

# Learning Discriminative Data Fitting Functions for Blind Image Deblurring

Jinshan Pan<sup>1</sup> Jiangxin Dong<sup>2</sup> Yu-Wing Tai<sup>3</sup> Zhixun Su<sup>2</sup> Ming-Hsuan Yang<sup>4</sup>  
<sup>1</sup>Nanjing University of Science and Technology <sup>2</sup>Dalian University of Technology  
<sup>3</sup>Tencent Youtu Lab <sup>4</sup>UC Merced

## Abstract

Solving blind image deblurring usually requires defining a data fitting function and image priors. While existing algorithms mainly focus on developing image priors for blur kernel estimation and non-blind deconvolution, only a few methods consider the effect of data fitting functions. In contrast to the state-of-the-art methods that use a single or a fixed data fitting term, we propose a data-driven approach to learn effective data fitting functions from a large set of motion blurred images with the associated ground truth blur kernels. The learned data fitting function facilitates estimating accurate blur kernels for generic scenes and domain-specific problems with corresponding image priors. In addition, we extend the learning approach for data fitting function to latent image restoration and non-uniform deblurring. Extensive experiments on challenging motion blurred images demonstrate the proposed algorithm performs favorably against the state-of-the-art methods.

## 1. Introduction

The goal of blind image deblurring is to recover a blur kernel and a sharp latent image from a blurred input. It is a classical vision problem, and significant progress has been made in recent years [10, 11, 14]. When the blur is spatially invariant, the blur process can be modeled by a convolution operation:

$$B = I * k + n, \quad (1)$$

where  $B$ ,  $I$ ,  $k$ , and  $n$  denote the blur image, latent image, blur kernel, and noise, respectively; and  $*$  is the convolution operator. Blind image deblurring is an ill-posed problem because there are infinite pairs of  $I$  and  $k$  which satisfy (1), and a trivial solution exists, i.e., original blurred image and delta blur kernel.

Although the number of solutions is infinite, the solution space of natural images can be constrained. Numerous methods [2, 4, 15, 16] have been developed based on sparsity of image gradients for kernel estimation. On the other hand, recent algorithms exploit various image priors to recover sharp images, e.g., normalized sparsity prior [13], re-

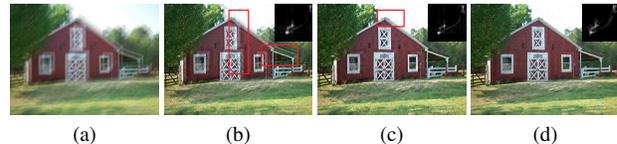


Figure 1. Effect of data fitting functions on kernel estimation. (a) Blurred image. (b) Results of [16] by a data fitting function based on intensity. (c) Results of [16] by a data fitting function based on image gradient. (d) Our results. All the results are generated with the same settings for fair comparisons. The parts enclosed in the red boxes in (b) and (c) contain significant blur residual and artifacts (Best viewed on high-resolution displays with zoom-in).

current internal patch recurrence [18], text image prior [20], and dark channel prior [23]. These image priors are based on statistical assumptions of clear images and have been shown to be effective in deblurring. Discriminative methods [32, 37] have been developed to learn effective image priors [25] for blur kernel estimation. In contrast to statistical priors, several methods use exemplars for kernel estimation [1, 6, 19, 28].

In addition to image priors, another group of methods focus on sharp edge predictions for blur kernel estimation. However, these methods usually involve heuristic edge selection steps [3, 33] to estimate blur kernels.

We note that the aforementioned methods focus on developing effective image priors for deblurring. Among the methods in the literature, intensity information is commonly used to define the data fitting term. Levin et al. [16] show that proper use of image gradients in a data fitting function helps improve blur kernel estimation. Some recent algorithms [3, 20, 22, 23, 33, 35] use intensity in latent image restoration (e.g., minimizing  $\ell_2$  reconstruction errors) and gradient in the kernel estimation (e.g., minimizing  $\ell_2$  errors). However, the effect of intensity and gradient information in blind image deblurring has not been well analyzed.

In this paper, instead of proposing image priors, we study the effect of data fitting functions for kernel estimation. We show that the data fitting function also plays a crucial role in blind image deblurring as it measures the goodness-of-fit to the motion blur model in (1). Figure 1 demonstrates the importance of different data fitting functions on blur kernel estimation.

To address this issue, we propose a two-stage approach for blind image deblurring. In the first stage, an effective data fitting function is learned for blur kernel estimation. In the second stage, the data fitting function is optimized for latent image restoration. We present an efficient numerical algorithm to learn the data fitting functions for both blur kernel estimation and latent image restoration. In addition, we show that the proposed algorithm can be applied to other domain-specific deblurring tasks with different priors and non-uniform deblurring.

## 2. Proposed Algorithm

In this section, we present an algorithm to learn effective data fitting functions for blind image deblurring. We first consider the blur kernel estimation problem, and extend it to the latent image restoration task. In this work, the blind deblurring problem is formulated as

$$E(I, k) = \sum_i \omega_i \|f_i * I * k - f_i * B\|_2^2 + \varphi(I) + \phi(k), \quad (2)$$

where  $\omega_i$  denotes the  $i$ -th weight,  $f_i$  denotes a linear filter operator which can be learned by fields of experts [24], and  $\varphi(I)$  as well as  $\phi(k)$  are the priors of latent image and blur kernel. As the weights control the importance of each term for blur kernel estimation, the main goal is to estimate these values effectively.

### 2.1. Learning Discriminative Data Functions

To estimate  $\omega = \{\omega_i\}$ , we collect a set of ground truth blur kernels  $\{k_j\}$  as well as a set of clear images  $\{I_j\}$ , and propose the following objective function,

$$\begin{aligned} \min_{\omega_i} \frac{1}{2} \sum_j \|k_j(\omega) - k_j^{gt}\|_2^2 \\ \text{s.t. } \omega_i \geq 0, \sum_i \omega_i = 1, \end{aligned} \quad (3)$$

where  $k_j^{gt}$  denotes the  $j$ -th ground truth blur kernel, and  $k_j(\omega)$  is the  $j$ -th estimated blur kernel, which can be obtained by

$$\arg \min_{k_j, I_j} \sum_j \sum_i \omega_i \|f_i * I_j * k_j - f_i * B_j\|_2^2 + \varphi(I_j) + \phi(k_j). \quad (4)$$

To derive the relationship between blur kernel  $k_j$  and weight  $\omega_i$  in (2), we propose an efficient algorithm to solve (4).

### 2.2. Optimizing (4)

Similar to the existing methods [20, 22, 35, 30], we adopt  $\varphi(I_j) = \lambda \|\nabla I_j\|_0$  and  $\phi(k_j) = \gamma \|k_j\|_2^2$  as the regularization for the latent image and blur kernel of (4), where  $\lambda$  and  $\gamma$  are weight parameters. We use the half-quadratic splitting  $L_0$  minimization method [34] and introduce an auxiliary variable  $g_j = (g_j^v, g_j^h)$  corresponding to the image

gradient  $\nabla I_j$ . Thus, (4) can be rewritten as

$$\begin{aligned} \min_{k_j, I_j, g_j} \sum_j \sum_i \omega_i \|f_i * I_j * k_j - f_i * B_j\|_2^2 \\ + \beta \|g_j - \nabla I_j\|_2^2 + \lambda \|g_j\|_0 + \gamma \|k_j\|_2^2. \end{aligned} \quad (5)$$

#### 2.2.1 Intermediate Blur Kernel Estimation

Given  $I_j$ , the optimization with respect to  $k_j$  is

$$\min_{k_j} \sum_j \sum_i \omega_i \|f_i * I_j * k_j - f_i * B_j\|_2^2 + \gamma \|k_j\|_2^2. \quad (6)$$

For simplicity, we use the matrix-vector form to express (6)

$$\min_{\mathbf{k}_j} \sum_j \sum_i \omega_i \|\mathbf{A}_{ij} \mathbf{k}_j - \mathbf{b}_{ij}\|_2^2 + \gamma \|\mathbf{k}_j\|_2^2, \quad (7)$$

where  $\mathbf{A}_{ij}$  is the matrix form of  $f_i * I_j$  with respect to blur kernel  $k_j$ ,  $\mathbf{b}_{ij}$  is the vector form of  $f_i * B_j$  with respect to blur kernel  $k_j$ , and  $\mathbf{k}_j$  is the vector form of  $k_j$ . Based on (7), the solution of  $k_j$  is

$$\mathbf{k}_j = \left( \sum_i \omega_i \mathbf{A}_{ij}^\top \mathbf{A}_{ij} + \gamma \right)^{-1} \left( \sum_i \omega_i \mathbf{A}_{ij}^\top \mathbf{b}_{ij} \right). \quad (8)$$

#### 2.2.2 Intermediate Latent Image Estimation

The optimization problem (5) with respect to intermediate latent image  $I_j$  is

$$\begin{aligned} \min_{I_j, g_j} \sum_j \sum_i \omega_i \|f_i * I_j * k_j - f_i * B_j\|_2^2 + \beta \|g_j - \nabla I_j\|_2^2 \\ + \lambda \|g_j\|_0. \end{aligned} \quad (9)$$

Note that this problem involves variables  $I_j$  and  $g_j$ . It can be efficiently solved through alternatively minimizing  $I_j$  and  $g_j$ .

In each iteration, the solution of  $g_j$  is obtained by solving

$$g_j = \begin{cases} \nabla I_j, & |\nabla I_j| \geq \frac{\lambda}{\beta}, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Given  $g_j$ , the intermediate latent image  $I_j$  can be obtained by solving

$$\min_{I_j} \sum_j \sum_i \omega_i \|f_i * I_j * k_j - f_i * B_j\|_2^2 + \beta \|g_j - \nabla I_j\|_2^2, \quad (11)$$

and the closed-form solution for this problem is

$$I_j = \mathcal{F}^{-1} \left( \frac{\sum_i \omega_i \overline{\mathcal{F}(f_i)} \mathcal{F}(k_j) \mathcal{F}(f_i * B_j) + \beta F_g}{F_k + \beta (\sum_{i \in \{h, v\}} \overline{\mathcal{F}(\nabla_i)} \mathcal{F}(\nabla_i))} \right), \quad (12)$$

---

**Algorithm 1** Solving (9)

---

**Input:** Blurred image  $B_j$  and blur kernel  $k_j$ .

$I_j \leftarrow B_j, \beta \leftarrow 2\lambda$ .

**repeat**

    solve  $g_j$  using (10).

    solve  $I_j$  using (12).

$\beta \leftarrow 2\beta$ .

**until**  $\beta > \beta_{\max}$

**Output:** Intermediate latent image  $I_j$ .

---

---

**Algorithm 2** Learning discriminative features

---

**Input:** Blurred images  $\{B_j\}$ , ground truth blur kernels  $\{k_j^{gt}\}$ .

$\omega_i \leftarrow 0$ .

initialize  $k_j$  with results from the coarser level.

**while**  $l \leq \max\_iter1$  **do**

**while**  $t \leq \max\_iter2$  **do**

        solve  $I_j$  using Algorithm 1.

        solve  $k_j$  using (8).

**end while**

$\omega_i = \omega_i - \alpha \sum_j \frac{\partial \mathcal{L}_j}{\partial \omega_i}$ .

**end while**

**Output:** The weight  $\omega_i$ .

---

where  $\mathcal{F}(\cdot)$  and  $\mathcal{F}^{-1}(\cdot)$  denote the Fourier transform and its inverse transform, respectively,  $\overline{\mathcal{F}(\cdot)}$  is the complex conjugate operator,  $F_k = \sum_i \omega_i \mathcal{F}(f_i) \mathcal{F}(k_j) \mathcal{F}(k_j) \mathcal{F}(f_i)$ , and  $F_g = \overline{\mathcal{F}(\nabla_h)} \mathcal{F}(g_j^h) + \overline{\mathcal{F}(\nabla_v)} \mathcal{F}(g_j^v)$ , where  $\nabla_h$  and  $\nabla_v$  denote the horizontal and vertical differential operators. In case all the values of  $\omega_i$  are zeros in (12) (which will lead to unstable kernel estimation), we set the terms  $\omega_0 \overline{\mathcal{F}(k_j)} \mathcal{F}(f_0 * B_j)$  and  $\omega_0 \mathcal{F}(k_j) \overline{\mathcal{F}(f_0 * B_j)}$  to be  $(\omega_0 + 1) \overline{\mathcal{F}(k_j)} \mathcal{F}(f_0 * B_j)$  and  $(\omega_0 + 1) \mathcal{F}(f_0) \overline{\mathcal{F}(k_j)} \mathcal{F}(k_j) \mathcal{F}(f_0)$ .

The main steps for intermediate latent image estimation are summarized in Algorithm 1.

### 2.3. Optimizing (3)

After obtaining  $k_j$  with respect to  $\omega_i$ , we can solve (3) by a gradient descent method. The gradient with respect to  $\omega_i$  is

$$\begin{aligned} \frac{\partial \mathcal{L}_j}{\partial \omega_i} &= \frac{\partial \mathcal{L}_j}{\partial \mathbf{k}_j} \frac{\partial \mathbf{k}_j}{\partial \omega_i} = -(\mathbf{k}_j - \mathbf{k}_j^{gt})^\top \left( \sum_i \omega_i \mathbf{A}_{ij}^\top \mathbf{A}_{ij} + \gamma \right)^{-1} \mathbf{A}_{ij}^\top \mathbf{A}_{ij} \mathbf{k}_j \\ &\quad + (\mathbf{k}_j - \mathbf{k}_j^{gt})^\top \left( \sum_i \omega_i \mathbf{A}_{ij}^\top \mathbf{A}_{ij} + \gamma \right)^{-1} \left( \mathbf{A}_{ij}^\top \mathbf{b}_{ij} \right), \end{aligned} \quad (13)$$

where  $\mathcal{L}_j = \frac{1}{2} \|\mathbf{k}_j(\omega) - \mathbf{k}_j^{gt}\|_2^2$ . The detailed derivations are presented in the supplemental material.

The main steps for learning discriminative data fitting functions are summarized in Algorithm 2. In this work, the step of gradient decent  $\alpha$  is set to be 0.01.

**Training Data.** We construct a training dataset to learn

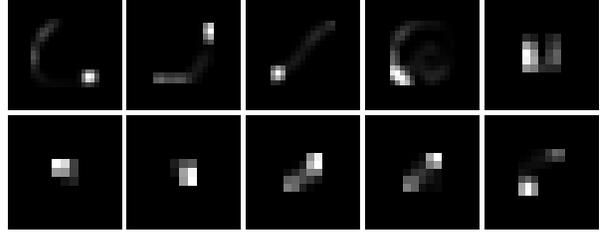


Figure 2. Some generated blur kernels that are used for training.

the weights in (2) by using 200 images from the BSDS dataset [17]. To generate blurred images  $\{B_j\}$  and blur kernels  $\{k_j\}$ , we synthesize realistic blur kernels by sampling random 3D trajectories used in [25]. These trajectories are then projected and rasterized to random square kernel sizes in the range from  $11 \times 11$  up to  $27 \times 27$  pixels. Some examples of the generated blur kernels are shown in Figure 2.

With the blur kernels, we synthetically generate blurred images by convolving each clean image with 100 generated blur kernels. A set of 200,000 blurred images is constructed for learning the weights of (3).

### 2.4. Kernel Estimation

After learning the weights, we solve (2) to obtain the blur kernels. That is, we alternatively solve the intermediate latent image and blur kernel. The optimization algorithms with respect to the blur kernel and latent image are the same to those described in Section 2.2.1 and 2.2.2.

For the linear filters  $\{f_i\}$ , we choose the commonly used zero-order operator corresponding to the intensity information, and gradient operators including the first (two directions) and second order (three directions) operators. The concrete forms of 6 linear operators are presented in Table 1.

### 2.5. Discriminative Non-Blind Deconvolution

Once blur kernels are obtained, we can use a variety of non-blind deconvolution methods to recover latent images. However, we note that the proposed method used in the kernel estimation process can also be applied to non-blind deconvolution. We formulate the non-blind deconvolution problem as

$$\min_I \sum_i \omega_i \|f_i * I * k - f_i * B\|_2^2 + \phi(I), \quad (14)$$

where  $\phi(I)$  is the regularization on the image  $I$ , e.g., hyper-Laplacian priors [12]. In this work, we use the commonly used total variation regularization, i.e.,  $\phi(I) = \mu \|\nabla I\|_1$ , for non-blind deconvolution.

The weight  $\omega_i$  can be obtained by solving

$$\begin{aligned} \min_{\omega_i} & \frac{1}{2} \sum_j \|I_j(\omega) - I_j^{gt}\|_2^2 \\ \text{s.t.} & \quad \omega_i \geq 0, \quad \sum_i \omega_i = 1, \end{aligned} \quad (15)$$

Table 1. Concrete forms of the linear filters used in the learning process.

Filters	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$
Type	zero-order	first order	first order	second order	second order	second order
Forms	$I * k - B$	$\nabla_h I * k - \nabla_h B$	$\nabla_v I * k - \nabla_v B$	$\nabla_h \nabla_h I * k - \nabla_h \nabla_h B$	$\nabla_v \nabla_v I * k - \nabla_v \nabla_v B$	$\nabla_h \nabla_v I * k - \nabla_h \nabla_v B$

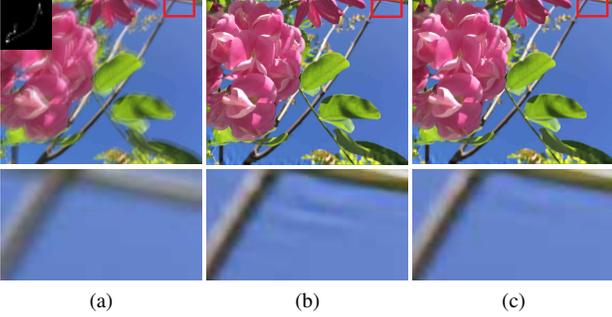


Figure 3. Non-blind deconvolution examples. (a) Blurred image and blur kernel. (b) Result with only intensity information in the data fitting function. (c) Our result. The part in the red box in (b) contains significant ringing artifacts (Best viewed on high-resolution display with zoom-in).

where  $I_j(\omega)$  is the solution of (14) and  $I_j^{gt}$  is the clear image.

To determine the relationship between  $I_j(\omega)$  and  $\omega$ , we use the same alternative minimization method described in Section 2.2.2 to obtain  $I_j(\omega)$ . The weight  $\omega_i$  can be obtained by

$$\omega_i = \omega_i - \alpha_I \sum_j (\mathbf{I}_j - \mathbf{I}_j^{gt})^\top \mathbf{W}_i, \quad (16)$$

where  $\alpha_I$  is the gradient descent step. In the above equation,  $\mathbf{I}_j$ ,  $\mathbf{I}_j^{gt}$ , as well as  $\mathbf{W}_i$  denote the vectorization of  $I_j$ ,  $I_j^{gt}$  and  $W_i$ , respectively. Each  $W_i$  is defined by

$$W_i = \mathcal{F}^{-1} \left( \frac{\Delta_b}{\Delta_d} - \frac{\Delta_f \Delta_n}{\Delta_d^2} \right), \quad (17)$$

where  $\Delta_d = F_k + \beta (\sum_{i \in \{h,v\}} \mathcal{F}(\nabla_i) \mathcal{F}(\nabla_i))$ ,  $\Delta_f = \mathcal{F}(f_i) \mathcal{F}(k_j) \mathcal{F}(k_j) \mathcal{F}(f_i)$ ,  $\Delta_b = \mathcal{F}(f_i) \mathcal{F}(k_j) \mathcal{F}(f_i * B)$ , and  $\Delta_n = \sum_i \omega_i \mathcal{F}(k_j) \mathcal{F}(f_i * B_j) + \beta F_g$ .

We use the same training data as discussed in Section 2.3 to learn  $\omega_i$ . The details regarding the gradient of (15) with respect to  $\omega_i$  and the optimization method with respect to  $I$  in (14) are included in the supplemental material.

Figure 3 shows an example of the non-blind deconvolution result using (14). We note that the recovered image by the conventional data fitting function contains some ringing artifacts (Figure 3(b)) while the one by the proposed method is sharper (Figure 3(c)).

### 3. Extension to Non-Uniform Deblurring

Our method can be directly extended to handle non-uniform deblurring where the blurred images are acquired from moving cameras (e.g., with rotational and translational movements) [5, 7, 26, 29, 31]. Based on the geometric model of camera motion [29, 31], the non-uniform blur process

can be formulated as:

$$\mathbf{B} = \mathbf{K}\mathbf{I} + \mathbf{n} = \mathbf{A}\mathbf{k} + \mathbf{n}, \quad (18)$$

where  $\mathbf{I}$ ,  $\mathbf{k}$ , and  $\mathbf{n}$  denote vector forms of  $I$ ,  $k$ ,  $n$  in (1). In this model,  $\mathbf{A}$  as well as  $\mathbf{K}$  denote the image matrix and blur kernel matrix with respect to the latent image  $\mathbf{I}$  and blur kernel  $\mathbf{k}$ . Based on (18), the non-uniform deblurring problem is solved by alternatively minimizing:

$$\min_{\mathbf{I}} \sum_i \omega_i \|\mathbf{K}\mathbf{F}_i \mathbf{I} - \mathbf{F}_i \mathbf{B}\|_2^2 + \lambda \|\nabla \mathbf{I}\|_0, \quad (19)$$

and

$$\min_{\mathbf{k}} \sum_i \omega_i \|\mathbf{A}_i \mathbf{k} - \mathbf{B}_i\|_2^2 + \gamma \|\mathbf{k}\|_2^2, \quad (20)$$

where  $\mathbf{F}_i$  is the matrix of the filter operator  $f_i$ . We use the fast forward approximation methods [7, 21] to estimate latent images and blur kernels.

## 4. Analysis of Proposed Algorithm

In this section, we analyze how the proposed algorithm performs on image deblurring. We also demonstrate the importance of the proposed learned data fitting functions and discuss the connections to other methods.

### 4.1. Effect of Discriminative Data Fitting Functions

The proposed method is able to automatically learn the most relevant data fitting functions for both blur kernel estimation and latent image restoration. We analyze its effect on blur kernel estimation and latent image restoration with comparisons to the commonly used data fitting functions.

**Effect on Blur Kernel Estimation.** The data fitting functions used in existing methods are based on intensity or gradient for both latent image restoration and kernel estimation [3, 20, 33, 35]. Figure 4 shows one example where the deblurred results by the methods with only intensity or gradient contain significant ringing artifacts. The methods using intensity for the intermediate latent image estimation and gradient for the kernel estimation perform better. However, the deblurred results still contain blur residual and blurry characters. In contrast, the deblurred results generated by the proposed method contain clearer characters, which indicate that the learned data fitting functions facilitate blur kernel estimation.

We quantitatively evaluate the proposed method and present the results in Table 2. The proposed method with learned data fitting functions performs well against other alternatives based on intensity, gradient, or combination.

Table 2. Quantitative comparisons with the commonly used data fitting functions for the image deblurring.

	With only intensity	With intensity and gradient	With only gradient	Ours
Average PSNRs	23.08	30.52	26.78	31.88

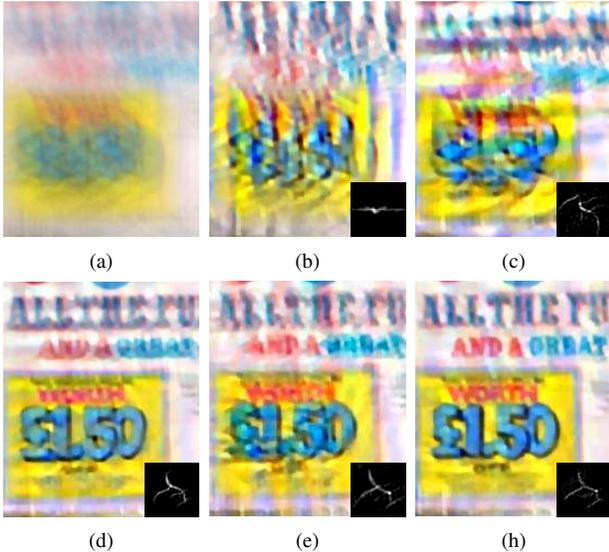


Figure 4. Effectiveness of the proposed learned data fitting function for blur kernel estimation. (a) Blurred image. (b) Results by only intensity in the data fitting function. (c) Results by only gradient in the data fitting function. (d) Results by intensity in the intermediate latent image estimation and gradients in the kernel estimation. (e) Deblurred results by [35]. (f) Our results.

Table 3. Learned weights for blur kernel estimation.

$\omega_1$	$\omega_2$	$\omega_3$	$\omega_4$	$\omega_5$	$\omega_6$
0	0.1954	0.1850	0.2463	0.2625	0.1108

**Learned Weights for Data Fitting Terms.** To illustrate the importance of each data fitting term in blur kernel estimation process, we show the learned weights in Table 3. We note that the learned weight of the data fitting term for intensity is 0, which demonstrates that intensity does not help the blur kernel estimation. The results are similar to the experimental analysis of the state-of-the-art methods [16, 3, 33, 35, 20]. In addition, we note that the weights, i.e.,  $\omega_4$  and  $\omega_5$ , for the data fitting terms with the second order information are much larger than those of the data fitting terms with the first order intensity, which indicate that higher order information plays more important roles for blur kernel estimation.

**Effect on Non-Blind Deconvolution.** The learned weights of data fitting terms for latent image restoration are shown in Table 4. We note that the weight of the data fitting term with the zero-order filter is much higher than the others. This indicates that intensity information plays an important role in non-blind deconvolution. The learned weights in Table 3 and 4 demonstrate that different data fitting terms should be used as kernel estimation and non-blind deconvolution are different processes.

Table 4. Learned weights for latent image estimation.

$\omega_1$	$\omega_2$	$\omega_3$	$\omega_4$	$\omega_5$	$\omega_6$
0.2095	0.1581	0.1581	0.1581	0.1581	0.1581

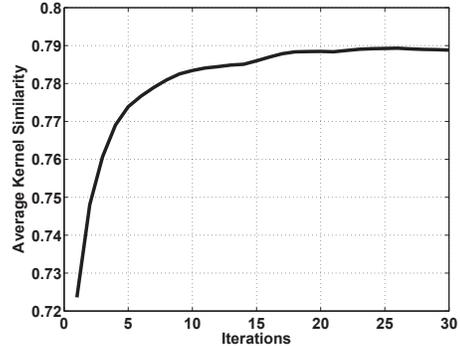


Figure 5. Fast convergence property of the proposed algorithm.

## 4.2. Fast Convergence Property

Compared to the existing methods with fixed data fitting functions based on only intensity, gradient or combination [20, 35], the proposed algorithm involves additional data fitting terms with different weights. However, this does not significantly increase the computational load. We evaluate the convergence rate of the proposed method on the dataset [15] and show the kernel similarity [9] with respect to iterations in Figure 5. The results demonstrate that the proposed method exhibits fast convergence. Table 5 shows that the run time of our method compares favorably against the competing methods.

Table 5. Run time (seconds) on the same computer with an Intel Core i7-4800MQ processor and 16 GB RAM. The run time of Xu et al. [35] is based on our implementation.

Method	255 × 255	600 × 600	800 × 800
Xu et al. [35]	3.10	19.10	36.53
Krishnan et al. [13]	34.01	196.09	315.41
Levin et al. [16]	144.61	501.67	862.81
Pan et al. [23]	17.07	115.86	195.80
Ours	4.93	23.11	41.52

## 5. Experimental Results

We present experimental evaluations of the proposed algorithm against the state-of-the-art deblurring methods. All the experiments are carried out on a machine with an Intel Core i7-4800MQ processor and 16 GB RAM. The run time for a 255 × 255 image is 5 seconds on MATLAB. In all the experiments, we set  $\lambda = 0.002$  and  $\gamma = 2$ . We empirically set  $\beta_{\max} = 10^5$  in Algorithm 1. We use the deblurring datasets by Sun et al. [27] and Levin et al. [15] as the main test datasets. Thus, the training and test datasets are not overlapped. For fair comparisons, we use the executable

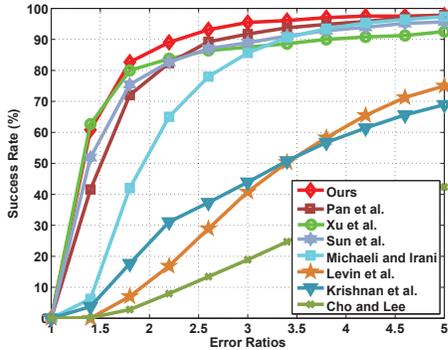


Figure 6. Quantitative evaluations on the benchmark dataset by Sun et al. [27]. Our method performs favorably against the state-of-the-art methods.

code provided by the authors and tune the parameters to generate the best possible results of other methods. The MATLAB code and datasets are publicly available on the authors’ websites. More experimental results can be found in the supplemental material.

### 5.1. Quantitative Evaluation

We evaluate the proposed method on the synthetic dataset by Sun et al. [27] and compare it with several state-of-the-art deblurring methods [3, 13, 16, 18, 21, 27, 35]. This dataset contains 80 images and 8 blur kernels from [15]. We use the original codes of the state-of-the-art methods [3, 13, 16, 18, 21, 27, 35] to estimate blur kernels and use the non-blind deblurring method [36] to generate the final deblurring results for fair comparisons. The error ratio [15] is used for performance evaluation. Figure 6 shows the quantitative results on the dataset [27]. Overall, the proposed algorithm performs favorably against the state-of-the-art deblurring methods. We note that the method [35] uses intensity in intermediate latent image estimation and gradient in the kernel estimation. Compared with this method, the proposed algorithm achieves higher success rates, which indicates the effectiveness of the learned data fitting functions.

### 5.2. Real Images

Figure 7 shows the deblurred results on a real image by the proposed algorithm and the state-of-the-art methods. We use the original source or binary codes and tune the parameters to generate the best possible results for fair comparisons. The deblurred images by [3, 13] contain significant ringing artifacts. While the state-of-the-art methods [16, 20, 23] generate better kernel estimates than other methods, the deblurred images contain significant ringing artifacts. We note that the main difference between the proposed algorithm and the method by Xu et al. [35] is that [35] uses intensity for latent image restoration and gradient for kernel estimation. However, the results demonstrate that

this manually designed data fitting function is not effective for blur kernel estimation on the real image. In contrast, the deblurred image by the proposed algorithm contains fewer artifacts, which shows that the learned function with different weighted combination of data fitting terms is effective for kernel estimation.

Figure 8 shows the deblurred results by the proposed algorithm and the state-of-the-art methods [3, 13, 16, 20, 23, 35] on a real blurred document image. The state-of-the-art deblurring methods designed for natural images [3, 13] do not generate clear images. Although the method by Pan et al. [20] mainly focuses on the text image deblurring, the deblurred results still contain significant blur residual and ringing artifacts. The recent method based on sparsity of dark channel prior for blur kernel estimation [23] does not perform well on this image as the assumption on zero-intensity values of an image does not hold. In contrast, the deblurred image by the proposed algorithm is clearer with significantly fewer artifacts. Furthermore, the deblurred results shown in Figure 8(e) and (h) demonstrate the effectiveness of the proposed algorithm that learns different weighted functions of data fitting terms for latent image restoration and blur kernel estimation.

### 5.3. Non-uniform Deblurring

As discussed in Section 3, the proposed algorithm can be extended to handle non-uniform blur. We present results on an image degraded by spatially variant motion blur provided by [5] in Figure 9. We compare the proposed algorithm with the state-of-the-art non-uniform deblurring methods [5, 7, 8, 21, 31, 35]. Figure 9 shows the letters of (b)-(f) contain ringing artifacts. Compared to the deblurred results by the state-of-the-art non-uniform methods, the restored image by the proposed algorithm contains sharper contents with fewer artifacts.

### 5.4. Extensions of Proposed Method

In this work, we focus on learning effective data fitting functions for blur kernel estimation and use the  $L_0$  norm of image gradient [35] as the image prior. However, our method can be applied to other deblurring tasks with specific image priors, e.g., normalized sparsity prior [13],  $L_0$ -regularized intensity and gradient prior [20], and dark channel prior [23], to name a few. To demonstrate the flexibility of the proposed method, we use the  $L_0$ -regularized intensity and gradient prior [20] as an example and show the deblurred results in Figure 10. We note that the original text deblurring method uses intensity for latent image and gradient for kernel estimation. However, this combination does not always help blur kernel estimation (see Figure 10(b)). In contrast, the method with the learned data fitting function generates deblurred images with clearer characters as shown in Figure 10(c).

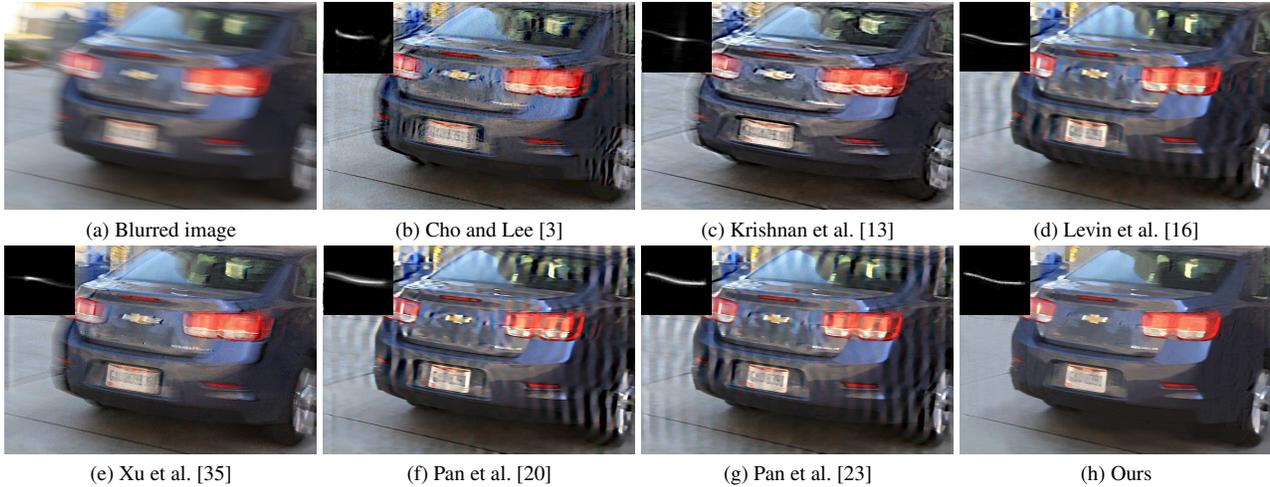


Figure 7. Comparisons on a real image. The proposed method generates a better kernel estimate and deblurred result with fewer ringing artifacts. The comparison results shown in (e) and (h) indicate the importance of the proposed automatically learned data fitting functions.

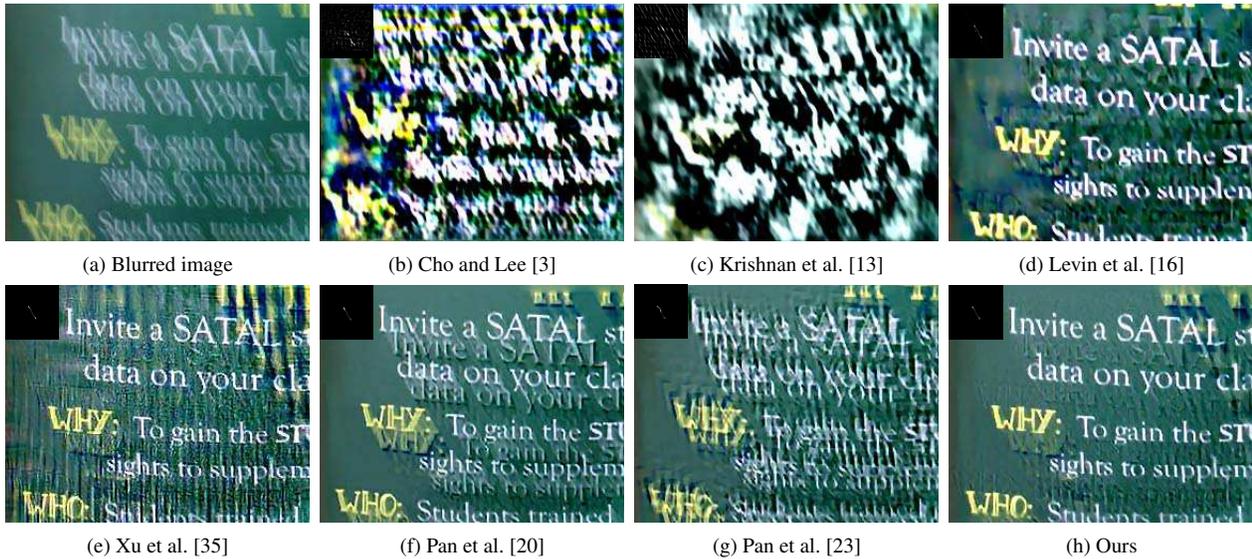


Figure 8. Comparisons on a real image with lots of characters. The proposed method generates an image with clearer characters. The comparison results shown in (e) and (h) indicate the importance of the proposed learned data fitting functions.

**Comparisons with [32].** The recent method [32] aims to learn high-order filters for image deblurring. The image prior based on the learned high-order filters is especially effective for text images. However, our method focuses on learning good data fitting functions instead of image priors for image deblurring. Figure 11 shows a real blurred text image from [20]. The proposed method with the  $L_0$ -regularized intensity and gradient prior performs competitively against the state-of-the-art text deblurring methods [20, 32].

## 6. Concluding Remarks

In this paper, we propose an effective algorithm which learns effective data fitting functions for both blur kernel estimation and latent image restoration. We discuss the effect of the proposed data fitting functions and show that intensity

has less effect on blur kernel estimation and has more effect on latent image restoration. We show that the performance of deblurring algorithms using the learned data fitting functions can be significantly improved. The proposed algorithm is also extended for non-uniform deblurring. In addition, we show that the proposed method can be extended to other specific deblurring tasks with corresponding image priors. Extensive experimental evaluations on benchmark datasets and real images demonstrate that the proposed algorithm performs favorably against the state-of-the-art methods for uniform as well as non-uniform deblurring.

While we focus on learning effective data fitting functions for blind image deblurring, the choice of linear filters is fixed. In addition, the optimization method may also play an important role [4, 16] and some models may accommo-



Figure 9. The proposed method directly applies to images with non-uniform blur. The images shown in (b)-(g) are directly obtained from the reported results in [21] (Best viewed on high-resolution display with zoom-in).

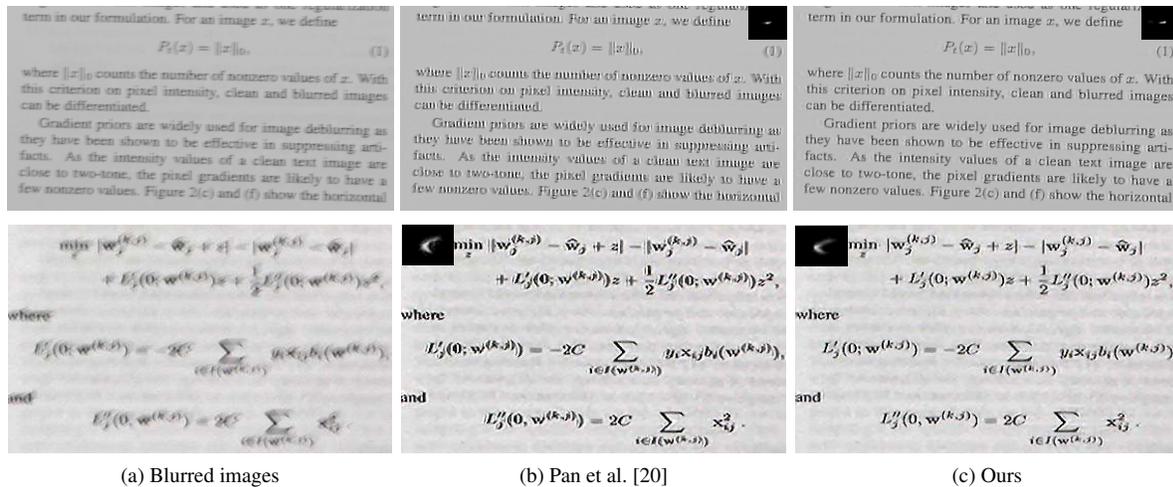


Figure 10. Extension of text image deblurring algorithm [20]. The results generated by the proposed method contain clearer characters and fewer ringing artifacts (Best viewed on high-resolution display with zoom-in).

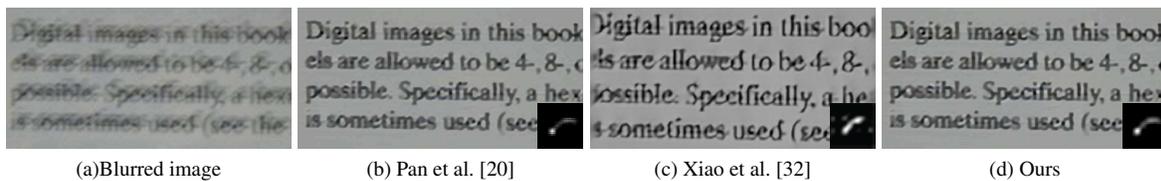


Figure 11. Comparisons with [32] on a real text image from [20]. The images shown in (b)-(c) are directly obtained from the reported results (Best viewed on high-resolution display with zoom-in).

date better optimization strategies than others. Our future work will focus on learning effective linear filters and optimization methods for image deblurring.

**Acknowledgements.** This work has been partially support-

ed by NSFC (No. 61572099, 61522203), NSF CAREER (No. 1149783), 973 Program (No. 2014CB347600), NSF of Jiangsu Province (No. BK20140058), the National Key R&D Program of China (No. 2016YFB1001001), and gifts from Adobe and Nvidia.

## References

- [1] S. Anwar, C. P. Huynh, and F. Porikli. Class-specific image deblurring. In *ICCV*, pages 495–503, 2015. 1
- [2] T. Chan and C. Wong. Total variation blind deconvolution. *IEEE TIP*, 7(3):370–375, 1998. 1
- [3] S. Cho and S. Lee. Fast motion deblurring. In *SIGGRAPH Asia*, volume 28, page 145, 2009. 1, 4, 5, 6, 7
- [4] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM SIGGRAPH*, 25(3):787–794, 2006. 1, 7
- [5] A. Gupta, N. Joshi, C. L. Zitnick, M. F. Cohen, and B. Curless. Single image deblurring using motion density functions. In *ECCV*, pages 171–184, 2010. 4, 6, 8
- [6] Y. HaCohen, E. Shechtman, and D. Lischinski. Deblurring by example using dense correspondence. In *ICCV*, pages 2384–2391, 2013. 1
- [7] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *ICCV*, pages 463–470, 2011. 4, 6, 8
- [8] Z. Hu and M.-H. Yang. Fast non-uniform deblurring using constrained camera pose subspace. In *BMVC*, pages 1–11, 2012. 6, 8
- [9] Z. Hu and M.-H. Yang. Good regions to deblur. In *ECCV*, pages 59–72, 2012. 5
- [10] J. Jia. *Mathematical models and practical solvers for uniform motion deblurring*. Cambridge University Press, 2014. 1
- [11] 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. 1
- [12] D. Krishnan and R. Fergus. Fast image deconvolution using Hyper-Laplacian priors. In *NIPS*, pages 1033–1041, 2009. 3
- [13] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, pages 2657–2664, 2011. 1, 5, 6, 7
- [14] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang. A comparative study for single-image blind deblurring. 2016. 1
- [15] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, pages 1964–1971, 2009. 1, 5, 6
- [16] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, pages 2657–2664, 2011. 1, 5, 6, 7
- [17] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–423, 2001. 3
- [18] T. Michaeli and M. Irani. Blind deblurring using internal patch recurrence. In *ECCV*, pages 783–798, 2014. 1, 6
- [19] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring face images with exemplars. In *ECCV*, pages 47–62, 2014. 1
- [20] J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via  $L_0$ -regularized intensity and gradient prior. In *CVPR*, pages 2901–2908, 2014. 1, 2, 4, 5, 6, 7, 8
- [21] J. Pan, Z. Hu, Z. Su, and M.-H. Yang.  $L_0$ -regularized intensity and gradient prior for deblurring text images and beyond. *IEEE TPAMI*, 39(2):342–355, 2017. 4, 6, 8
- [22] J. Pan and Z. Su. Fast  $\ell^0$ -regularized kernel estimation for robust motion deblurring. *IEEE Signal Processing Letters*, 20(9):841–844, 2013. 1, 2
- [23] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *CVPR*, pages 1628–1636. 1, 5, 6, 7
- [24] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *CVPR*, pages 860–867, 2005. 2
- [25] U. Schmidt, C. Rother, S. Nowozin, J. Jancsary, and S. Roth. Discriminative non-blind deblurring. In *CVPR*, pages 604–611, 2013. 1, 3
- [26] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, pages 1–8, 2007. 4
- [27] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *ICCV*, 2013. 5, 6
- [28] L. Sun, S. Cho, J. Wang, and J. Hays. Good image priors for non-blind deconvolution - generic vs. specific. In *ECCV*, pages 231–246, 2014. 1
- [29] 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. 4
- [30] J. Tang, X. Shu, G. Qi, Z. Li, M. Wang, S. Yan, and R. Jain. Tri-clustered tensor completion for social-aware image tag refinement. *IEEE TPAMI*, 39(8):1662–1674, 2017. 2
- [31] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *IJCV*, 98(2):168–186, 2012. 4, 6, 8
- [32] L. Xiao, J. Wang, W. Heidrich, and M. Hirsch. Learning high-order filters for efficient blind deconvolution of document photographs. In *ECCV*, 2016. 1, 7, 8
- [33] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, pages 157–170, 2010. 1, 4, 5
- [34] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via  $L_0$  gradient minimization. In *SIGGRAPH Asia*, volume 30, page 174, 2011. 2
- [35] L. Xu, S. Zheng, and J. Jia. Unnatural  $L_0$  sparse representation for natural image deblurring. In *CVPR*, pages 1107–1114, 2013. 1, 2, 4, 5, 6, 7, 8
- [36] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, pages 479–486, 2011. 6
- [37] W. Zuo, D. Ren, S. Gu, L. Lin, and L. Zhang. Discriminative learning of iteration-wise priors for blind deconvolution. In *CVPR*, pages 3232–3240, 2015. 1