implemented in MATLAB on a computer with an Intel Core i7-4790 CPU and 28 GB RAM. The kernel estimation process takes 3 minutes for a 255×255 image with a 25×25 kernel without code optimization. All the color images are converted to grayscale ones in the kernel estimation process. The parameter λ_c in (5) is set to be 0.5, γ in (13) is set to be 0.01, θ in (7) is set to be 1, a and c in (10) are set to be 0.0055 and 1300.5, respectively, and λ in (15) is adjusted according to the amount of outliers. In the final deconvolution process, each color channel is processed independently. The MATLAB code and dataset are available at the authors' websites. In the following, we present examples with different outliers, *e.g.*, impulse noise and saturated areas. As mentioned in Section 1, the nonlinear CRF can be estimated before kernel estimation, so we do not consider this outlier in the following synthetic and real examples. Due to the space limit, we present large images and more results in the supplemental material.

Synthetic Images: Figure 11(a) shows a synthetic example with saturated regions and impulse noise. As the pixels in the saturated regions cannot be described well by the linear convolution model, one state-of-the-art method [38] is less effective in estimating kernels from this input image. The estimated kernels by this method is similar to delta functions, which can be explained by the analysis in Section 4.1. We note that the recent work [12] hinges on

(a) Input   (b) [40]   (c) [38]   (d) [12]
PSNR: 17.84   PSNR: 16.53   PSNR: 18.06   PSNR: 15.02

(e) [26]   (f) Our kernel+[4] (g) Our kernel+[34] (h) Ours
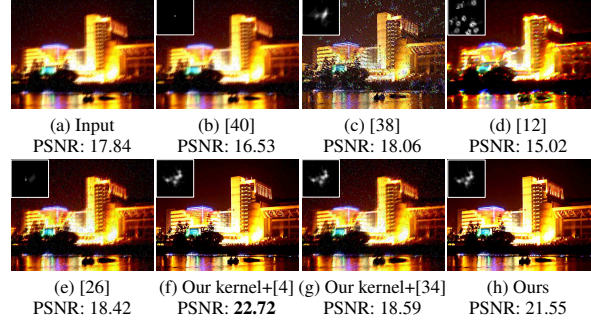PSNR: 18.42   PSNR: **22.72**   PSNR: 18.59   PSNR: 21.55

Figure 11. A synthetic example with saturated regions and impulse noise (best viewed on high-resolution displays with zoom-in).

detection of light streaks from a blurred image for kernel estimation. When light streaks are not detected well, this method is less effective as shown in Figure 11(d). Although the method [40] is designed to deal with Gaussian noise, it is less effective for saturated areas. We also note that the method [26] is able to deal with saturated images, but less effective for this example due to noise (See Figure 11(e)). In contrast, the proposed algorithm is able to detect the saturated regions, which facilitates kernel estimation and the deblurred results contain fine textures. We further compare with outlier handling methods [4, 34]. As [4, 34] mainly focus on non-blind image deblurring with outliers, we use our estimated kernels to generate the final results. With our estimated kernel, high-quality deblurred results can be obtained by [4]. As the method by [34] is mainly designed for handling saturated areas, it is less robust to impulse noise.

Real Images: We use a real example to evaluate the proposed algorithm. Figure 12(a) shows a real captured example with several saturated areas and unknown noise. Again, state-of-the-art methods [17, 21, 38] do not perform well on this example due to effects of saturated areas. Method [12] also fails to generate clear results due to unavailable light-streaks (See Figure 12(e)). Although the method by [4] generates much clearer results by our estimated kernel, the deblurred image still contains significant artifacts. In contrast, our method successfully estimates the blur kernel and generates a better deblurred result. Moreover, the comparison results shown in Figure 12(g) and (h) demonstrate that the proposed algorithm with $\mathcal{M}$ is able to remove outliers in the kernel estimation.

Non-Uniform Deblurring: We evaluate the proposed algorithm against the state-of-the-art methods [35, 38] for non-uniform image deblurring. Figure 13 shows one real-captured image from [35] in which the proposed algorithm performs favorably with sharper results.

Images without Outliers and Noise: As discussed in Section 4.4, the proposed algorithm can be applied to deblur images without containing outliers and noise. Figure 14(a) shows an example without outliers from [38]. The proposed method is able to detect the positions of ambiguous edges
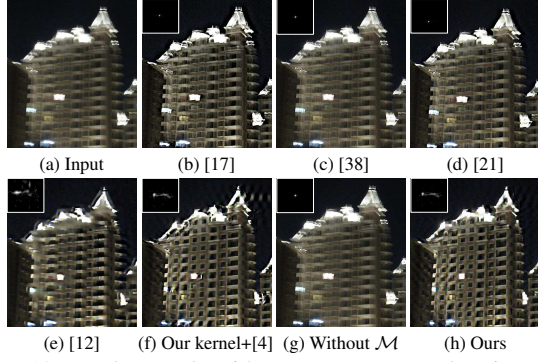
(a) Input   (b) [17]   (c) [38]   (d) [21]

(e) [12]   (f) Our kernel+[4]   (g) Without $\mathcal{M}$ (h) Ours

Figure 12. Real example with numerous saturated regions (*e.g.*, the light blobs). The kernel size is estimated at 45×45 pixels.
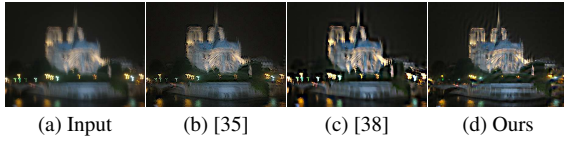

(a) Input   (b) [35]   (c) [38]   (d) Ours

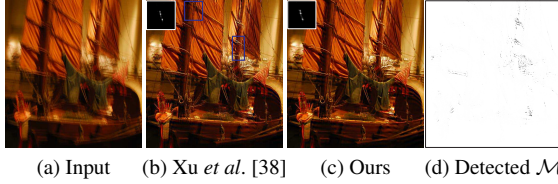Figure 13. Real captured non-uniform example from [35] (best viewed on high-resolution displays with zoom-in).

(a) Input   (b) Xu *et al.* [38]   (c) Ours   (d) Detected $\mathcal{M}$

Figure 14. A blurred image without outliers. The parts enclosed in blue boxes in (b) contain ringing artifacts. The size of estimated kernel is 45×45 pixels (best viewed on high-resolution displays with zoom-in).

where the linear convolution model does not hold from extracted salient edges (See Figure 14(d)). The estimated latent image contains fewer ringing artifacts compared to the result by [38].

Benchmark Datasets with and without Outliers: We use the benchmark dataset by Levin *et al.* [20] for quantitative evaluation in which we add the salt and pepper noise (as it is one of the most common outliers [1]) to each image. The noise density is set to be 0.01. We evaluate the performance of the proposed algorithm against the state-of-the-art methods [3, 7, 17, 21, 28, 36] and one deblurring algorithm that also deals with noise [40]. The error ratio metric [20] is used for quantitative evaluations. Figure 15(a) shows that the proposed algorithm achieves favorable results against state-of-the-art methods. We further evaluate the proposed algorithm using the images with different noise densities. Figure 15(b) shows that the proposed algorithm performs well even when the noise density is high.

We create a dataset containing 10 ground truth images with saturated regions and 8 kernels from [20]. The size of the saturated regions in this dataset is from 5×5 to 400×400 pixels. Similar to [4], we stretch the intensity histogram range of each image into $[0, 2]$ and apply 8 different blur
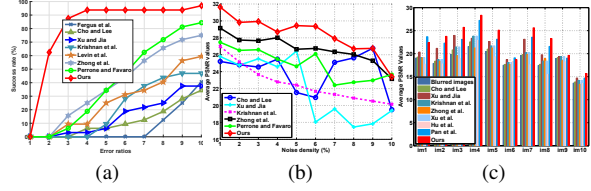

(a)   (b)   (c)

Figure 15. Quantitative evaluation on the dataset with outliers. (a) Results on the dataset with salt and pepper noise. (b) PSNR values of blind deblurring methods on the 10 input images with noise density from 1% to 10%. (c) Results on the dataset with saturated regions (best viewed on high-resolution displays with zoom-in).

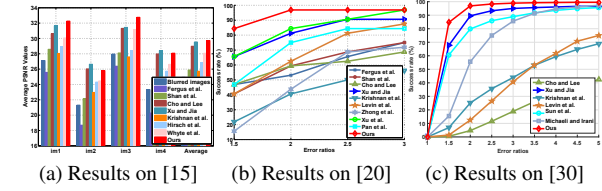(a) Results on [15]   (b) Results on [20]   (c) Results on [30]

Figure 16. Quantitative evaluations on two benchmark datasets.

kernels to generate blurred images where the pixel intensities are clipped into the range of $[0, 1]$. Finally, we add 1% random noise on each blurred image. Figure 15(c) shows that the proposed algorithm achieves favorable results compared with state-of-the-arts. We note that the proposed algorithm can also be applied to deblur images with Gaussian noise. More experimental results are included in the supplemental document.

In addition, we use the natural image deblurring datasets [15], [20], and [30] for evaluation with corresponding metrics [15, 20]. Figure 16 shows that the proposed algorithm performs well on both datasets against the state-of-the-art blind deblurring methods.

6. Conclusions

In this work, we propose a robust kernel estimation algorithm in which effective edges are selected for deblurring images containing significant amount of outliers. We present detailed analysis on the effects of outliers on kernel estimation. Furthermore, we show that the proposed method can be applied to improve the accuracy of existing blind deblurring methods. In the final deconvolution step, we develop a robust method to restore the latent image under the guidance of the proposed outlier-aware function where the effects of outliers are minimized. Extensive experimental evaluations on real images and benchmark datasets demonstrate the proposed algorithm performs favorably against the state-of-the-art methods for uniform as well as non-uniform deblurring.

References

[1] L. Bar, N. Kiryati, and N. A. Sochen. Image deblurring in the presence of impulsive noise. *IJCV*, 70(3):279–298, 2006.

[2] J.-F. Cai, H. Ji, C. Liu, and Z. Shen. Framelet based blind motion deblurring from a single image. *IEEE TIP*, 21(2):562–572, 2012.

[3] S. Cho and S. Lee. Fast motion deblurring. In *SIGGRAPH Asia*, pages 145:1–145:8, 2009.

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

[5] T. S. Cho, S. Paris, B. K. P. Horn, and W. T. Freeman. Blur kernel estimation using the radon transform. In *CVPR*, pages 241–248, 2011.

[6] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski. Edge preserving decompositions for multi-scale tone and detail manipulation. In *SIGGRAPH*, 2008.

[7] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *SIGGRAPH*, pages 787–794, 2006.

[8] M. D. Grossberg and S. K. Nayar. Modeling the space of camera response functions. *IEEE TPAMI*, 26(10):1272–1282, 2004.

[9] S. Harmeling, S. Sra, M. Hirsch, and B. Schölkopf. Multi-frame blind deconvolution, super-resolution, and saturation correction via incremental EM. In *ICIP*, pages 3313–3316, 2010.

[10] J. A. Hertz, A. S. Krogh, and R. G. Palmer. *Introduction to the Theory of Neural Computation*. Westview Press, 1991.

[11] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *ICCV*, pages 463–470, 2011.

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

[13] J. Jia. *Mathematical models and practical solvers for uniform motion deblurring*. Cambridge University Press, 2014.

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

[15] 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 *ECCV*, pages 27–40, 2012.

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

[17] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, pages 2657–2664, 2011.

[18] S. Lee and S. Cho. Recent advances in image deblurring. In *SIGGRAPH Asia 2013 Course*, 2013.

[19] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *SIGGRAPH*, pages 70–78, 2007.

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

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

[22] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astronomy Journal*, 79(6):745–754, 1974.

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

[24] S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, 1990.

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

[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.

[27] J. Pan, R. Liu, Z. Su, and X. Gu. Kernel estimation from salient structure for robust motion deblurring. *Signal Processing: Image Communication*, 28(9):1156–1170, 2013.

[28] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *CVPR*, pages 2909–2916, 2014.

[29] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *SIGGRAPH*, 2008.

[30] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *ICCP*, 2013.

[31] Y.-W. Tai, X. Chen, S. Kim, S. J. Kim, F. Li, J. Yang, J. Yu, Y. Matsushita, and M. S. Brown. Nonlinear camera response functions and image deblurring: Theoretical analysis and practice. *IEEE TPAMI*, 35(10):2498–2512, 2013.

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

[33] Y.-W. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under a projective motion path. *IEEE TPAMI*, 33(8):1603–1618, 2011.

[34] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. In *ICCV Workshops*, pages 745–752, 2011.

[35] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *IJCV*, 98(2):168–186, 2012.

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

[37] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *NIPS*, pages 1790–1798, 2014.

[38] L. Xu, S. Zheng, and J. Jia. Unnatural L_0 sparse representation for natural image deblurring. In *CVPR*, pages 1107–1114, 2013.

[39] H. Zhang, D. P. Wipf, and Y. Zhang. Multi-image blind deblurring using a coupled adaptive sparse prior. In *CVPR*, pages 1051–1058, 2013.

[40] L. Zhong, S. Cho, D. Metaxas, S. Paris, and J. Wang. Handling noise in single image deblurring using directional filters. In *CVPR*, pages 612–619, 2013.